Nummer 27 **Oktober 2000**

Werbeseite der Fa. Texas Instruments

Mitteilungen der Sprecher

Liebe Mitglieder der Fachgruppe Computeralgebra,

am 22. September trafen sich die Mitglieder der Fachgruppenleitung im Anschluss an die DMV-Tagung in Leipzig, um über aktuelle Fragen zu diskutieren und die vorliegende Ausgabe des Rundbriefs zu besprechen.

Das Hauptthema unseres Treffens war die Entwicklung der Computeralgebra in der Lehre. Wir blickten zurück auf die von der Fachgruppe, insbesondere aber von den Kollegen Prof. Dr. Kerber, StD Knechtel und Prof. Dr. Koepf organisierte Tagung in Thurnau. Das Gelingen dieser Tagung gibt uns Mut, uns weiterhin für die Förderung der Computeralgebra an Schulen einzusetzen. Einen Bericht zur Tagung finden Sie in diesem Heft.

Wie im letzten Heft des Rundbriefs angekündigt, steht in dieser Ausgabe ein vergleichender Überblick über drei CAS-Taschenrechner. Der Autor dieses Vergleichs, Herr Prof. Dr. Henn (Univ. Dortmund), ist seit langem bekannt als Experte für den Computereinsatz in Schulen. Angeregt durch seinen Artikel hat sich die Fachgruppenleitung entschlossen, den von Herrn Prof. Dr. Henn zitierten Beitrag "Welche handwerklichen Rechenkompetenzen sind im CAS-Zeitalter unverzichtbar?" *ebenfalls hier abzudrucken. Dieser Beitrag ist, wie aus seiner Zusammenfassung hervorgeht, eine Aufforderung zu Diskussion. Wir können uns vorstellen, dass auch unsere Leser unterschiedlicher Ansicht sind. Wir möchten im Rundbrief Gelegenheit geben zu unterschiedlichen Meinungsäußerungen und laden Sie herzlich ein, uns per email oder per Post Ihre Meinung zu sagen.*

Als weiteren Beitrag zur Förderung der Computeralgebra in Lehre, Ausbildung und Weiterbildung ist unser Angebot zu sehen, Internetpublikationen zu begutachten und die Ergebnisse im Rundbrief zu veröffentlichen. Bitte machen Sie uns auf interessante Internetseiten zu diesem Thema aufmerksam. Bei uns sammelt Prof. Dr. Koepf die zur Begutachtung vorgesehenen Seiten. Wir hoffen, im nächsten Heft schon die ersten Begutachtungen veröffentlichen zu können.

H. Michael Möller *M. Pohst*

Hinweise auf Konferenzen

1. **ACAT'2000 – VII International Workshop on Advanced Computing and Analysis Techniques in Physics Research**

   Fermi National Accelerator Laboratory, USA, 16.10. – 20.10.2000

   Recent revolutions in computer hardware and software technologies have paved the way for the development of advanced methodologies in computing, data exploration and analyses. Adopting these advanced techniques is crucial to the success of increasingly more complex and challenging projects we undertake in physics research. The aim of this workshop is to provide an open forum where knowledge of these techniques can be disseminated and discussed with the goal of generating new ideas and initiatives that will have a significant beneficial impact on research.

   The workshop will cover four main topics:
   (1) Artificial Intelligence(AI),
   (2) Innovative Software Engineering(ISE),
   (3) Computer-aided Symbolic Algebra(CASA) and
   (4) Very Large-scale Computing(VLC).

   Participants will learn about, and be able to discuss, the conceptual advances in these areas. We shall address many of the practical problems encountered in: on-line triggering, monitoring and analysis of data in large physics experiments; operation and control of accelerators; data selection and pattern recognition of complex and rare physical phenomena; computation of complex and large-scale theoretical calculations to a high degree of precision, and large-scale simulation of events, accelerator and detector components.

   The workshop should be of great interest to the on-going and future high energy, nuclear and astrophysics experiments which will need effective means of mining and exploring, and efficiently analyzing, unprecedented amounts of data. Present and future endeavors in accelerator physics should similarly benefit from the topics covered.

   Co-Chairs : Pushpalatha C. Bhat, Matthias Kasemann, `acat2000@fnal.gov` .

   `http://conferences.fnal.gov/acat2000/`

2. **ASCM'2000 – The 4th Asian Symposium on Computer Mathematics**

Chiang Mai, Thailand, 17.12. – 21.12.2000

The 4th Asian Symposium on Computer Mathematics (ASCM'2000) will be held jointly with the 5th Asian Technology Conference in Mathematics (ATCM'2000) in Chiang Mai, Thailand from December 17-21, 2000. The previous symposia ASCM'95, 96, 98 in the series were held in Beijing (China), Kobe (Japan), and Lanzhou (China), respectively.

ASCM'2000 will offer an opportunity for participants to present original research, to learn of research progress and new developments, and to exchange ideas and views on doing mathematics using computers. Original research papers on all aspects of computer mathematics are solicited for ASCM'2000.

Specific topics for the symposium include but are not limited to:

Symbolic, algebraic, and geometric computation, Automated mathematical reasoning, Computer-aided problem solving and instruction, Computational algebra and geometry, Symbolic/numeric hybrid methods, Parallel/distributed/network computing, Mathematical software design and implementation, Applications in CAGD/CAD, robotics and computer vision.

The scientific program of ASCM'2000 will contain invited talks and presentations of contributed papers, together with software demonstration.

Invited speakers: Gaston H. Gonnet (ETH Zurich, Switzerland), Daniel Lazard (Universite Paris VI, France), William McCune (Argonne National Laboratory, USA).

Program committee:

Xiao-Shan Gao (Chinese Academy of Sciences, China) Co-Chair, Dongming Wang (CNRS, France) Co-Chair, Shang-Ching Chou (Wichita State University, USA), Mark J. Encarnacion (University of the Philippines, Philippines), Jieh Hsiang (National Taiwan University, Taiwan), Hoon Hong (North Carolina State University, USA), Hidetsune Kobayashi (Nihon University, Japan), Hongbo Li (Chinese Academy of Sciences, China), Tien-Yien Li (Michigan State University, USA), Zhibin Li (Lanzhou University, China), Zhuojun Liu (Chinese Academy of Sciences, China), Matu-Tarow Noda (Ehime University, Japan), Deepak Kapur (University of New Mexico, USA), Tadashi Takahashi (Kobe University, Japan), Paul S. Wang (Kent State University, USA), Lu Yang (Chinese Academy of Sciences, China), Kazuhiro Yokoyama (Fujitsu Laboratories Limited, Japan), Hangtao Zhang (University of Iowa, USA), Zhiming Zheng (Beijing University, China).

Paper submission:

Authors are invited to submit papers by E-mail (or snail mail) to both of the PC co-chairs:
Prof. Xiao-Shan Gao, Institute of Systems Science, Academia Sinica, Beijing 100080, P. R. China,
E-mail: `xgao@mmrc.iss.ac.cn`, Fax: (86) 10 6263 0706.
Dr. Dongming Wang, Laboratoire d'Informatique de Paris 6, Universite Pierre et Marie Curie - CNRS, 4, place Jussieu, F-75252 Paris Cedex 05, France, E-mail: `Dongming.Wang@lip6.fr`, Fax: (33) 1 44 27 40 42.

Papers should be written in English and should not exceed 10 pages. The first page of each paper should contain its title, author(s) with affiliation(s) and e-mail address(es), and an abstract.

Important dates:
June 30, 2000: deadline for submission of papers
August 15, 2000: notification of acceptance
August 31, 2000: deadline for camera-ready copy

Further information: `http://www.mmrc.iss.ac.cn/~ascm/`, `ascm@mmrc.iss.ac.cn` .

3. **ATCM'2000 – The 4th Asian Technology Conference in Mathematics**

Chiang Mai, Thailand, 17.12. – 21.12.2000

`http://www.cs.runet.edu/ atcm/ATCM00/`

4. **Finite Fields: Theory and Applications**

Oberwolfach, 7.1. – 13.1.2001

Organizers: J.von zur Gathen (Paderborn), I. E. Shparlinski (NSW)

5. **PASI Conference on Computational Materials Science**

Santiago, Chile, 8.1. – 19.1.2001

Computational Materials Science (CMS) is one of the broadest, most rapidly evolving and technologically significant areas of science today. From its foundations in basic science CMS branches to new materials, to processing, and to the life sciences. Due to rapid advances in computational capacity, including new algorithms, software macro-packages and low-cost PC's, CMS problems previously thought intractable are actively being pursued. The enhanced capabilities of CMS are increasingly impacting the search for new materials, the improvement and design of processing techniques, and the research in bio-areas such as protein folding.

The PASI conference will help establish networking contacts in related but crucial fields (including computer science and applied mathematics).

The following **guest speakers** at the PASI conference in Santiago Chile will be: Mark Robbins, Johns Hopkins University, USA, Malcolm Stocks, Oak Ridge National Laboratory, USA, Jos Onuchic, University of California - San Diego, USA, Ana Mara Llois, Comisin Nacional de Energa Atmica, Argentina, Mark Gordon, Iowa State University, USA, Franco Nori, University of Michigan, USA, Ivan K. Schuller, University of California - San Diego, USA, Renata Wentzcovitch, University of Minnesota, USA, Rafael Barrios, UNAM, Mxico, Mariana Weissmann, Centro Nacional de Energia Atomica, Argentina, Ricardo Ramirez, Pontificia Universidad Catlica, Chile, Jorge Sofo, Centro Atmico Bariloche-Instituto Balseiro, Argentina.

Deadline for application submission is June 15, 2000. Decisions will be announced by July 1, 2000. Please use the link provided below to submit application via the world wide web.

For any **further information:** contact: Adah Leshem-Ackerman (`adah@iastate.edu`) Phone: (515) 294-1253 Fax: (515) 294-9933 `http://www.iitap.iastate.edu/iitap/pasi.html`

6. **T^3 – teachers teaching teachers**

   Columbus, USA, 16.3. – 18.3.2001

   `http://www.t3ww.org/t3/confoverview.htm`

7. **Dagstuhl-Seminar 01201: Algorithms and Number Theory**

   Dagstuhl, 13.5 – 18.5.2001

   Organizers: J. Buhler (Berkeley), H. Niederreiter (Wien), M. Pohst (Berlin)

8. **IMACS-ACA'2001 – The Seventh International IMACS Conference on Applications of Computer Algebra**

   Albuquerque, New Mexico, USA, 31.5. – 3.6.2001

   IMACS (International Association for Mathematics and Computers in Simulation) will be a joint conference with NMMATYC (New Mexico Mathematical Association of Two-Year Colleges), whose conference will be held June 1–3 at TVI.

   General Chair: Bill Pletsch (`bpletsch@tvi.cc.nm.us`)
   Program Chair: Stanly Steinberg (`stanly@math.unm.edu`)
   Dis Organizer: Michael Wester (`wester@math.unm.edu`)

   Scope: The meeting will focus on actual or possible applications of nontrivial computer algebra techniques to other fields and substantial interactions of computer algebra with other fields.

   Conference Topics: Recent IMACS-ACA conferences have included: applications of computer algebra to robotics, signal processing, quantifier elimination, approximate algebraic computation, symbolic-numeric algorithms, combinatorial and computational methods in algebraic geometry, computations in pure mathematics, computer aided geometric design, dynamical systems and mechanics, education, demos of computer algebra software, applications to physics, number theory, computer science, mathematics on the Internet, problem solving environments and other specialized sessions.

   Further information can be found at: `http://math.unm.edu/ACA/2001/200.html` or `http://math.unm.edu/aca.html` (main ACA page)

9. **Théorie de Galois et Géométrie**

   Luminy, Frankreich, 4.6. – 8.6.2001

   Organizers: P. Dèbes (Lille), D. Harbater (Philadelphia), F. Pop (Bonn)
   (Tagung des Training and Research Network: Galois Theory and Explicit Methods (GTEM))

10. **Explicit Methods in Number Theory**

    Oberwolfach, 22.7. – 28.7.2001

    Organizers: H. Cohen (Talence), H. W. Lenstra (Berkeley/Leiden), D. B. Zagier (Bonn/Utrecht)

11. **ISSAC 2001 – International Symposium on Symbolic and Algebraic Computation**

    Ontario, Canada, 22.7. – 25.7.2001

    ISSAC is the yearly premier international symposium in Symbolic and Algebraic Computation that provides an opportunity to learn of new developments and to present original research results in all areas of symbolic mathematical computation. ISSAC'2001 will be locally hosted by the University of Western Ontario at London and the Ontario Research Centre for Computer Algebra (ORCCA).

    Important dates:
    Before January 15, 2001 Submission to the Program Committee Chair deadline
    March 12, 2001 Notification of acceptance
    April 2, 2001 Camera -ready copy received

Original research results and insightful analyses of current concerns are solicited for submission. Papers will be reviewed by a program committee and referees. Survey articles may be suitable for submission if identified as such, they will be considered in a separate category from the research papers. Proceedings will be distributed at the symposium. Electronic submission is encouraged.

Submissions must not substantially duplicate work published or submitted for publication elsewhere. Late submissions will be rejected.

Program Committee Chair: Gilles Villard `Gilles.Villard@ens-lyon.fr`

Conference Topics: Topics of the meeting include, but are not limited to:
Algorithmic mathematics. Algebraic, symbolic and symbolic-numeric algorithms. Simplification, function manipulation, equations, summation, integration, ODE/PDE, linear algebra, number theory, group and geometric computing.

Computer Science. Theoretical and practical problems in symbolic computation. Systems, problem solving environments, user interfaces, softwares, libraries, parallel/distributed computing and programming languages for symbolic computation, concrete analysis, benchmarking, complexity of computer algebra algorithms, automatic differentiation, code generation, mathematical data structures and exchange protocols.

Applications. Problem treatments using algebraic, symbolic or symbolic-numeric computation in an essential or a novel way. Engineering, economics and finance, physical and biological sciences, computer science, logic, mathematics, statistics, education.

Program Committee:

Chair: Gilles Villard, CNRS IMAG Grenoble, `Gilles.Villard@ens-lyon.fr`
Proceedings Editor: Bernard Mourrain, INRIA Sophia Antipolis, `mourrain@sophia.inria.fr`

Best student author award: This award will be given to the best student author. An author is eligible if full-time student at the time of submission, this should be indicated.

Notification: Authors will be sent notification of acceptance or rejection by e-mail on or before March 12, 2001. A final copy of each accepted paper will be required by April 2, 2001. This is again a firm deadline. An author of each accepted paper must attend the Conference and present the paper, or make arrangements to have it presented.

Preparing final versions: Formatting requirements will be based on the ACM Proceedings Templates, for instance in LaTeX2e you should use the acm_proc_article-sp.cls document class file to format your document.

Further information: `http://www.orcca.on.ca/issac2001/`

12. Computational Group Theory

Oberwolfach, 29.7. – 4.8.2001

Organizers: G. Hiš (Aachen), D. F. Holt (Warwick), M. F. Newman (Canberra), H. Pahlings (Aachen)

13. ICTMT5 – The Fifth International Conference on Technology in Mathematics Teaching

Klagenfurt, 6.8. – 9.8.2001

The aim of this conference is to bring together classroom practioners, curriculum designers, mathematics education researchers, educational software designers, all of them sharing an interest to improve the quality of teaching and learning by effective use of technology.

The big strands of ICTMT in the past were: The impact of technology on teaching and learning, Access to education through technology, Technology and assessment, Future trends in technology affecting teaching and learning.

The level of education comprises schools, colleges, and unversities. Subjects of interest are: mathematics; related subjects using mathematical models; applications of mathematics in industry and commerce.

Previous sessions of the ICTMT conferences were held at Birmingham 93, Edinburgh 95, Koblenz 97, Plymouth 99.

Scientific Committee: *Honour president:* Adrian Oldknow `a_oldknow@compuserve.com`
Executive presidents: Manfred Borovcnik `manfred.borovcnik@uni-klu.ac.at`,
Hermann Kautschitsch `hermann.kautschitsch@uni-klu.ac.at`

Members: Giulio Cesare Barozzi `barozzi@ciram.ing.unibo.it`, Peter Bender `bender@uni-paderborn.de`, Jaime e Silva Carvalho `jaimecs@mat.uc.pt`, John Cosgrave `john.cosgrave@spd.ie`, Bill Davis `davis@math.ohio-state.edu`, Jan de Lange `j.deLange@fi.uu.nl`, Joachim Engel `engel.joachim@ph-ludwigsburg.de`,
Wolfgang Fraunholz `w.fraunholz@ifm.uni-koblenz.de`, Karl-Josef Fuchs `karl.fuchs@sbg.ac.at`, Dirk Janssens `dirk.janssens@wis.kuleuven.ac.be`, Peter Jones `pjones@swin.edu.au`, Eva-Stina Kllgarden `ekn@mdh.se`, Urs Kirchgraber `kirchgra@math.ethz.ch`, Michalis Kourkoulos `mkourk@edc.uoc.gr`, Bernhard Kutzler `b.kutzler@eunet.at`, Jean Baptiste Lagrange `Lagrange@univ-rennes1.fr`, Vassilis Makrakis `makrakis@softlab.ntua.gr`, Adrian James Oldknow Heinz Schumann `schumann@ph-weingarten.de`, John Searl `j.w.searl@ed.ac.uk`, Jenny Sharp `jsharp@plymouth.ac.uk`, Jose Ramon Vizmanos `jrvizmanos@yalos.com`, Bert Waits `waitsb@math.ohio-state.edu`, Hans-Georg Weigand `hans-georg.weigand@math.uni-giessen.de`, Bernard Winkelmann `bernard.winkelmann@post.uni-bielefeld.de`.

Further information: `http://www.uni-klu.ac.at/ictmt5/`

14. **3rd International ISAAC Congress**

Berlin, 20.8. – 25.8.2001

We would like to make advertisement for the session *Computer Algebra and Computer Analysis* of the 3rd International ISAAC Congress (ISAAC = International Society for Analysis, Applications and Computing; this is **not** the ISSAC!) which will take place in Berlin from 20-25 August 2001, see `http://www.math.fu-berlin.de/~isaac`. The meeting offers sessions in a wide variety of topics primarily circling around applied analysis. Session VII.4 is devoted to *Computer Algebra and Computer Analysis*, and will be organized by Karin Gatermann (Berlin) and Wolfram Koepf (Kassel).

Researchers in analysis often are not familiar with computer algebra or find its use unsuitable for their work. In contrast, we believe that computer algebra can be a very stimulating research tool in analysis. Therefore we would like to put a session program together which is an advertisement for our field, and which presents applications of computer algebra to topics relevant for analysis.

The following speakers have already confirmed to present keynote lectures in our session:

- Liz Mansfield (Canterbury, Great Britain): title to be announced
- Jan Sanders (Amsterdam, The Netherlands): *The classification of integrable evolution equations using number theory*
- Werner Seiler (Mannheim, Germany): *Completion to involution and the numerical integration of general systems of PDEs*

Please contact one of us a.s.a.p. if you are interested to participate and/or to present a lecture in our session. The duration of the lectures will be probably 30 minutes (including discussion), but final decisions will be made later.

Email: Karin Gatermann, Berlin `<gatermann@zib.de>`, Wolfram Koepf, Kassel `<koepf@mathematik.uni-kassel.de>`.

Berichte von Konferenzen

1. **2. Thurnau-Tagung: Computeralgebra in Lehre, Ausbildung und Weiterbildung**

Schloß Thurnau bei Bayreuth, 26. – 28.4.2000

Vom 26. bis zum 28. April 2000 fand im Schloß Thurnau bei Bayreuth die zweite Tagung *Computeralgebra in Lehre, Ausbildung und Weiterbildung* statt, welche von der Fachgruppe Computeralgebra (`http://www.gwdg.de/~cais`) organisiert wurde. Die Tagung hatte 58 Teilnehmerinnen und Teilnehmer, vornehmlich aus Deutschland. Zwei Teilnehmer kamen aus Österreich, drei aus den USA und einer kam aus Weißrußland. Damit war die Beteiligung diesmal fast doppelt so groß wie bei der ersten Thurnau-Tagung 1998. Zu unserer Freude war diesmal die Hälfte der Bundesländer durch offizielle Vertreter beteiligt:

- **Baden-Württemberg:** OStR. Walter Kinkelin (Ministerium), Rolf Reimer (Schulversuch *Pilotprojekt Mobiles Klassenzimmer*)
- **Bayern:** Alois Einhauser (Staatsinstitut für Schulpädagogik und Bildungsforschung), Detlev Kirmse (Zentralstelle für Computer im Unterricht), OStR. Konrad Rudert
- **Brandenburg:** Dr. Götz Bieber (Pädagogisches Landesinstitut Brandenburg)
- **Niedersachsen:** Heiko Knechtel, Hans-Dieter Stenten-Langenbach, Wilhelm Weiskirch
- **Nordrhein-Westfalen:** Dr. Norbert Esper hatte leider aus gesundheitlichen Gründen kurzfristig abgesagt
- **Sachsen:** Jürgen Wagner (Comenius-Institut), Dr. Andreas Meißner (Staatliches Seminar für das Höhere Lehramt an Gymnasien)
- **Sachsen-Anhalt:** Willi Lichtenberg (Landesinstitut für Lehrerfortbildung, Lehrerweiterbildung und Unterrichtsforschung von Sachsen-Anhalt)
- **Schleswig-Holstein:** StD. Dietrich Pohlmann
- **Thüringen:** Dr. Wolfgang Moldenhauer (Thüringer Institut für Lehrerfortbildung, Lehrplanentwicklung und Medien)

Ferner war Landesschulrat Dr. Helmut Heugl aus Österreich vertreten.

In insgesamt 26 Vorträgen wurde über Schulversuche und Initiativen zum Computeralgebra-Einsatz an deutschen Schulen berichtet, wurden Computeralgebrasysteme, computeralgebrafähige Taschen- und Tischrechner sowie interaktive Medien vorgestellt und wurden Erfahrungen mit dem Einsatz von Computeralgebrasystemen im Unterricht weitergegeben. Drei der Vorträge hatten Computeralgebra an Fachhochschule und Universität zum Thema.

Eine komplette Teilnehmerliste findet sich im Internet bei
`http://www.mathematik.uni-kassel.de/~koepf/Thurnau2000.html`.
Die Vorträge werden im Internet veröffentlicht und sind, soweit sie eingereicht wurden, über die Teilnehmerliste elektronisch abrufbar. Das Tagungsprogramm kann bei der Adresse
`http://www.mathematik.uni-kassel.de/~koepf/ProgrammThurnau2000.html`
abgerufen werden.

Der erste Tag stand unter der Überschrift *Computer- und Softwareausstattung an deutschen Schulen*. Zu diesem Thema war die Initiative D21 (`http://www.d21.de`), welche 1999 von IBM in Kooperation mit der deutschen Industrie gegründet wurde, durch die Geschäftsführerin Ariane Alpmann vertreten. Die Initiative *Schulen ans Netz* (`http://www.san-ev.de`), 1996 vom Bundesministerium für Bildung und Forschung und der Deutschen Telekom gegründet, wurde durch den Geschäftsstellenleiter Michael Drabe repräsentiert.

Die Initiative D21 hat sich zum Ziel gesetzt, 20000 deutsche Schulen mit Computern auszustatten. Interessierte Schulen können sich bei der Initiative anmelden. Die Initiative versucht dann, lokale Sponsoren für die Schulen zu finden. Von den bisher 600 angemeldeten Schulen konnten 100 ausgestattet werden.

Die Initiative *Schulen ans Netz* will bis Ende 2001 alle 44000 deutschen Schulen mit einem kostenlosen Internetanschluß versorgen. Bislang haben 17000 Schulen einen Internetanschluß. Zum Vergleich: Während nur 40% der deutschen Schulen Zugang zum Internet besitzen, liegt diese Quote in den USA bereits bei 90%, in Finnland sogar bei 100%.

Bei der Podiumsdiskussion am Ende des ersten Tages wurden folgende Themen als besonders bedeutsam benannt:

- Die Softwareausstattung der Schulcomputer sollte die wichtigsten **Browser und Plugins** enthalten, um Probleme beim Herunterladen aus dem Netz zu vermeiden; dies ist insbesondere unter dem Aspekt zu sehen, daß in den Schulen ja i. a. keine Systemadministratoren vorhanden sind. Ohne diese Softwareausstattung kommt der Computereinsatz schnell zum Erliegen.
- Auf Dauer ist eine Ausstattung mit Schulcomputern nur mit **technischem Betreuungspersonal** zu administrieren. Hierzu müssen die Länder bzw. Kommunen Konzepte entwickeln. In Brandenburg wird eine derartige Konzeption beispielsweise gerade erarbeitet.
- Es sollte mindestens ein **Computeralgebrasystem** zur Verfügung stehen. Hierfür müssen Landeslizenzen angestrebt werden (in einigen Bundesländern gibt es Landeslizenzen) oder aber bundesweite Lizenzen oder Sonderkonditionen für die Schulen ausgehandelt werden.
- Ein wichtiges Thema ist die **Lehrerfortbildung**. Es wurde kontrovers diskutiert, ob die **Fortbildungspflicht** in Niedersachsen das richtige Mittel sei; eine große Mehrheit der Teilnehmer hält dies aber doch für angemessen. Nachdem die Computeralgebra-Fortbildung zuerst teilweise mißmutig aufgenommen wurde, sind die Reaktionen inzwischen eher positiv. In Sachsen ist seit langem der Einsatz von graphikfähigen Taschenrechnern möglich, aber erst seit der **obligatorischen** Einführung für alle Schulen ab Klasse 8 konnte diese Konzeption auch landesweit umgesetzt werden. Es ist eine bekannte Erfahrung, daß über Kann-Regelungen lediglich ein Drittel der Lehrer ansprechbar ist.
- Erst wenn Computeralgebrasysteme auch in der **Prüfungssituation** zugelassen sind, ist der flächendeckende Einsatz möglich.

Der Rest der Konferenz stand unter dem Schwerpunkt des Einsatzes von Computeralgebra in den einzelnen Bundesländern. Es ergab sich folgender aktueller Stand:

- **Baden-Württemberg** (`http://www.uni-karlsruhe.de/~za242/CAS`): Im Rahmen des PiMoKl-Schulprojekts (*Pilotprojekt Mobiles Klassenzimmer*), an welchem seit 1996 fünf Gymnasien ab Klasse 11 beteiligt waren, wurden 1999 zum ersten Mal (mit *eigenem* Zentralabitur)

Abiturprüfungen abgelegt (Computeralgebra-Rundbrief 24, S. 15 und Rundbrief 25, S. 17–18, s. `http://www.gwdg.de/~cais`, bzw. ZEIT vom 6. Mai 1999). Alle beteiligten Schulen sind weiterhin an diesem Projekt beteiligt, und es sind inzwischen drei neue Schulen hinzugekommen.
Ansprechpartner: OStR. Walter Kinkelin (`kinkelin@km-bw.de`).

- **Bayern** (`http://www.zs-augsburg.de`):
Seit 1992 werden graphikfähige Taschenrechner eingesetzt, s. *Forum Graphische Taschenrechner* (`http://www.zs-augsburg.de/rs/fgtr/fgtr.htm`). Dies betrifft ca. 8 % der Schulen, auf Grund des Zentralabiturs sind allerdings keine Gymnasien beteiligt. Seit 1996 gibt es erste Versuchseinsätze symbolischer Taschenrechner. Seit 1999 sind an diesem Schulversuch 350 Schüler aus 13 Klassen, hauptsächlich aus Realschulen, beteiligt.
Ansprechpartner: Detlev Kirmse (`kirmse@zs-augsburg.de`).
- **Brandenburg** (`http://www.uni-potsdam.de/u/PLIB`): Es gibt seit 1999 die Medienoffensive *MAUS* (Medien an unseren Schulen), da Brandenburg beim Einsatz von Medien in der Schule unter 16 Bundesländern einen schlechten 15. Platz einnimmt. In diesem Rahmen werden in den nächsten 5 Jahren Mittel bereitgestellt für eine weitere Ausstattung mit Hard- und Software, z. B. auch für Landeslizenzen. Es ist eine breit angelegte Fortbildung vorgesehen, und es sollen Systemadministratoren ausgebildet werden. Schließlich wird im Laufe des Frühjahrs 2000 ein neuer Rahmenplan für den Mathematikunterricht der Sekundarstufe I erstellt, in welchem moderne Medien eine größere Rolle spielen sollen. Beispielsweise soll dynamische Geometriesoftware verbindlich eingesetzt werden, aber auch an den propädeutischen Einsatz von Computeralgebrasoftware ist gedacht.
Ansprechpartner: Dr. Götz Bieber (`goetz.bieber@plib.brandenburg.de`).
- **Mecklenburg-Vorpommern**: Im Ministerium werden momentan Vorbereitungen getroffen, um den Einsatz von Computeralgebrasystemen im Abitur ab 2002 möglich zu machen. Das Abitur wird sich dann aus einer Pflichtaufgabe für alle Schüler, bei dem CAS- Kundige kein en Vorteil haben werden, und einem Wahlteil zusammensetzen. Der Wahlteil besteht aus je 4 Aufgaben, aus d enen Schüler zwei auszuwählen haben und zwar aus einem Wahlteil für Schüler mit CAS und aus ein em ohne.
Ansprechpartner: Dr. Jochen Weitendorf (`dr.weitendorf@t-online.de`).
- **Niedersachsen**: Auf Grund des dezentralen Abiturs werden vor allem der TI92 und der TI89 bereits von recht vielen Lehrern bis hin zum Abitur eingesetzt. Seit 1999 gibt es eine Lehrerfortbildungsoffensive mit Fortbildungspflicht für Mathematiklehrer, s. Computeralgebra-Rundbrief 25, S. 22–24.
Ansprechpartner: Heiko Knechtel (`HKnechtel@aol.com`).
- **Nordrhein-Westfalen:** (`http://www.learn-line.nrw.de/angebote/cas/medio.htm`): Seit 1998 gibt es einen Schulversuch, bei welchem von knapp 30 Klassen im Rahmen des Mathematikunterrichts der gymnasialen Oberstufe DERIVE bzw. der TI-92 eingesetzt wird.
Ansprechpartner: Rainer Opitz (`rainer.opitz@mail.lsw.nrw.de`).
- **Sachsen** (`http://www.sn.schule.de/~ci`): Sachsen ist das einzige Bundesland, in welchem graphikfähige Taschenrechner (seit 1997) verbindlich ab Klasse 8 vorgesehen sind. Im Jahr 2000 ist dieses Projekt in der Abiturstufe angekommen und werden die ersten Abiturprüfungen unter Einbeziehung von graphikfähigen Taschenrechnern gestellt. Es gibt einen Modellversuch zum Einsatz von Tabellenkalkulation, Geolog und MathCad in Klassen 7–10, eine Fortsetzung in die Oberstufe ist vorgesehen.
Ansprechpartner: Jürgen Wagner (`wagner.comenius@t-online.de`).
- **Sachsen-Anhalt** (`http://server1.schule.uni-halle.de/~lisa/homepag2.htm`): Taschenrechner und Tabellenkalkulation gibt es ab Klasse 7, graphikfähige Taschenrechner und Computeralgebrasysteme sind erlaubt, aber nicht im Abitur zugelassen. Ein erster Schulversuch wurde 1998 etabliert mit einem Grund- und einem Leistungskurs (TI89) sowie einem weiteren Leistungskurs (CASIO Algebra FX2.0).
Ansprechpartner: Willi Lichtenberg (`Wlichtenberg@lisa.mk.lsa-net.de`).
- **Thüringen** (`http://www.thillm.th.schule.de`):
Seit 1997 wurden an 8 Schulen die zehnten Klassen mit symbolischen Taschenrechnern TI89 ausgestattet. Diese Ausstattung wurde weitergeführt und ab 2002 ist dann Zentralabitur mit dem TI89 möglich. Die beteiligten Lehrer treffen sich alle zwei Monate.
Ansprechpartner: Dr. Wolfgang Moldenhauer (`WMoldenhauer@thillm.thueringen.de`).

In der abschließenden Podiumsdiskussion wurde debattiert, wie der Einsatz von Computeralgebrasystemen in den Schulen weiter vorangebracht werden kann. Es wurde insbesondere festgestellt, daß sich an den Schulen in den letzten Jahren zwar einiges getan hat, daß aber der Funke noch nicht recht auf die Universitäten und die dortige **Lehrerausbildung** übergesprungen ist. Auf der anderen Seite gibt es an einigen Universitäten durchaus ein entsprechendes Angebot (s. die regelmäßige Rubrik *Lehrveranstaltungen zu Computeralgebra* des Rundbriefs), welches aber von Staatsexamenskandidaten nur zögerlich angenommen wird. Hier sollte man die **Prüfungsinhalte des Staatsexamens** überdenken und entsprechend anpassen. Als Minimalforderung sollte die Staatsexamensausbildung wenigstens die Schulinhalte abdecken, und dazu gehören heutzutage auch Computeralgebrakenntnisse. Es wurde eingeschätzt, daß insbesondere die zwei Drittel der eher weniger motivierten Lehrer nicht eingebunden werden können, so lange die Universitätsausbildung stagniert. Als Nahziel wurde die Forderung formuliert, die 10%-Marke zu überspringen. Schließlich wurde auf allgemeinen Wunsch beschlossen, die Tagung in zwei Jahren fortzusetzen.

Mein persönlicher Eindruck, welcher durch unzählige Gespräche bestätigt wurde, war der, daß die Tagung allen Beteiligten einen sehr guten Überblick verschafft hat über den momentanen Stand des Computeralgebra-Einsatzes an deutschen Schulen.

Wolfram Koepf (Kassel)

2. ISSAC 2000 – International Symposium on Symbolic and Algebraic Computation

St Andrews, Schottland, 7.8. – 9.8.2000

Näheres zu der Tagung und der ISSAC-Serie findet man auf der URL:
`http://www-history.mcs.st-and.ac.uk/issac2000/`.

Bereits traditionell fanden am Vortag der Konferenz drei Tutorien zu aktuellen Themen der Computeralgebra statt:

Martin Escardo, *Exact Numerical Computation*

Alexander Hulpke, *Using GAP*

Mohamed Omar Rayes, *Application of Computer Algebra in Mathematics Education*

Die diesjährige ISSAC war eng mit der CALCULEMUS 2000 (8th Symposium on the Integration of Symbolic Computation and Mechanized Reasoning) verbunden. Die Gesamtzahl der Teilnehmer beider Tagungen betrug 160. Als gemeinsame Veranstaltung beider Kongresse fand der eingeladene Hauptvortrag

Henk Barendregt, Arjeh M. Cohen, *Representing and handling mathematical concepts by humans and machines*

statt. Weitere eingeladene Hauptvorträge der ISSAC 2000:

Felipe Cucker, *Solving Polynomial Systems: a Complexitya Theory Viewpoint*

Derek F. Holt, *Computation in Word-Hyperbolic Groups*

Die Autoren der 38 in 12 (sequentiellen) Sitzungen vorgestellten Beiträge waren: M. Zhang und R. Goldman; A. Chtcherba und D. Kapur; C. W. Brown; P. Lisonek und R. Israel; A. Dolzmann und V. Weispfenning; G. Ivanyos; W. Eberly; T. Mulders und A. Storjohann; J.-G. Dumas, B. D. Saunders und G. Villard; V. Pan; S. A. Abramov und M. Bronstein; A. Gupta, P. Rohatgi und R. Agarwal; C.-P. Jeannerod; A. C. Norman; J. Harris; N. Hur und J. H. Davenport; H. Cheng und E. Zima; G. E. Collins und W. Krandick; B. Mourrain und Ph. Trebuchet; G. Bodnar und J. Schicho; F. Binder, E. Aichinger, J. Ecker, P. Mayr und Ch. Nobauer; G. Lecerf; J. Schicho; G. Landsmann, E. Hillgarter, J. Schicho und F. Winkler; D. Grigoriev und N. Vorobjov; M. Noecker; I. Miyamoto; P. Fernandez-Ferreiros und M. Gomez-Molleda; H. Anai und V. Weispfenning; A. Fredet; G. J. Reid und A. D. Wittkopf; F. Boulier und F. Lemaire; J. C. Rosales, P. A. Garcia-Sanchez und J. I. Garcia-Garcia; O. Cormier, M. F. Singer und F. Ulmer; J. Abbott, V. Shoup und P. Zimmermann; M. Monagan und A. D. Wittkopf; Y. Huang, H. J. Stetter, W. Wu und L. Zhi; E. Kaltofen, W. Lee und A. A. Lobo.

Darüberhinaus wurden 21 Poster vorgestellt und es fanden 4 Systemdemonstrationen statt.

Joachim Apel (Leipzig)

3. 2. MuPAD Workshop 2000

Paderborn, 8.9. – 9.9.2000

Dieser Workshop war einerseits dem zehnjährigen Jubiläum des Computeralgebrasystems MuPAD als auch der neuen Release MuPAD 2.0 gewidmet. Einige Übersichtsvorträge zum Thema Computeralgebra von

- Michael Wester: *Some Perspectives on the Usability of Computer Algebra Systems*
- Wolfram Koepf: *Computeralgebraeinsatz in Deutschland*

- Thomas Himmelbauer: *Computeralgebraeinsatz in Österreich*
- Wolfram Luther: *Hypermediales Lehren und Lernen*

und ein MuPAD-Kurs rundeten das Programm ab. Das Tagungsprogramm richtete sich in erster Linie an Lehrer mit dem Interesse, Computeralgebra im Schulunterricht einzusetzen.

Die folgenden Vorträge beschäftigten sich mit Komponenten von MuPAD 2.0:

- Oliver Kluge: *Introduction to MuPAD 2.0*
- Ralf Hillebrand: *MuPAD 2.0 User Interfaces and Documentation*
- Walter Oevel: *MuPAD 2.0: The New Library*
- Klaus Drescher: *Kernel News - Changes in the MuPAD Language and Kernel for Version 2.0*
- Steve Swanson: *The Scientific WorkPlace Computational Interface (Powered by MuPAD)*
- Claude Gomez: *Computer Algebra and Numerical Computations: the MuPAD/Scilab link*
- Frank Postel: *MuPAD in der Lehre: Aktuelle und zuknftige Entwicklungen*
- Andreas Sorgatz: *MuPAD als Rechenkomponente im Web*
- Eno Tonisson: *Mupad's Capability Of Solving School Algebra Problems*

Homepage der Tagung mit Teilnehmerliste bei `http://www.mupad.de/mw2000/index.shtml`.

Wolfram Koepf (Kassel)

4. **Jahrestagung der DMV – Sektion "Computeralgebra"**

Dresden , 18.9. – 22.9.2000

Die Jahrestagung der DMV fand in diesem Jahr in Dresden vom 18. bis zum 22. September statt. Auch diesmal gab es eine eigenständige Sektion "Computeralgebra", die von B. H. Matzat (Heidelberg) und M. Geck (Lyon) geleitet wurde.

In 5 Übersichtsvorträgen wurde über vielfältige Aspekte und Anwendungsbereiche der Computeralgebra berichtet: das Spektrum reichte von dünnbesetzten Gleichungssystemen in der Chemie (K. Gatermann, Berlin), der konstruktiven Galoistheorie (J. Klüners, Heidelberg) über die konstruktive Erkennung von Matrixgruppen (G. Hiß, Aachen) und das "Modular Atlas Project" (J. Müller, Aachen) bis hin zum Parametrisierungsproblem für algebraische Flächen (J. Schicho, Linz).

In 5 weiteren Vorträgen wurden folgende Themen behandelt: Der PCS-Beweiser im Rahmen des Theorem-Projektes (B. Buchberger, Linz), das Computer-Algebra Paket ACE (M. Geck, Lyon), der Bahnalgorithmus in der Darstellungstheorie (M. Neunhöffer, Aachen), die effektive Darstellung von auf semialgebraischen Kompakta positiven Polynomen (M. Schweighofer, Konstanz) und ein schneller Resolutionsalgorithmus für Kurvensingularitäten (P. Stadelmayer, Linz).

Mit ingesamt 10 Vorträgen lag diese Sektion etwas unter dem Sektionsdurchschnitt von ungefähr 16 Vorträgen (gemittelt über alle Sektionen). Nichtsdestoweniger waren einige Vorträge sehr gut besucht und lösten lebhafte Diskussionen aus. Bemerkenswert war auch der (äusserst professionelle!) Einsatz von Computer-Vorführungen bei den Vorträgen. Dadurch konnten sich die Zuhörer ein genaueres Bild auch von tatsächlichen Implementationen der vorgestellten Algorithmen machen.

M. Geck (Lyon)

Neues über Systeme und Hardware

SINGULAR
A Computeralgebra-System for Polynomial Computations

G.M. Greuel, G.Pfister, H. Schönemann
Fachbereich Mathematik; Universität Kaiserslautern
67653 Kaiserslautern; Germany

Singular is a specialized computer algebra system for polynomial computations with emphasize on the needs of commutative algebra, algebraic geometry, and singularity theory.

Main functionality.

Singular's main computational objects are polynomials, ideals and modules over a large variety of rings. Singular features one of the fastest and most general implementations of various algorithms for computing standard resp. Gröbner bases. Furthermore, it provides multivariate polynomial factorization, resultant, characteristic set and gcd computations, syzygy and free-resolution computations, numerical root–finding, visualisation, and many more related functionalities.

Based on an easy-to-use interactive shell and C-like programming language, Singular's internal functionality is augmented and user-extendable by libraries written in the Singular programming language or in C++. A general and efficient implementation of links as endpoints of communications allows Singular to make its functionality available to and be easily incorporated into other programmes.

The main goal of the Singular–group is to further develop and implement *advanced* algorithms to be used for mathematical research, in particular in commutative algebra, algebraic geometry and singularity theory.

Indeed there exist already several libraries providing such algorithms, including full primary decomposition for several ground fields, ring normalization (integral closure), versal deformations of arbitrary isolated singularities, monodromy and spectral numbers for hypersurface singularities, Hamburger–Noether (Puiseux)–expansions of plane curve singularities and many more. Most of these algorithms are not available in any other system.

Recently, due to nonmathematical applications of Singular, we are experimenting with symbolic–numerical polynomial solving.

As a specialized system, Singular's aim is not to provide all the functionality of a general purpose system. The main strength of the system, besides the above mentioned functionality, is the speed of the important basic algorithms such as Gröbner basis, syzygy, and free resolution computations for modules. It is impossible to detail any of the algorithms, we rather refer to the literature, given in the references.

Singular's online help system is available in various formats where the HTML format is especially user-friendly.

Availability.

Singular is publicly available as a binary program for all common Unix platforms including Linux, for Windows 95/98/NT and for MacOS.
The current version number is 1.3.8. It can be downloaded by anonymous ftp from
`ftp://www.mathematik.uni-kl.de/pub/Math/Singular` or per WWW from
`http://www.singular.uni-kl.de/distribution.html`

Besides the executable Singular program, the distribution contains the source code of all Singular libraries, the user manual (resp. tutorial) in various formats (PostScript, info, and HTML) and some utility programs (for visualisation etc.).

Moreover, on request, we provide the source code of the Factory library for multivariate polynomial gcd, resultant, and factorization which is part of Singular but may be used by other systems (as it is e.g., by Macaulay 2).

For more and always up-to-date information, Singular's home page can be reached at
`http://www.singular.uni-kl.de`

Mathematical Features.

Singular's primary computational objects are ideals resp. modules which are generated by polynomials resp. polynomial-vectors over polynomial rings or, more generally, over the localization of a polynomial ring with respect to any ordering on the set of monomials which is compatible with the semigroup structure.
Supported baserings include:

- polynomial rings with a large variety of polynomial orderings (common simple, block, elimination, weighted, and general matrix orderings),
- localization of a polynomial ring at a prime ideal generated by a subset of the variables
- factor rings by an ideal of one of the above,
- rings of tensor products of one of the above.

Supported ground fields for these rings include:
- rational numbers (of arbitrary length) Q,
- finite fields Z/p (where p is a prime ≤ 32003),
- Galois fields (finite fields with $q = p^n \leq 2^{15}$ elements),
- transcendental extensions ($K(A, B, C, \ldots)$, $K = Q$ or Z/p),
- algebraic extensions ($K[t]$/minimal-polynomial, $K = Q$ or Z/p), and
- floating point real and complex numbers with arbitrary predefined precision.

The main algorithms implemented in Singular are:
- general standard basis algorithm for *any* monomial ordering which is compatible with the natural semi-group structure of the exponents. This includes well-orderings (Buchberger algorithm) and tangent cone orderings (Mora algorithm) as special cases,
- Hilbert–driven Gröbner basis algorithms, weighted–ecart–method and high–corner–method, FGLM algorithm for change of ordering,
- factorizing Buchberger algorithm,
- intersection, quotient, elimination and saturation of ideals,
- Schreyer's, La Scala's and Siebert's algorithm for computations of syzygies and free resolutions of modules,
- combinatorial algorithms for computations of dimensions of factor rings, Hilbert series and multiplicities of modules,
- multivariate polynomial gcd, resultant, and factorization algorithms,
- Wang's algorithm to compute characteristic sets.

See [GGM], [GP96], [GPS98], [GP98] for more details on the implemented algorithms.

Computational Features.

Singular has a convenient and intuitive interactive user interface (shell) which has key-bindings similar to those of Unix' `tcsh` shell. Alternatively, an Emacs mode provides the possibility to run Singular within an Emacs/XEmacs window. Singular's user interface provides both, access to Singular's mathematical functionality and a convenient, powerful, and C-like programming language (strongly typed, and lexicographically scoped) which includes all the usual programming constructs (like loops, procedures, local/global variables, etc). Based on this programming language, users may extend Singular's functionality by writing their own libraries. Libraries may be written in the Singular programming language or in C++.

At the moment, the Singular distribution includes the following libraries:

standard.lib	extensions of Singular kernel
General purpose	
all.lib	load all other libraries
general.lib	procedures of general type
inout.lib	procedures for manipulating in- and output
poly.lib	procedures for manipulating polynomials and ideals
random.lib	procedures of random/sparse matrix and poly operations
ring.lib	procedures for manipulating rings and maps
Linear algebra	
matrix.lib	procedures for matrix operations
jordan.lib	procedures to compute the jordan normal form
linalg.lib	procedures for algorithmic linear algebra

Commutative algebra

algebra.lib	procedures for computing with algebras and maps
elim.lib	procedures for elimination, saturation and blowing up
homolog.lib	procedures for homological algebra
mregular.lib	procedures for Castelnuovo-Mumford regularity
normal.lib	procedures for normalization
primdec.lib	procedures for primary decomposition
primitiv.lib	procedures for finding a primitive element
intprog.lib	procedures for integer programming
toric.lib	procedures for toric ideals

Singularities

classify.lib	procedures for the Arnold-classifier of singularities
deform.lib	procedures for computing miniversal deformation
hnoether.lib	procedures for Hamburger-Noether (Puiseux) development
mondromy.lib	procedures to compute the monodromy of a singularity
sing.lib	procedures for computing invariants of singularities
spcurve.lib	procedures for Cohen–Macaulay codimension 2 singularities

Invariant theory

finvar.lib	procedures to calculate invariant rings of finite groups
ainvar.lib	procedures for invariant rings of the additive group

Symbolic-numerical solving

presolve.lib	procedures for pre-solving polynomial equations
solve.lib	procedures to solve polynomial systems
triang.lib	procedures for decomposing zero-dimensional ideals
ntsolve.lib	procedures for real Newton solving

Visualization

graphics.lib	procedures to draw with Mathematica
latex.lib	procedures for typesetting in TeX
surf.lib	interface to the surf programme

Links.

Singular furthermore features links as general endpoints of communications, i.e. as something, Singular can read from or write to. To this point, the following link types are implemented:

Ascii text	Output can conveniently be viewed and manipulated. Read/write is not the fastest.
DBM	Provides access to data stored in a data base.
MP file	Stores data in the binary Multi Protocol (MP) format [BGS96]. Read/write is very fast.
MP TCP	Exchanges data in binary MP format between processes (on the same or different computers); data exchange is very efficient

The functionality of theses links is provided to the user by a general, consistent and convenient link interface.

Based on MP TCP links, Singular can very efficiently communicate with itself which opens the door for implementations of parallel/distributed algorithms. Furthermore, the same links can be used to communicate with other Computer Algebra programs which have an MP interface. At the moment, there are MP interfaces for MuPAD and Mathematica, enabling the use of one system from the others (e.g. one can use Singular's functionality from within Mathematica or MuPAD).

Performance.

Singular's kernel is implemented in C/C++. Singular's main implementation design goal is speed of the mainly used algorithms. Therefore, all time-consuming operations like standard bases computations or factorization are implemented in its kernel. As another consequence, Singular has the concept of a global ring which needs to be defined prior to any polynomial operations. Arbitrary precision integer and floating point arithmetic is accomplished by linking Singular with the GNU multiple precision library `gmp` and modulo arithmetic is accomplished by using look-up tables. Polynomials are internally represented as linked lists of monomials, where a monomial consists of a coefficient and an exponent vector.

To illustrate Singular's computing speed, the table below shows timings of various systems for solving problem 6 of the ISSAC'97 system challenge (computation of a lexicographical Gröbner basis, 16 roots). The first row shows the timings for the Gröbner basis computation (using the default command as the respective manual describes them). The second row shows the time for numerical solving (finding all complex roots) of the same system (commands `NSolve` (Mathematica), `evalf(solve(..))` (Maple) and `triangL_solve` (Singular)). (The new symbolic-numerical algorithms for polynomial system solving in Singular seem to be especially promising, cf. [G00-2]). All timings were taken on a Pentium Pro 200 with 128 MB of RAM running Linux:

	Mathematica 4.0	Maple V.5.1	GB v3	SINGULAR
Gröbner basis	159.49 sec	14.66 sec	207 sec	0.61 sec
numerical solution	15.91 sec	out of mem.	-	4.28 sec

Computations with MuPAD 1.4, Reduce, CoCoA (3.0.2) and Macaulay 2 (0.8.14) could not finish the Gröbner basis computation within 15 CPU hours.

For more examples see: `http://www.symbolicadata.org`, a project benchmarking CA system.

SINGULAR 2.0.

The new version, to be released at the end of 2000, has a new internal engine for polynomial arithmetic which dramatically increases its computational efficiency while at the same time decreases its memory usage.

ESingular is a new program for out–of–the–box, precustomized Emacs/ XEmacs which runs Singular. Hence, using ESingular, such features as colour input–output highlighting, pull–down menus, line truncation, a demo mode and tab completion are provided. Singular 2.0 has also a new Windows port which includes ESingular.

Future Work.

Singular's development is an actively ongoing project. Currently, the following features are under development:

- more flexible ring concepts
- faster computations by better adaption of datastructures to algorithms
- Newton polyhedron algorithms
- integral closure of an ideal
- primary decomposition for modules

Furthermore, an independent version Singular::Plural is under development which is able to compute standard bases and syzygies in very general *non–commutative* structures with applications especially for Lie algebras.

References.

[BGS96] O. Bachman, S. Gray, and H. Schönemann. MP: A Frame work for Distributed Polynomial Systems Based on MP. In *Proc. of the International Symposium on Symbolic and Algebraic Computation (ISSAC'96)*, Zurich, Switzerland, July 1996, ACM Press.

[GGM] H. Grassmann, G.-M. Greuel, B. Martin, W. Neumann, G. Pfister, W. Pohl, H. Schönemann, and T. Siebert. On an implementaion of standard bases and syzygies in Singular. *Computational methods in Lie Theory.* AAECC, 7:235-249, 1996.

[GP96] G.-M. Greuel and G. Pfister. Advances and improvements in the theory of standard bases and syzygies. *Arch. d. Math.*, 66:163-176, 1996

[GP98] G.-M. Greuel and G. Pfister. Groebner bases and algebraic geometry. In B. Buchberger and F. Winkler, editors, *Groebner bases and applications*, volume 251 of *London Mathematical Society Lecture Notes*, pages 109-143, Cambridge University Press, 1998

[GPS98] G.-M. Greuel, G. Pfister, and H. Schönemann. Singular version 1.2 User Manual. In *Reports On Computer Algebra*, number 21, Center for Computer Algebra. University of Kaiserslautern. June 1998.

[GR00-1] G.-M. Greuel. Computeralgebra and Algebraic Geometry Achievements and perspectives. To appear in JSC, 2000

[G00-2] G.-M. Greuel. Applications of Computer Algebra to Algebraic Geometry, Singularity Theory and Symbolic-Numerical Solving. To appear in *Proceedings of the 3rd ECM, Barcelona 2000*

G.M. Greuel, G.Pfister, H. Schönemann (Kaiserslautern)
`[greuel,pfister,hannes]@mathematik.uni-kl.de`

Mathematica, die vierte

Im Herbst 2000 über die Änderungen in *Mathematica* 4 zu schreiben scheint schon fast nicht mehr der Mühe wert, denn schon in Kürze wird 4.1 verfügbar sein. Nichtsdestotrotz lohnt sich eine Übersicht, denn viele der hier erwähnten Punkte werden auch auf 4.1 zutreffen.

Sicherlich eine der wichtigsten Neuerungen von *Mathematica* 4 im Vergleich zur Version 3 ist die Geschwindigkeit bei numerischen Berechnungen. Der Geschwindigkeitszuwachs ist vor allem auf die interne Verwendung von sog. packed arrays zurückzuführen. Man stelle sich ein packed array grob wie ein normales Array in einer Hochsprache wie C vor, d.h. es enthält nur die Daten und nicht für jedes Element den kompletten Overhead eines normalen *Mathematica*-Objekts. Das macht die packed arrays nicht nur einen Faktor 4–5 kompakter im Speicherbedarf, sondern sie sind auch besser an bestehende Datenstrukturen angepaßt, z.B. werden intern viele Funktionen aus Standardlibraries direkt mit packed arrays als Argumenten aufgerufen. Dazu kommt noch, daß im Vergleich zur Version 3 viel mehr Funktionen automatisch vorkompiliert werden. So wurden bislang die Argumente von Funktionen wie `Plot` implizit kompiliert, während in *Mathematica* 4 z.B. auch Funktionen, die auf Listen losgelassen werden, automatisch kompiliert werden. Mit der packed-array-Technologie ist *Mathematica* 4 bei Matrizenoperationen etwa so schnell wie Matlab geworden. Einige Vergleichswerte für eine 500×500-Matrix mit reellen Zufallszahlen (`m = Table[Random[], {500}, {500}]`):

Ausdruck	*Mathematica* 3	*Mathematica* 4	Geschwindigkeitszuwachs
`Sin[m]`	2.433 s	0.13 s	18.7
`(m + 1)^100`	4.425 s	0.311 s	14.23
`Min[m]`	4.487 s	0.02 s	224

Eine Liste weiterer Benchmarks findet man z.B. bei http://smc.vnet.net/mathbench.html.

Jedoch nicht nur numerische Berechnungen sind schneller worden. Beim pattern matching, einem integralen Bestandteil der *Mathematica* Sprache, hat Version 4 ebenfalls zugelegt (allerdings nicht so sehr wie im numerischen Bereich), und das wirkt sich in erster Linie auf symbolische Berechnungen aus.

Das grafische Frontend hat in *Mathematica* 4 mehr Funktionen bekommen bzw. vom Kernel übernommen, und diese Tendenz wird sich fortsetzen. Es ist abzusehen, daß der Kernel in zukünftigen Versionen praktisch keine Grafikfähigkeiten mehr besitzen wird und das Frontend diese Funktionen komplett übernimmt, d.h. der Kernel wird dann automatisch zur Grafikausgabe das Frontend aufrufen. (Für Kernel-Puristen: keine Angst, das geschieht intern, ist also nicht am Bildschirm sichtbar.) Die eingebauten Renderingfunktionen des Kernel (`Display`, `Export`) sind seit Version 3 praktisch nicht mehr geändert worden, was sich in der jetzigen Version leider etwas unangenehm darin bemerkbar macht, daß Programme z.T. unterschiedlichen Grafik-Output erzeugen, je nachdem, ob sie aus dem Frontend oder dem Kernel heraus gestartet werden.

Neu in Version 4 sind auch die `Import`- und `Export`-Funktion, mit der sich viele gängige Dateiformate mühelos von und nach *Mathematica* transferieren lassen. Z.B. um eine Tabelle mit Daten einzulesen, genügt

```
Import["file.dat", "Table"]
```

Besonders überzeugend gelöst ist der Export von Frontend-Notebooks in HTML- und TeX-Code, was vermutlich auf das Bestreben von Wolfram Research zurückzuführen ist, mit den Notebooks ein universelles Format für mathematische Dokumente zu schaffen, das eine möglichst weite Verbreitung finden soll. In Version 4.1 wird die Liste der unterstützten Ein- und Ausgabeformate noch um einige erweitert sein, insbesondere wird 4.1 in der Lage sein, MathML-Code zu erzeugen. [MathML ist eine Erweiterung von HTML, die mathematischen Formelsatz einschließt und in der nächsten Generation von Browsern, z.B. Netscape 6, verfügbar sein soll.]

Eine kleine, aber wichtige Erweiterung ist die Möglichkeit, bei `Simplify` und `FullSimplify` einschränkende Annahmen anzugeben, die bei der Vereinfachung berücksichtigt werden, z.B.

```
Simplify[Sqrt[a^2]] -> Sqrt[a^2]

Simplify[Sqrt[a^2], Element[a, Reals]] -> Abs[a]
```

Mathematica dürfte zur Zeit auch eine der vollständigsten Sammlungen von Integralen und speziellen Funktionen besitzen (für eine Übersicht der Fähigkeiten verschiedener Systeme siehe z.B. http://www.scientificweb.c
Als kleiner Wermutstropfen muß allerdings auch erwähnt werden, daß *Mathematica* in vielen Fällen immer noch opportune Parameter annimmt, etwa in

```
Integrate[x^n, x] -> x^(1 + n)/(1 + n)
```

obwohl es eigentlich in der Lage ist, Fallunterscheidungen durchzuführen.

Einen ganz neuen Bereich von Anwendungen erschließt das *MathLink* API, mit der sich Hochsprachen-Programme in C, C++, Fortran, Perl und Visual BASIC anbinden lassen. Obwohl *MathLink* nicht erst seit Version 4 existiert, war es bis Version 3 nur in der Lage, C-Programme anzubinden. Ein auf *MathLink* basierendes Zusatzpaket ist J/Link, das man kostenlos bei http://www.wolfram.com/news/jlink.html bekommt. J/Link verknüpft *Mathematica* mit Java. Es ist extrem simpel zu benutzen und stellt auf einen Schlag alle bestehenden Java-Klassen in *Mathematica* zur Verfügung. Um z.B. ein Fenster auf dem Bildschirm zu öffnen:

```
<< JLink`
InstallJava[]
frame = JavaNew["java.awt.frame"]
```

Das ist vor allem deswegen so nützlich, weil *Mathematica* zwar hervorragende Grafik-*Ausgabe*routinen besitzt, aber zur Eingabe im wesentlichen auf die Tastatur beschränkt ist.

Abschließend sollte man erwähnen, daß die Unterschiede zwischen Version 3 und 4 zwar wichtige Verbesserungen darstellen, aber längst nicht so drastisch sind wie zwischen 2.2 und 3. Bei letzterem Upgrade mußte man in vielen Fällen bestehende Programme umzuschreiben, und das ist beim Umstieg von 3 auf 4 nur in Ausnahmefällen nötig.

Thomas Hahn (Karlsruhe, hahn@feynarts.de)

Computeralgebra auf Taschenrechnern

Wir haben nun einige Informationen über die Computeralgebra-Fähigkeiten von Taschenrechnern zusammengetragen und diese unter dieser Rubrik gesammelt.

Zunächst stellen die Hersteller TI und CASIO ihre Rechner vor, und danach folgt ein vergleichendes Referat von Prof. Dr. Hans-Wolfgang Henn.

Einige fortgeschrittene CAS Fähigkeiten der Taschencomputer TI-89 and TI-92 Plus

TI-89 und TI-92 Plus sind leistungsstarke Taschencomputer mit ernsthaften programmiertechnischen, graphischen, numerischen und symbolischen Fähigkeiten. Zum Beispiel gewann der Vorgänger TI-92 den Wettbewerb während des International Symposium on Symbolic and Algebraic Computation gegen REDUCE, MACSYMA, Mathematica, und MuPad. (Wegen der Aufgaben und Lösungen siehe das SIGSAM Bulletin Ausgabe 122 und 123, veröffentlicht von der Association for Computing Machinery.)

Im folgenden einige Beispiele zu den mehr fortgeschrittenen CAS Fähigkeiten von TI-89 und TI-92 Plus formuliert als Aufgaben:

1. Finde (in höchstens 1 Sekunde) die allgemeine symbolische Lösung der nicht-linearen Differentialgleichung zweiter Ordnung $xyy'' + xy'^2 + yy' = 0$.

2. Gib (in höchstens 8 Sekunden) alle allgemeinen Lösungen des folgenden Systems nicht-linearer algebraischer Gleichungen in Abhängigkeit des Parameters k:

$$\begin{aligned} wxyz &= k\,, \\ wxy + xyz + yzw + zwx &= 0\,, \\ wx + xy + yz + zw &= 0\,, \\ w + x + y + z &= 0\,. \end{aligned}$$

3. Berechne (in höchstens 2 Sekunden) den folgenden symbolischen Grenzwert exakt:

$$\lim_{x\to\infty} e^x \cdot \left(e^{1/x+e^{-x}+e^{-x^2}} - e^{1/x-1/e^{e^x}} \right) .$$

4. Berechne (in höchstens 8 Sekunden) die folgende Stammfunktion exakt:

$$\int \left(x^2 \cdot \Big(\cos(b \ln x) + \arcsin x \Big) + \frac{x^{1/3}}{(a + x^{4/5})^{2/3}} \right) dx \,.$$

5. Berechne (in höchstens 8 Sekunden) die symbolische Determinante der symmetrischen 10×10 Tri-Diagonalmatrix mit $x^2 + 1$ in der Diagonalen und x in der oberen und unteren Nebendiagonalen.

6. Multipliziere (in höchstens 9 Sekunden) vollständig aus: $(x + y)^{100}$.

7. Faktorisiere (in höchstens 1 Sekunde) über den ganzen Zahlen $x^{100} + y^{100}$.

David R. Stoutemyer
`dstoutemyer@ti.com`

Der CAS-Graphikrechner Algebra FX 2.0 von Casio

Das Maß aller bisherigen CAS-Rechner im Taschenformat ist der TI 92 bzw. der TI 89. Welche Alternative dazu stellt der Casio Algebra FX 2.0 dar?

Neben den vielen schon bekannten numerischen Möglichkeiten, die man vom Modell CFX 9850BG Plus schon kennt, findet man in der Menüliste zusätzlich die Menüs CAS, Algebra und Tutor, hinter denen sich ein Computeralgebrasystem verbirgt. Die symbolischen Rechenmöglichkeiten sind den Themen der Sekundarstufe II zuzuordnen. Neben algebraischen Termumformungen wie expand, factor, simplify, combinate, collect und substitute stehen formale Lösungsstrategien von Gleichungen zur Verfügung. Im Rahmen der Analysis können Ableitungen, Integrale, Grenzwerte, Summen und Produkte bestimmt werden. Es ist weiterhin möglich die Taylorentwicklung einer Funktion, die Länge eines Kurvenstückes und die Tangentengleichung zu erhalten. Dieser Funktionsumfang kann vor dem Hintergrund der o.g. Texasrechner als durchaus gering angesehen werden. Für den Unterrichtsalltag erscheinen unter Beachtung des traditionellen Curriculums der Analysis diese Funktionen eines CAS ein sinnvolles und ausreichendes Werkzeug, letztlich auch der dabei günstige Preis. Dagegen stehen keine Hilfen im CAS zur Verfügung, wenn es um Lineare Algebra, Vektorrechnung und Wahrscheinlichkeitsrechnung geht. Hier stehen dann nur numerische Werkzeuge wie Listen, Matrizen und Wahrscheinlichkeitsfunktionen mit numerischen Koeffizienten zur Verfügung. Hier kann man nur hoffen, dass es bald eine Weiterentwicklung des Rechners geben wird. In Frankreich ist der GRAPH 100 - so wird er dort bezeichnet - ein Verkaufsrenner auch gegenüber dem TI 89 / TI92, wohl auch deshalb, weil numerische Graphikrechner seit Jahrzehnten dort im Unterricht verbreitet sind und ein Wechsel zum Algebra FX 2.0 dann dem bisherigen Nutzer sehr leicht fällt.

Der Computer Extender von Casio

Technische Daten des Cassiopeia:
Windows CE 2.11 Betriebssystem einschließlich Pocket Word, Pocket Excel and Pocket PowerPoint; Hitachi SH-3 (80MHz) CPU Prozessor; 16MB RAM; 2MB eingebautes Flash Memory; 640 x 240 Punkte Anzeige; Bedienung mit Stift auf der Anzeigeoberfläche (Pen Touch Screen); RS 232C serielle Schnittstelle; Zusätzliche serielle Schnittstelle mit Chinchstecker; PCMCIA Karteneinschub für weitere Schnittstellen: Internet, Netzwerk, VGA-Adapter und anderes; Infrarotschnittstelle; Einschubkarte mit Geometer's Sketchpad, Maple; Math Resources Graphing; weitere Windows CE Programme wie Taschenrechner, Terminplaner etc.

Die Besonderheit dieses Handheld von Casio liegt in der verfügbaren Software, die ein bewährtes CAS-System, nämlich Maple V unter Windows CE ermöglicht. Maple for Cassiopeia für Windows CE entspricht weitgehend dem bekannten Maple V für den PC. Die Kompatibilität ist ab Release 4.0 gesichert. Die Bedienung ist benutzerfreundlich dialogorientiert. Es werden alle verfügbaren Befehle aus Menüs heraus mit dem Stift markiert. Der Leistungsumfang ist entsprechend groß. Alle Terme können über eine eingeblendete Tabelle mit dem Stift geschrieben werden, nach dem Editieren erfolgt eine Syntaxprüfung. Ein korrekt geschriebener Ausdruck kann dann über die Aktionsliste mit allen verfügbaren Befehlen weiterbearbeitet werden. Die Graphikfähigkeiten schließen zwei- und dreidimensionale Darstellungen ein, ebenso können Scharen von Funktionen gezeichnet werden. Dabei wird die typische, im einzelnen mühsame Syntax von Maple V wie beim Desktop nicht benötigt. Worksheets aus Maple V können im Handheld verarbeitet werden, ebenso können Dateien des Cassiopeia im Desktop genutzt werden. Daneben kann der versierte Maple V Nutzer die zeichenorientierten Befehle eingeben. Eine relativ große Tastatur erleichtert die Eingabe von Zeichen. Der Cassiopeia wird derzeit an Gymnasien und Realschulen im Unterricht versuchsweise eingesetzt. Die Ergebnisse werden sicher zu gegebener Zeit Interessierten zugänglich gemacht.

Karel Tschacher (Heroldsberg)
karel.tschacher@t-online.de

CAS-Taschenrechner: Ein vergleichender Überblick

H.-W. Henn

Computer in der Schule

Die Entwicklung und leichte Verfügbarkeit von Hard- und Software für den Mathematikunterricht macht, wie es Hans Schupp schon 1994 ausdrückte, es notwendig, über Dinge nachzudenken, über die wir auch ohne Computer schon lange hätten nachdenken müssen. Insbesondere denke ich an die Computeralgebrasysteme (CAS), die die vorwiegende Kalkülorientierung des Mathematikunterrichts infrage stellen. Der Einsatz des Werkzeugs CAS kann durch seine Visualisierungsmöglichkeiten den Erwerb mathematischer Begriffe und den Aufbau adäquater Grundvorstellungen unterstützen, kann durch die leichte Möglichkeit von Beispielrechungen und Experimenten heuristische Phasen und Problemlösestrategien unterstützen und kann durch die Übernahme kalkülbasierter, komplizierter Rechnungen den Schülerinnen und Schülern die Konzentration auf Idee und Bedeutung mathematischer Konstrukte und wirkliche mathematische Modellierung erleichtern. Der Rechner kann aber nur dann den Lernenden als ein zentrales und bereicherndes Werkzeug zum Lernen von Mathematik dienen, wenn er bei Bedarf auch tatsächlich zur Verfügung steht, und zwar sowohl bei der Arbeit in der Schule und zu Hause als auch bei Prüfungen. Aus diesem Grund sind relativ preiswerte „Taschen-Computer" (TC) besonders wichtig.

Die drei untersuchten „Taschen-Computer"

Allgemeine Bemerkungen

Für meine vergleichende Analyse habe ich mit den drei Taschencomputern Casio FX-2.0, Texas Instruments TI-89 und Casio Cassiopeia, die mir von den Herstellern freundlicherweise zur Verfügung gestellt worden sind, gearbeitet. Ausgegangen bin ich von den Anforderungen im gymnasialen Mathematikunterricht der Oberstufe. Der Rechner sollte, einmal angeschafft, bis zum Ende der Schulzeit eingesetzt werden können, möglichst einfach zu bedienen sein und allen im schulischen Mathematikunterricht (derzeit und in der absehbaren Zukunft) anfallenden Aufgaben gewachsen sein. Auf Fähigkeiten der drei TC, die für den schulischen Einsatz nicht primär von Bedeutung sind, werde ich nicht oder nur am Rande eingehen. Während die ersten beiden Rechner in derselben Preisregion liegen, ist der Casio Cassiopeia aufgrund seines Preises einer ganz anderen Kategorie zuzuordnen; nur aufgrund seiner Größe und seines Gewichts passt er zu den anderen beiden Rechnern. Allerdings kann sich der Preisabstand bei der rasanten Entwicklung in diesem Bereich schnell verringern.

Die Verwendung dieser Rechner ist für einen „bequem gewordenen" Benutzer der Windows-Versionen von Derive und Maple, wie ich es bin, nicht einfach. Wer nicht von Graphik-Taschenrechnern „aufsteigt", wird insbesondere mit den vielen Tastenkombinationen, die für die reichhaltigen Möglichkeiten der beiden ersten Rechner nötig sind, zunächst Schwierigkeiten haben. Hier wäre m.E. oft „weniger mehr".

Für alle drei gilt, dass sie den üblichen Aufgaben der Schulmathematik gewachsen sind. Die Bedienungsanleitungen der beiden ersten Rechner sind sehr dick, aber unübersichtlich und wenig benutzerfreundlich geschrieben (was leider bei Anleitungen im Computerbereich eher die Regel ist). Die Bedienungsanleitung des dritten Rechners ist dagegen sehr kurz.

Nach Betätigung des Einschaltknopfs sind alle drei TC sofort betriebsbereit; man kann so schnell eine kurze Rechnung machen oder einen Graphen zeichnen lassen, ohne die extrem langen Startzeiten eines üblichen Windows-PCs abwarten zu müssen. Dies ist ein für den Unterrichtsalltag wichtiger Aspekt. Ein weiterer wichtiger Aspekt ist, dass man auch nach längerer Nichtbenutzung (man denke etwa an die fast siebenwöchigen Sommerferien) gleich wieder mit dem Rechner arbeiten kann und nicht alle nötigen Befehle und Tastencodes erneut nachschlagen muss. In diesem Punkt schneiden die beiden ersten Rechner nicht besonders gut ab. Die vielen nötigen Tastenkombinationen sind software-ergonomisch und mnemotechnisch ungünstig, während beim Cassiopeia die Menüführung es auch dem unbedarften Benutzer relativ einfach macht, wieder „zur Sache zu kommen". Für alle drei Rechner sind Verbindungskabel und Treiberprogramme für den Daten- und File-Austausch mit einem PC erhältlich. Fehlermeldungen sind beim Cassiopeia am spezifiziertesten. Beim FS-2.0 erhält man meistens die Meldung „Syntaxfehler" und weiß dann oft nicht, was falsch ist. Die Verwendung schon berechneter Terme ist am einfachsten beim TI-89 und beim Cassiopeia. Durch die Flash-Technologie von TI-89 und FX-2.0 sind elektronische Updates und die Verwendung von neuen Hilfsroutinen möglich, während der Cassiopeia durch seine Steckkarten ausbaufähig ist.

Zunächst will ich eine Kurzbeschreibung für die drei Rechner geben. Genauer gehe ich (durchaus nach subjektiven Maßstäben) auf die hier im Vordergrund stehenden Graphik- und CAS-Möglichkeiten der drei Rechner ein. Einige (Graphik- und CAS-Möglichkeiten betreffende) Vergleiche zwischen den drei Rechnern werden dann in Abschnitt 3 gemacht.

Casio FX-2.0

Der FX-2.0 ist ein sehr leistungsfähiger, programmierbarer Graphik-Taschenrechner, der zusätzlich ein CAS-Modul bekommen hat. Der Rechner ist ca. $17,5 \times 8,5 \times 2,5$ cm^3 groß und wiegt ca. 280 g. Betrieben wird er mit 4 Microbatterien und einer Lithiumbatterie (als Sicherungsbatterie). Der Bildschirm ist ca. 62×40 mm^2 groß und hat 127×63 Pixel. Der Rechner kostet ca. 180 DM. Für den Einsatz im Klassenzimmer gibt es auch eine spezielle Version des FX-2.0 mit einem netzunabhängigen Overhead-Aufsatz, beides wird bei Bestellung eines Klassensatzes des FX-2.0 kostenlos mitgeliefert.

Der FX-2.0 hat ein Hauptmenü mit Untermenüs (z.B. Graphen und Wertetabellen, Kegelschnitte, Statistik, Gleichungen lösen), bei denen die mathematischen Aufgaben alle numerisch gelöst werden.

Beispielsweise werden im Untermenü EQUA(tions) alle n reellen und komplexen Näherungslösungen algebraischer Gleichungen bis zum Grad $n = 30$ gefunden. Dies benötigt allerdings seine Zeit, für die Bestimmung der Nullstellen des Kreisteilungspolynoms $x^{30} - 1$ braucht der Rechner ca. 4 Minuten. Im Untermenü GRPH-TBL können bis zu 20 Funktionsterme eingegeben werden, deren Graphen gezeichnet werden können und von denen Wertetabellen berechnet werden können. Für uns ist besonders das Untermenü CAS interessant. In diesem Menü werden in der oberen Zeile des Bildschirms Befehle und Ausdrücke eingegeben, deren Ergebnisse dann nach Drücken der Exe-Taste in „pretty print“ im unteren Teil des Fensters ausgegeben werden. Durch den beschränkten Bildschirm ist die Eingabe-Darstellung etwas gewöhnungsbedürftig. Beispielsweise ist `∫(e-X^2` eine korrekte Schreibweise für das Integral $\int e^{-x^2}\,dx$. Die Interaktion zwischen CAS-Modul und den anderen Modulen, etwa dem Graphik/Tabellen-Modul, ist etwas umständlich. Im CAS-Modus können z.B. Funktionsterme aus dem Graphik-Speicher aufgerufen und weiterbearbeitet werden, und es können Ergebnisse an einen Funktionsterm im Graphik-Speicher zugewiesen werden.

TI-89

Der TI-89 ist eine vereinfachte und deutlich verkleinerte Version des TI-92, in dem im wesentlichen das CAS Derive implementiert ist. Diese Rechner stehen auch in der Tradition der graphikfähigen und programmierbaren TR von Texas Instruments, sind aber speziell als „Derive-Taschencomputer“ entwickelt worden. Der TI-92 hatte noch das Dynamische Geometriesystem Cabri II implementiert, das aber in der Zwischenzeit (wie auch das DGS Geometer's Sketchpad) als Zusatzsoftware relativ preiswert auch für den TI-89 erhältlich ist. (Bei den Geometrieprogrammen setzt allerdings der beschränkte Graphik-Bildschirm Grenzen.) Der Rechner ist $18{,}5 \times 9 \times 2{,}5$ cm^3 groß und wiegt ca. 280 g. Betrieben wird er mit 4 Microbatterien und einer Lithiumbatterie (als Sicherungsbatterie). Der Bildschirm ist ca. 63×38 mm^2 groß und hat 160×100 Pixel. Der Rechner kostet ca. 260 DM. Der Overhead-Zusatz (ca. 400 DM) für den Einsatz im Klassenzimmer wird einfach an einen TI-89 angeschlossen und benötigt keine eigene Stromversorgung (so kann z.B. jeder Schüler seine Bearbeitung auf seinem Rechner schnell der ganzen Klasse vorführen). Während der TI-92 automatisch einen Overhead-Anschluss hat, muss hierfür am TI-89 leider ein spezieller Anschluss vorhanden sein, der den Rechner ca. 50 DM teurer macht. Im Algebra-Fenster werden Befehle und Ausdrücke in der unteren Bildschirmzeile eingegeben. Nach Drücken der Enter-Taste werden sie im oberen Teil des Fensters links nochmals wie eingegeben, rechts in „pretty print“ ausgegeben. Graphen werden in einem Graphik-Fenster ausgegeben. Wie beim „normalen“ Derive können Fenster auch gesplittet werden, etwa ein Algebra- und ein zugehöriges Graphik-Fenster nebeneinander. Es gibt schon viel Literatur zum Einsatz dieses Rechners, die auf die deutsche Schullandschaft passt.

Casio Cassiopeia

Dieser Rechner ist eine Klasse für sich, es ist eigentlich ein kleiner Computer, der unter dem sehr stark vereinfachten Betriebssystem Windows CE läuft und „von Hause aus“ nur eine vereinfachte Office-Version (mit Word, Excel und PowerPoint) implementiert hat. Der Bildschirm ist berührungssensitiv; ein spezieller Stift dient als Maus-Ersatz (ganz zufriedenstellend ist diese Technik aber noch nicht; oft muss man mehrfach mit dem Stift einen Befehl anklicken, bis er ausgeführt wird). Via Steckkarten kann man auch das CAS „Maple-CE“ und das DGS Geometer's Sketchpad betreiben. Stromquelle sind 2 Microbatterien und eine Lithiumbatterie (als Sicherungsbatterie). Mitgeliefert werden ein wiederaufladbares Batterienset und ein Netzgerät (diese sind wegen der geringen Lebensdauer der normalen Batterien auch sehr empfehlenswert). Der Cassiopeia hat ein Gewicht von ca. 450 g und zugeklappt die Maße von ca. $18.5 \times 9{,}5 \times 2{,}5$ cm^3. Der aufgeklappte Rechner hat im unteren Teil eine kleine, aber (fast) normale Tastatur (mit allerdings sehr kleinen Tasten) und im oberen Teil einen Bildschirm, der 157×61 mm^2 groß ist, 640×240 Pixel hat und vier Graustufen darstellen kann. Der Prozessor ist deutlich schneller als die Prozessoren der beiden anderen TC. Leider hat der Cassiopeia den gewaltigen, seinen breiten Einsatz in der Schule noch verhindernden Nachteil eines Preises von ca. 1.500 DM (davon entfallen ca. 300 DM auf die hier interessierende Karte für Maple-CE). Der Bildschirm ist allerdings auch bei Kontrastoptimierung nur bei „geeigneten" Lichtbedingen gut ablesbar. Das Maple-CE erinnert in seiner Bedienung an Derive (for Windows), ist aber mit dem „normalen" Windows-Maple - was den mathematischen Teil angeht - weitgehend kompatibel, ohne allerdings alle Befehle zu kennen und dessen Worksheet-Oberfläche zu haben. Ausdrücke und Befehle werden in der unteren Bildschirmzeile eingegeben und nach Drücken der Enter-Taste im oberen Fenster nochmals wie eingegeben, dann ausgeführt ausgegeben. Die Darstellung ist leider nicht in „pretty print" dargestellt und daher oft nicht besonders schön, da z.B. das Integralzeichen aus den Zeichen / und — „zusammengebastelt" wird, so dass man für ein bestimmtes Integral 7 bildschirmfüllende Zeilen benötigt. Darüber hinaus können zwischen den Mathematik-Zeilen noch Kommentar-Zeilen eingefügt werden. Wie das „normale" Maple ist auch das Cassiopeia-Maple Befehl-sensitiv: wird ein Befehl oder eine Ausgabe markiert, so werden im Menü *Action* diejenigen Befehle angezeigt, die man anwenden kann. Nach Anwählen eines Befehls wird er dann in vollständiger Syntax in die Editor-Zeile geschrieben; dort, wo noch etwas einzufügen ist, stehen Fragezeichen. Das Cassiopeia-Maple versteht viele der „normalen" Maple-Befehle; aus dem mitgeliefertern kleinen Handbuch geht allerdings nicht hervor, welche. Darüber hinaus können zwischen den Mathematik-Zeilen noch Kommentar-Zeilen eingegeben werden. Leider funktioniert das Übertragen von Text oder Graphik aus dem Cassiopeia-Maple in das Cassiopeia-Word nicht.

Vergleiche einiger beispielhafter, für die Schule wichtigen Graphik- und CAS-Anwendungen

Graphik

2-D-Graphik

Alle drei Rechner können Funktionen einer Variablen in kartesischen oder Polar-Koordinaten graphisch darstellen. Besonders viele verschiedene Graphikmöglichkeiten beherrscht der TI-89. Leider hat das Plot-Fenster des Cassiopeia-Maple nur ca. ein Drittel des Bildschirms und kann nicht vergrößert werden, so dass der Rechner seinen „Pixel-Vorteil" gar nicht so stark ausnutzt. Es können bei allen drei TC auch Punkte (aus Liste oder als Folgenwerte) (als Punkt-Plot oder als Polygonzug) und Parameterkurven graphisch dargestellt werden. Beim FX-2.0 und beim TI-89 können mit einem einfachen Befehl auch „Spinnweb-Diagramme" erzeugt werden. Natürlich kann der Bildausschnitt vorgegeben und später verändert werden. Eine Zoom-Möglichkeit (mit Wahl des Zoomzentrums und des Zoomfaktors) haben nur der FX-2.0 und der TI-89. Diese Möglichkeit ist z.B. zur Visualisierung der lokalen Linearität differenzierbarer Kurven sehr gut. Ebenfalls nur beim FX-2.0 und beim TI-89 können Graphen mit Hilfe des „Trace-Modus" abgetastet werden, so dass schnell eine qualitative Funktionsuntersuchung gemacht werden kann.

Implizite Funktionen werden unterschiedlich behandelt: Der FX-2.0 kann Kegelschnitte (anhand vorgegebener Gleichungstypen im Untermenü CONICS) darstellen, während die beiden anderen Rechner die Graphen „beliebiger" impliziter Funktionen $f(x, y) = 0$ darstellen können (was beim TI-89 je nach dem längere Rechenzeiten erfordert).

3-D-Graphik

Der FX-2.0 kann keine 3-D-Graphik darstellen. Der TI-92 kann 3-D-Graphen erzeugen und erlaubt es, Blickwinkel, Größe und Darstellungsart zu verändern. Der Cassiopeia kann Graphen von $z = f(x, y)$ darstellen, allerdings nur mit vorgegebenem Koordinatensystem und Blickwinkel; ändern kann man nur den dargestellten x- und y-Bereich. 3-D-Graphen sind natürlich rechenzeitaufwendig und bei der beschränkten Bildschirmauflösung nicht sehr detailreich.

Dynamische Graphik

Im Untermenü DYNA(misierung) des FX-2.0 kann man nach Angabe eines Parameterbereichs „a = Startwert (Schrittweite) Endwert" die Graphen der Funktionen $f(a, x)$ berechnen lassen. Sie werden dann nacheinander auf dem Bildschirm dargestellt. Damit kann z.B. der Einfluss eines Parameters auf die Kurvenform studiert werden. Die Berechnung ist natürlich zeitaufwendig (z.B. für die Funktionen $f(a, x) = \sin(a \cdot x)$ mit $a = -1$ (1) 5 etwa 45 Sekunden). Maple-CE des Cassiopeia kennt nicht den entsprechenden Befehl *animate* des „normalen" Maple. Beim TI-89 lassen sich 3-D-Graphen z.B. durch Verändern des Betrachtungswinkels animieren.

Faktorisierungsbeispiel: Primfaktorzerlegung zweier Fermatzahlen

	FX-2.0	**TI-89**	**Cassiopeia**
$2^{32}+1$	Zerlegung in ca. 6 Sek.	Zerlegung in ca. 2 Sek.	Zerlegung in ca. 2 Sek.
$2^{64}+1$	*Keine* Zerlegung; Ausmultiplizieren ca. 8 Sek.	Zerlegung in ca. 30 Sek.	Zerlegung in ca. 3 Sek.

Allerdings war bei der nächsten Fermatzahl $2^{128}+1$ auch der Cassiopeia überfordert; nach 2,5 Stunden habe ich die Rechnung abgebrochen (das Ausmultiplizieren der 128 Zweier-Faktoren ging natürlich in Sekundenschnelle).

Exaktes Lösen von Gleichungen

Gemeinsame Eigenschaften: Exaktes Lösen linearer Gleichungssysteme (auch mit Parametern), beim FX-2.0 müssen allerdings n Gleichungen mit n zu bestimmenden Unbekannten eingegeben werden. Bei algebraischen Gleichungen werden die rationalen Lösungen gefunden, bei algebraischen Gleichungen vom Grad 1 und 2 (auch mit Parametern) werden alle Lösungen gefunden. Der Cassiopeia kann auch algebraische Gleichungen vom Grad 3 exakt lösen. Alle können sehr einfache nicht-lineare Gleichungen (z.B. $\sqrt{x^2+4}=5$) lösen. Einfache nicht-lineare Gleichungssysteme (z.B. das System $x \cdot y = 2,\ x + y = 3$) werden (manchmal) vom TI-89 und vom Cassiopeia gelöst.

Terme und Termumformungen

Alle Rechner haben einen Befehl zum Approximieren. Beim FX-2.0 kann mit dem Befehl APPROX im CAS-Modus allerdings nur eine konkrete reelle Zahl rational approximiert werden, der Befehl APPROX kann z.B. nicht auf ein bestimmtes Integral angewandt werden.

Folgen werden beim FX-2.0 über das Untermenü RECUR(sion) eingegeben; sie können als einzigen Parameter den Folgenindex n haben. Beim FX-2.0 und beim TI-89 können Folgen auch rekursiv definiert werden (beim FX-2.0 kann f_n maximal von den beiden vorhergehenden Gliedern abhängen, beim TI-89 von beliebig vielen). Beim Cassiopeia muss dies über Maple-Prozeduren realisiert werden, was komplizierter ist.

Bei allen drei Rechnern lassen sich rationale algebraische Terme eingeben und algebraisch manipulieren (Ausmultiplizieren, Faktorisieren, Zusammenfassen, Vereinfachen – wobei letzteres bei allen CAS unterschiedliche Ergebnisse liefern kann).

Calculus

Alle Rechner beherrschen die „Grundaufgaben der Analysis“ bei viel mehr als den in der Schule vorkommenden Funktionen: das Bestimmen von Ableitungs- und Stammfunktionen, Grenzwerten, Summen, Produkten, Taylorpolynomen. Bei Ableitungs- und Stammfunktionen dürfen auch Parameter vorkommen. Die Syntax der Befehle ist bei FX-2.0 und TI-89 ähnlich und relativ einfach zu merken. Beim Cassiopeia ist die Eingabe durch die Kontextsensitivität von Maple am einfachsten.

Beispielsweise bei Summen gibt es Unterschiede: Beim FX-2.0 dürfen nur konkrete natürliche Zahlen als Summen-Start- und -End-Wert vorkommen. Der Endwert „∞“ wird akzeptiert, aber in der Regel wird die Summe unbearbeitet zurückgegeben. Insbesondere der Cassiopeia ist hier leistungsstärker, wie die folgenden Beispiele zeigen:

	$\sum_{i=1}^{n} i$	$\sum_{i=1}^{\infty} i$	$\sum_{i=1}^{n} \frac{1}{i}$	$\sum_{i=1}^{\infty} \frac{1}{i}$	$\sum_{i=1}^{n} \frac{1}{i^2}$	$\sum_{i=1}^{\infty} \frac{1}{i^2}$
TI-92	$\frac{n\cdot(n+1)}{2}$	∞	$\sum_{i=1}^{n} \frac{1}{i}$	$\sum_{i=1}^{\infty} \frac{1}{i}$	$\sum_{i=1}^{n} \frac{1}{i^2}$	$1/6\pi^2$
Cassiopeia	$1/2(n+1)^2 - 1/2n - 1/2$	∞	$\psi(n+1)+\gamma$	∞	$-\psi(1,n+1)+1/6\pi^2$	$1/6\pi^2$

Maple-CE kennt also die Digamma-Funktion. Bei konkretem Endwert, z.B. $n = 10,\ 100,\ 1000$ zeigen FX-2.0 und TI-89 den exakten Wert mit dramatisch steigenden Rechenzeiten an, während der Cassiopeia bei $n = 10$ und $n = 100$ sehr schnell das exakte Ergebnis anzeigt und bei $n = 1000$ das Ergebnis $\psi(1001)+\gamma$. Ich glaube allerdings nicht, dass eine Rechner-Antwort mit der Digamma-Funktion für die Schule sinnvoll ist.

Alle drei Rechner berechnen Grenzwerte mit einem Limit-Befehl (wohl mit Hilfe der Regel von de l'Hospital). Die Grenzwerte

$$\text{a)}\quad \lim_{x\to 0}\frac{\sin x}{x}\,, \qquad \text{b)}\quad \lim_{x\to 0}\left(\frac{\sin x}{x}\right)^n$$

sind jeweils (für alle $n \in \mathbf{N}$) gleich 1, was alle drei Rechner auch für den Grenzwert a) herausfinden. Die beiden „kleinen“ Rechner finden für alle n, die ich ausprobiert habe, diesen Grenzwert auch bei b), dagegen gibt Maple-CE den Grenzwert b) ab $n = 12$ unbearbeitet zurück. Anscheinend wird der de l'Hospital nur 12 mal hintereinander durchgeführt. Wieder eine völlig andere Frage, nämlich eine didaktische, ist es natürlich, ob und, wenn ja, wo der Limit-Befehl in der *Schule* sinnvoll ist.

Einfache Differentialgleichungen werden in vielen Leistungskurs-Curricula genannt, so dass entsprechende Möglichkeiten des verwendeten Rechners durchaus wünschenswert sind. Der FX-2.0 bietet hier keine Möglichkeiten. Mit dem TI-89 und dem Cassiopeia-Maple kann man exakte Lösungsfunktionen einfacher DGL berechnen und kann darüber hinaus Richtungsfelder und Lösungskurven zeichnen.

Vektoren und Matrizen

Für Probleme der Analytischen Geometrie oder etwa für die in vielen Lehrplänen genannten mehrstufigen Prozesse (einfache Markoffketten) sind Operationen mit Vektoren und Matrizen nützlich. Der FX-2.0 beherrscht über das Untermenü RUN-MAT(rix) eine einfache Matrizenalgebra, allerdings ausschließlich mit numerischen Werten und im Approx-Modus. Die Größe der Matrizen ist nur durch den Speicherplatz beschränkt. Die beiden anderen Rechner können auch symbolisch mit Parametern und im Exact-Modus mit Matrizen rechnen und können z.B. Determinanten und Inverse bestimmen und die Gauss-Elimination durchführen.

Schlussbemerkungen

In einem neulich erschienenen Papier [2, 2000] wurde von den Autoren u.a. die Frage gestellt, ob in Zukunft bei der Existenz von CAS die Primfaktorzerlegung von 30 noch „händisch" machbar sein müsse. Ohne auf den Sinn oder Unsinn dieser Fragestellung eingehen zu wollen: Wer soweit kommt, dass er den FX-2.0 oder den TI-89 sinnvoll bedienen kann, wird wohl noch ganz andere Dinge im Kopf machen können . . . .

In der deutschen Schulrealität sind der TI-92 und der TI-89 ziemlich bekannt, Texas Instruments dominiert und fördert stark die T^3-Initiative (Teachers Teach with Technology). Der FX-2.0 ist erst seit kurzem auf dem deutschen Markt erhältlich, der Cassiopeia ist „brandneu". Dementsprechend gibt es noch wenig Schulerfahrung mit diesen Geräten. Als eines der Nachfolgeprojekte des Baden- Württembergischen CAS-Projekts *Mobiles Klassenzimmer* (vgl. [1, 1998] arbeitet eine Oberstufenklasse auch mit dem Cassiopeia.

Keiner der drei Rechner ist „rundherum" zufriedenstellend, in jedem Fall sind aber alle drei in der Schule brauchbar und auf „dem richtigen Weg". Meiner Meinung nach haben FX-2.0 und TI-89 zu viele, für den schulischen Einsatz nicht notwendige Möglichkeiten, die aber die Bedienung unnötig verkomplizieren. Mein Wunsch ist ein TC mit der leichten Bedienbarkeit und dem Display des Cassiopeia und dem Preis des FX-2.0. Ich bin sicher, dass dieser Wunsch bald in Erfüllung gehen wird.

Literatur

[1] Henn, H.-W.: Das Pilotprojekt *Mobiles Klassenzimmer* in Baden-Württemberg. – In: Beiträge zum Mathematikunterricht. Hildesheim: Franzbecker 1998, S. 279 – 282

[2] Herget, W. u.a.: Welche handwerklichen Rechenkompetenzen sind im CAS-Zeitalter unverzichtbar? – `http://www.kutzler.com/bk/a-is-g/rech-komp.htm` [1]

Hans–Wolfgang Henn (Dortmund)
`Wolfgang.Henn@mathematik.uni-dortmund.de`

[1] s. auch S. 25 dieses Rundbriefs.

Welche handwerklichen Rechenkompetenzen sind im CAS-Zeitalter unverzichtbar?

Zusammenfassung: Wir gehen der Frage nach, welche handwerklichen Rechenfertigkeiten trotz der Verfügbarkeit algebraischer Taschenrechner und Computer mit Computeralgebrasystemen (CAS) unverzichtbar sind: Was sollte auch in Zukunft jede Schülerin und jeder Schüler noch „per Hand", d. h. allein mit Schreibstift und Papier, können? Dieser Text entstand in einer zweitägigen Diskussion der vier Autoren zu diesem Thema. Das vorliegende Ergebnis ist sicherlich eine Herausforderung – wir möchten damit zu einer breiten Diskussion über im Mathematikunterricht zu vermittelnde unverzichtbare, dauerhaft verfügbare Rechenkompetenzen beitragen bzw. eine solche in Gang setzen.

Einleitung

Weit verbreitete CAS an deutschen und österreichischen Schulen sind das Computerprogramm Derive und die Taschenrechner TI-92 und TI-89. Einführungen in die Bedienung dieser Werkzeuge sind [Kutzler&Kokol-Voljc 2000] für Derive 5, [Kutzler 1996] für den TI-92 und [Kutzler 1998] für den TI-89. Diese und ähnliche Werkzeuge werden an den Schulen bald ebenso selbstverständlich sein, wie es heute numerische Taschenrechner sind.

Ausgangspunkt: Schriftliche Prüfung ohne Rechner

Wir gehen aus von einer zweigeteilten Prüfung, bei der ein Teil ohne moderne technische Hilfsmittel stattfindet – es darf also nicht einmal ein einfacher wissenschaftlicher Taschenrechner verwendet werden – während beim zweiten Teil Technologie[2] wie insbesondere leistungsfähige Taschenrechner und Computer mit CAS eingesetzt werden dürfen. Dieses Modell einer zweigeteilten Prüfung wird in manchen Ländern, z. B. in Österreich, erprobt; in anderen Ländern, z. B. in England, wird es bereits eingesetzt. Dieser Ansatz *könnte ein Kompromiss* sein, um sowohl den Wünschen der Technologie-Befürworter als auch den Vorbehalten der Technologie-Gegner zu entsprechen. Prinzipielle Gedanken zu einer zweigeteilten Prüfung sind in [Kutzler 1999] formuliert.

Wir stellen uns im Folgenden eine fiktive *schriftliche* technologie-freie Prüfung vor und suchen nach Aufgaben und Aufgabentypen, die in einer derartigen Prüfung gestellt werden könnten.

Die Grenzziehung zwischen Aufgaben, die bei einer technologie-freien Prüfung gestellt würden, und Aufgaben, die bei einer solchen Prüfung nicht gestellt werden sollten, läuft auf die eingangs gestellte Frage hinaus, welche handwerklichen Rechenkompetenzen Schülerinnen und Schüler heute noch haben sollten. Die *fiktive Prüfungssituation* ist für uns daher Mittel zum Zweck. Unsere Diskussion und die dabei erzielten Ergebnisse haben eine weit über Prüfungssituationen hinausgehende Bedeutung. Sie ist fundamental für eine Weiterentwicklung des Mathematikunterrichts in den nächsten Jahren und Jahrzehnten. Nachdem die Bedeutung der Rechenfertigkeiten neu überdacht und stark zurückgedrängt wurde, ist es besonders wichtig, die sich daraus für den Unterricht ergebenden Konsequenzen zu diskutieren. Mit diesem Thema werden wir uns im Anschluss an diese Arbeit beschäftigen.

Drei Töpfe: –T, ?T, +T

Die gesuchte Grenze zwischen Aufgaben, die bei einer fiktiven technologie-freien Prüfung gestellt würden, und Aufgaben, die bei einer solchen Prüfung nicht gestellt werden sollten, ist fließend und hängt von vielen Parametern ab, natürlich auch vom Schultyp. Wir versuchen eine möglichst allgemeingültige Antwort und schaffen dazu drei „Töpfe", die wir −T, ?T und +T nennen.

[2] Hier müßte es richtig „Technik" oder „Rechner" heißen. Das Wort „Technologie" ist laut DUDEN „die Lehre von der Umwandlung von Rohstoffen in Fertigprodukte." Allerdings hat sich in der Literatur zur Verwendung von Taschenrechnern und Computern im Unterricht die Verwendung dieses Wortes eingeschliffen, weshalb wir – um Verwirrungen zu vermeiden – das Wort in diesem Text ebenso einsetzen.

- Der erste Topf, **–T** (= ohne Technologie), beinhaltet jene Aufgaben, die bei einer technologie-freien Prüfung zu stellen wären. In diesen Topf kommen also all jene Aufgaben, von denen wir erwarten, dass Schülerinnen und Schüler sie ohne Zuhilfenahme *irgend*eines Taschenrechners oder Computers lösen können.

- Die durch den Topf –T bezeichneten Rechenfertigkeiten sollen ab der 8. Jahrgangsstufe gelten bzw. ab jener Jahrgangsstufe, in der der betreffende Stoff behandelt wird. Diese Rechenfertigkeiten sollen dann über die jeweilige Jahrgangsstufe hinaus *dauerhaft* erhalten bleiben und wirklich *jederzeit* gefordert werden können.

- Der dritte Topf, **+T** (= mit Technologie), beinhaltet jene Aufgaben, die bei einer solchen Prüfung nicht gestellt werden sollten, d. h. bei der Lösung dieser Aufgaben darf ein leistungsfähiger Taschenrechner oder ein Computer mit CAS verwendet werden.

- Der zweite Topf, **?T**, spiegelt unsere Zweifel, unsere unterschiedlichen Einstellungen und zum Teil auch die grundsätzliche Problematik dieses Themas wieder. Bei den in diesem Topf gelandeten Aufgaben gingen die Meinungen der vier Autoren auseinander, oder wir waren uns einig, dass wir keine Zuordnung zu einem der beiden anderen Töpfe vornehmen wollten oder konnten. Dieser Topf kennzeichnet, wie fließend die Grenze für uns (noch) ist.

Wo immer es machbar war, haben wir das Spektrum und die Grenzen eines konkreten Aufgabentyps dadurch abgesteckt, dass wir vergleichbare Aufgabenvarianten für –T und +T angegeben haben.

Höhere Anforderungen in Übungsphasen und Unterricht

Die Aufgaben im Topf +T sind solche, die wir in einer technologie-freien Prüfung nicht stellen würden – aber wir würden derartige Aufgaben auch in einer Prüfung nicht stellen, bei der moderne Rechner verfügbar sind: Denn diese Aufgaben erscheinen uns nur in einem passenden Problem-Zusammenhang als sinnvoll, nicht aber als isolierter Prüfungsteil. Sie wären wohl lediglich im Stande, Fertigkeiten in der Bedienung des Rechners zu testen. Die von uns in den Topf –T gegebenen Aufgaben beschreiben eine langfristig zu erhaltende handwerkliche Kompetenz. Um dieses Ziel zu erreichen, sollte sehr wohl in der anfänglichen Übungsphase „die Latte entsprechend höher gelegt“ werden! Aus solchen Gründen könnte es in begrenztem Umfang auch sinnvoll sein, selbst Aufgaben aus +T im Unterricht sogar *technologie-frei* zu üben.

Andere wichtige Kompetenzen

Neben der Rechenkompetenz gibt es auch noch andere, wichtige Kompetenzen, die ihre Bedeutung im CAS-Zeitalter behalten oder sogar an Bedeutung gewinnen – jedenfalls unverzichtbar sind (siehe auch [Heugl 1999]). Beispiele solcher Kompetenzen sind:

- die Kompetenz, Terme zu finden
- die Strukturerkennungskompetenz
- die Testkompetenz
- die Visualisierungskompetenz
- die Kompetenz, Technologie passend einzusetzen
- die Kompetenz, Rechnerarbeit passend zur Aufgabenstellung zu dokumentieren.

Zur Visualisierungskompetenz gehört z. B. die Fähigkeit, eine „richtige Handbewegung“ ausführen zu können, wenn der Verlauf des Graphen von zum Beispiel x^2 oder $\sin(x)$ gefragt ist.

In der Gesamtheit der im Mathematikunterricht zu vermittelnden Kompetenzen kommt der Rechenkompetenz eine wichtige Rolle zu. Sie zu vermitteln ist nicht nur Selbstzweck (dann wäre ihre Bedeutung angesichts leistungsfähiger Rechner sehr in Frage gestellt!), sondern in einem gewissen Rahmen auch erforderlich für den Erwerb und die Nutzung „höherer“ Kompetenzen wie den oben genannten. Daher spielen die genannten und weitere Kompetenzen bei der Bewertung der Bedeutung von Rechenfertigkeiten eine mitentscheidende Rolle und waren deshalb auch Inhalt unserer Diskussion. Zum Teil geht das aus den Kommentaren hervor, die zu einigen Aufgaben gegeben werden.

Mathematikunterricht wird nicht einfacher!

Keineswegs glauben wir, dass der Mathematikunterricht in Zukunft einfacher werden wird – im Gegenteil. Mit dem in den folgenden Tabellen zum Ausdruck gebrachten geringeren Ansprüchen bei handwerklichen Rechenfertigkeiten wird zugleich unsere Überzeugung ausgedrückt, dass CAS bald zu einem Standardwerkzeug des Mathematikunterrichtes und der Mathematikanwender gehören wird. In der Folge wird Mathematik nutzbarer und damit sehr wahrscheinlich auch insgesamt anspruchsvoller, keinesfalls aber trivialer. Wir wollen nach der unglücklichen Diskussion zum Thema „7 Jahre Matheunterricht sind genug“ nicht eine ebensolche zum Thema „Triviale Termumformungen sind genug“ aufkommen lassen. Zentral für uns ist eine Unterscheidung zwischen den Zielen „Rechnungen ausführen können“ (das kann teilweise an den Rechner delegiert werden) und „über Strategien entscheiden können“ (das kann der Rechner **nicht** übernehmen.)

Selbstverständlich haben die folgenden Darlegungen Auswirkungen auf viele Bereiche des Mathematikunterrichts und sein Umfeld: Auf die Unterrichtsführung, auf neuartige Übungsformen, auf Hausarbeiten, Lehrpläne, auf die Unterrichtsinhalte in den späteren Jahrgängen, auf die erforderlichen Kompetenzen der Lehrenden usw. Wir haben solche Aspekte zwar andiskutiert, aber nicht ausdiskutiert. Sie werden deshalb hier nicht angesprochen.

Unser Ziel: dauerhaft verfügbare Mindest-Rechenkompetenzen

Es ist unser Ziel, mit diesem Bericht die erforderliche und zum Teil schon überfällige Diskussion über inhaltliche, didaktisch-methodische und organisatorische Konsequenzen des Einsatzes von CAS und anderer Mathematik-Software in Gang zu bringen bzw. zu fördern.

Dieser Text ist daher bewusst herausfordernd, vielleicht sogar provokativ. Es gilt, sich der Herausforderung durch die neuen Möglichkeiten zu stellen und daraus Konsequenzen zu ziehen. Das verlangt insbesondere auch die Bereitschaft, von Vertrautem Abschied zu nehmen, wenn dies als sinnvoll oder sogar unvermeidbar erkannt wird.

Aufgaben und Aufgabentypen

Wir beschränken uns in dieser Arbeit auf Aufgaben, zu deren Lösung leistungsfähige Taschenrechner und Computer mit CAS verwendet werden können.

Arithmetik – langfristige Mindestkompetenzen

	–T (ohne Technologie)	*?T*	*+T (mit Technologie)*
01	Berechne $3 \cdot 40$		Berechne $3.2987 \cdot 4.1298$
02	Berechne $\sqrt{81}$		Berechne $\sqrt{80}$ auf ... Stellen
03	Schätze $\sqrt{80}$		Ziehe teilweise die Wurzel: $\sqrt{80}$
04			Berechne $\sqrt{11 \cdot \sqrt[3]{11}}$
05	Faktorisiere 15		Faktorisiere 30

Das Beispiel $\sqrt{80}$ (mit den Varianten –T03, +T02 und +T03) zeigt, wie wichtig die Aufgaben-Formulierung für die Entscheidung, in welchen Topf die Aufgabe kommt, ist. Je geringer die Bedeutung der handwerklichen Rechenfertigkeit, desto höher ist die Bedeutung einer passenden Aufgabenformulierung, um die Zielsetzung der Aufgabe zu verdeutlichen. An weiter unten gegebenen Aufgabentypen wird das noch klarer. Beim Lernziel „Werte schätzen können“ waren wir uns einig, dass diese Fähigkeit weit über das Beispiel (–T03) hinaus erwünscht, also allgemein so wichtig ist, dass es ohne Rechnereinsatz erreicht werden sollte. Dennoch kann hier der Einsatz eines Rechners im Unterricht sinnvoll sein, zum Beispiel als Kontrollwerkzeug, um die Güte der Schätzung zu prüfen – und den Fehler zu bestimmen – oder den Sinn des Schätzens überhaupt zu verdeutlichen.

Um möglichen Missverständnissen zu begegnen: Die Aufgaben im Topf +T sind, wie oben ausgeführt, solche, die wir in einer technologie-freien Prüfung nicht stellen würden. Allerdings würden wir solche Aufgaben auch in einer technologie-unterstützten Prüfung nicht stellen, weil diese Aufgaben als solches nutzlos erscheinen und einzig die Fertigkeit in der Bedienung eines Werkzeuges testen.

Die Aufgaben aus +T verlangen allein die *Fertigkeit,* den betreffenden, z. B. aus einer umfassenderen Aufgabe stammenden Rechenausdruck auszuwerten. Dies kann langfristig an den Taschenrechner delegiert werden. Sehr wohl aber ist sicherzustellen, dass die Schülerinnen und Schüler *verstehen,* was derartige Rechenausdrücke bedeuten – um ein solches *Verständnis* abzuprüfen, bedarf es aber ganz anderer Aufgabentypen!

Allerdings – auch dies zur Erinnerung – könnten Aufgaben aus +T sehr wohl im Unterricht geübt werden, und zwar mit und ohne Rechner-Nutzung. Dies könnte je nach Situation angebracht oder sogar erforderlich sein, um die von uns in den Topf –T gegebenen Aufgaben als langfristig zu erhaltende handwerkliche Kompetenz zu sichern.

Für unsere Vorschläge gilt grundsätzlich, dass elementare Rechenschritte (wie das Faktorisieren einer Zahl mit nur zwei Faktoren, z. B. 15) zu den unverzichtbaren Kompetenzen (und damit in den Topf –T) gehören, wohingegen Rechenschritte, die eine wiederholte Anwendung elementarer Rechenschritte erfordern (wie das Faktorisieren einer Zahl mit drei oder mehr Faktoren, z. B. 30) man bereits dem Rechner überlassen darf.

Brüche und Bruchterme – langfristige Mindestkompetenzen

	–T (ohne Technologie)	*?T*	*+T (mit Technologie)*
01	Vereinfache $\frac{10^2}{5^2}$		Vereinfache $7 \cdot \frac{2}{5} : \frac{4}{6}$
02	Vereinfache $\frac{10^2}{10^5}$		Vereinfache $\frac{100x^3y^2}{10xy^5}$
03	Vereinfache $2 : \frac{1}{2}$		
04	Vereinfache $\frac{2}{\frac{1}{2}}$		
05	Vereinfache $\frac{5a}{5}$		
06	Vereinfache $\frac{a}{5} \cdot 5$		
07	Vereinfache $\frac{2}{x} \cdot \frac{x}{y}$		Vereinfache $\frac{a}{b} \cdot \frac{b^2}{3ac}$
08			Vereinfache $3x^2 : \frac{2x}{5y^3}$
09	Vereinfache $2a - \frac{a}{3}$		Vereinfache $2a - \frac{a}{3} + \frac{a}{7}$
10	Vereinfache $\frac{a}{3} + \frac{a}{7}$		
11	Vereinfache $\frac{5}{x} - \frac{2}{x}$		
12	Vereinfache $\frac{2}{x} - \frac{5}{y}$	Vereinfache $\frac{2}{x} - \frac{x}{5}$	

–T01: Hier sollte die naheliegende Rechnung $\frac{100}{25} = 4$ gesehen werden. Auch das will gekonnt sein!

–T02: Ausdrücke dieser Art werden in der Physik gebraucht.

–T03: Eine sich hieraus ergebende alternative Fragestellung (mit höherem Anspruch) wäre: *„Warum* ist $2 : \frac{1}{2}$ gleich 4?“ Damit würde die dahinterstehende Strukturerkennungskompetenz angesprochen.

Die Regel $\frac{a}{b} + \frac{c}{d} = \frac{ad+bc}{bd}$ soll bewusst nicht als Formel „abgefragt“ werden. Sie ist für uns ein *Hintergrund-Ziel – ein Ziel,* das in dieser Form in einer schriftlichen Prüfung nicht explizit abgefragt zu werden braucht.

Das Auswendiglernen dieser Formel führt eher dazu, dass Schüler sie beim Addieren von Brüchen „stur“ verwenden, statt den in vielen Fällen günstigeren Weg der Berechnung des kleinsten gemeinsamen Vielfachen beider Nenner zu gehen. Auch $\frac{a}{b} \cdot \frac{c}{d} = \frac{ac}{bd}$ ist für uns ein solches Hintergrund-Ziel. Dennoch sind diese Formeln (die in gleicher Weise auch vom CAS erzeugt werden) ein wichtiges Thema im Unterricht, da es hierbei u. a. um Beispiele für die unerlässliche Strukturierung mathematischer Sachverhalte geht.

Terme – mit und ohne Klammern – langfristige Mindestkompetenzen

Wie bereits erwähnt ist die Aufgabenformulierung für den Wert einer Aufgabe mitentscheidend. In der folgenden Tabelle haben wir daher bewusst auf die übliche Aufforderung „Multipliziere aus“ verzichtet und statt dessen „Schreibe ohne Klammern“ verlangt. Während Ersteres die Anwendung des Distributivgesetzes suggeriert ist Letzteres neutral und erhöht damit den Wert der Aufgabe.

	–T (ohne Technologie)	?T	+T (mit Technologie)
01	Schreibe ohne Klammern: $a - (b+3)$	Schreibe ohne Klammern: $(5+p)^2$	Schreibe ohne Klammern: $3a^2(5a-2b)$
02	Schreibe ohne Klammern: $2(a+b)$		Schreibe ohne Klammern: $(a^2-3b)(-3a+5b^2)$
03	Schreibe ohne Klammern: $2(ab)$		Schreibe ohne Klammern: $(2a+t)^2$
04	Schreibe ohne Klammern: $3(5a-2b)$		Schreibe ohne Klammern: $(5+p)^3$
05	Schreibe ohne Klammern: $(3+a)(b-7)$		
06	Schreibe anders: $2a+2b$		
07	Vereinfache $x^2y^2+(xy)^2$		
08	Faktorisiere $3ab+6ac$		
09	Faktorisiere x^2-4	Faktorisiere x^2+4x+4	Faktorisiere x^2-x-6

–T09: Diese Aufgabe ist wichtig, weil sie Entscheidungs- und Begründungskompetenz entwickeln hilft, was wiederum gebraucht wird, um auf einem Taschenrechner etwa die Taste „factor“ sinnvoll wählen zu können.

Ein Hintergrund-Ziel (im Sinne der Bemerkungen zum Abschnitt „Brüche und Bruchterme“) ist hier das Distributivgesetz $a \cdot (b+c) = a \cdot b + a \cdot c$.

Über die Aufgabentypen ?T01 und ?T09 wurde besonders lange diskutiert. Gerade die eingangs erwähnte Strukturerkennungskompetenz wäre laut Meinung eines Teiles unserer Gruppe ohne die durch diese Aufgaben ausgedrückte Rechenkompetenz nicht gewährleistet. Auf der anderen Seite wurden in den österreichischen Computeralgebra-Projekten Anzeichen dafür gefunden, dass durch das Verwenden von Technologie die Strategiekompetenz gefördert wird, ohne dass eine gute Entwicklung von Rechenkompetenz an dieser Stelle unbedingt erforderlich wäre.

Lineare Gleichungen – langfristige Mindestkompetenzen

	–T (ohne Technologie)	?T	+T (mit Technologie)
01	Löse nach $x: x-6=0$		
02	Löse nach $x: 5-x=2$		
03	Löse nach $x: 3x=12$		
04	Löse nach $x: 5x-6=15$		Löse nach $x: 5x-6=2x+15$
05	Löse nach $y: \frac{y}{3}=5$		Löse nach $x: 2x+3=\frac{4}{3}$
06	Löse nach $x: a \cdot x = 5$	Löse nach $x: a \cdot x - 6 = 15$	
07	Löse nach $x: x+1=x$	Löse nach $x: 2(x+1)=2x$	
08	Löse nach $x: x+1=x+1$	Löse nach $x: 2(x+1)=2x+2$	
09	Löse nach $t: s=v \cdot t$	Löse nach $x: K=k \cdot x + F$	
10	Löse nach $r: U=2r\pi$		
11	Löse nach $x: \lvert x \rvert = 1$		

–T06: Dieses Beispiel ist wichtig, weil die heute verfügbaren CAS die hier erforderliche Fallunterscheidung bezüglich a nicht machen.

–T11: Da bei einem CAS die Betragsfunktion oft im Ergebnis auftritt, sollen Schülerinnen und Schüler diese Funktion kennen und in einfachen Situationen wie hier auch technologie-frei handhaben können.

Quadratische Gleichungen – langfristige Mindestkompetenzen

	−T (ohne Technologie)	?T	+T (mit Technologie)
01	Löse nach $x : x^2 = 4$		Löse nach $x : 9x^2 = 4$
02	Löse nach $x : x^2 - 4 = 0$		Löse nach $x : 9x^2 - 4 = 0$
03	Löse nach $x : x^2 - x = 0$		
04	Löse nach $x : x^2 - 4x = 0$	Löse nach $x : x^2 + 4x + 4 = 0$	Löse nach $x : 2x^2 - 5x + 9 = 0$
05	Löse nach $x : x^2 = a$		
06	Löse nach $r : A = 4\pi r^2$		Löse nach $v_0 : x = \frac{1}{2a} \cdot v_0^2$

+T04 und ?T04 markieren eine der auf den ersten Blick einschneidendsten Veränderungen: Die „*p-q*-Formel“ für die Lösung einer quadratischen Gleichung zählt für uns nicht mehr zum verbindlichen Katalog der sicheren handwerklichen Fähigkeiten, bleibt aber wegen ihrer Bedeutung und den typischen Fallunterscheidungen eines der Hintergrund-Ziele. Das bisher übliche Lösen quadratischer Gleichungen nach Rezept (ob mit einer der Formeln oder jeweils mit quadratischer Ergänzung) ist unserer Überzeugung nach ein „aussterbendes Rezept“ (vgl. [Herget 1996].) Entsprechend sind Rechenstab und Logarithmentafel fast „über Nacht“ aus dem Mathematikunterricht verschwunden, als die umfangreichen Berechnungen den Taschenrechnern übertragen werden konnten.

Ungleichungen – langfristige Mindestkompetenzen

	−T (ohne Technologie)	?T	+T (mit Technologie)
01	Für welche x gilt: $x - 2 < 4$	Für welche x gilt: $x-2 < x+3$	Für welche x gilt: $3x + 1 < 2x - 1$
02	Für welche x gilt: $-2x < 4$		Für welche x gilt: $\frac{1}{x-1} \leq 2$
03	Für welche x gilt: $x < x + 1$		Für welche x gilt: $ax < 4$
04	Für welche x gilt: $x < x$		
05		Für welche x gilt: $\|x\| < 1$	Für welche x gilt: $\|x - 2\| < 1$

Bei den Ungleichungen ist beim Einsatz von CAS besonders deutlich eine Verschiebung von der Rechen- zur Visualisierungskompetenz beobachtbar.

Differenzieren – langfristige Mindestkompetenzen

	−T (ohne Technologie)	?T	+T (mit Technologie)
01	Diff. nach $x : y = x^4$		
02	Diff. nach $x : y = 7x^2 + 3x + 1$		
03	Diff. nach $x : y = \frac{1}{x^2}$		
04	Diff. nach $x : y = 3$		
05	Diff. nach $x : y = \sqrt{x}$		
06	Diff. nach $x : y = \sin x$	Diff. nach $x : y = x^2 + \cos x$	Diff. nach $x : y = x \sin x$
07		Diff. nach $x : y = 2 \cos x$	Diff. nach $x : y = \sin^2 x$
08		Diff. nach $x : y = 3 \sin 2x$	Diff. nach $x : y = \frac{\sin x}{x}$
09	Diff. nach $x : y = e^x$	Diff. nach $x : y = e^{2x}$	Diff. nach $x : y = 2^x$
10	Diff. nach $x : y = \ln x$		
11	Diff. nach $x : y = \|x\|$		

Diese Tabelle kennzeichnet einen weiteren Schwerpunkt der zukünftigen Entwicklung: Gerade im klassischen Analysisunterricht dominieren die Rechenfertigkeiten. Daher ist hier besonderer Veränderungsbedarf beim Einsatz moderner Technologie gegeben.

Schlussbemerkung und Bitte

Wie eingangs erwähnt, möchten wir diesen Beitrag als Anstoß für eine möglichst breite Diskussion verstanden wissen. Unser Ziel war es nicht, eine feinsinnige, möglichst unangreifbare didaktische Analyse vorzulegen – uns ging es vielmehr um eine pragmatische, knappe Darstellung unserer augenblicklichen Einschätzung zum sehr vielschichtigen Problem der handwerklichen Rechenfähigkeiten. Wir sind uns bewusst, welche Herausforderung unsere Position bedeutet und wie sehr damit vertraute Grundpfeiler des Mathematikunterrichts ins Wanken geraten. Schreiben Sie uns Ihre Meinung.

Literatur

[1] Herget, Wilfried, 1996.: Rettet die Ideen! – Rettet die Rezepte? In: Hischer, H. / Weiß, M. (Hrsg.): Rechenfertigkeit und Begriffsbildung – Zu wesentlichen Aspekten des Mathematikunterrichts vor dem Hintergrund von Computeralgebrasystemen. Hildesheim: Franzbecker, S.156-169.

[2] Herget, Wilfried, 1999: Wie viel Termumformung braucht der Mensch? – Taschencomputer und Mathematikunterricht. In: Amelung, Udo (Hrsg.): Der TI-92 im Mathematikunterricht. Pfingsttagung 1998. Zentrale Koordination Lehrerausbildung, ZKL-Texte Nr. 7, Westfälische Wilhelms-Universität, Münster, S. 3-19.

[3] Heugl, Helmut, 1999: The necessary fundamental algebraic competence in the age of Computeralgebra Systems. Proceedings of the 5th ACDCA Summer Academy, 1999, `http://www.acdca.ac.at`.

[4] Kutzler, Bernhard, 1996: Symbolrechner TI-92 (Computeralgebra im Taschenformat). Bonn: Addison-Wesley.

[5] Kutzler, Bernhard, 1998: Einführung in den TI-89. Hagenberg: bk teachware.

[6] Kutzler, Bernhard, 1999: Der algebraische Taschencomputer als pädagogisches Werkzeug. Bonn: Profil – Zeitschrift des Deutschen Philologenverbandes, März, April 1999. Auch: `http://www.kutzler.com`.

[7] Kutzler, Bernhard & Kokol-Voljc, Vlasta, 2000: Introduction to Derive 5. Hagenberg: Soft Warehouse Europe.

[8] Lehmann, Eberhard, 1999a: Terme im Mathematikunterricht unter Verwendung von Computergrafik und Computeralgebra, Hannover: Schroedel-Verlag.

[9] Lehmann, Eberhard, 1999b: Neue Aspekte im Unterricht über Terme durch Einsatz von Computeralgebra-Systemen. Erstellt für das BLK-Programm SINUS im Auftrag des IPN Kiel.

Wilfried Herget (Halle), `herget@mathematik.uni-halle.de`
Helmut Heugl (Wien), `hheugl@netway.at`
Bernhard Kutzler (Leonding), `b.kutzler@eunet.at`
Eberhard Lehmann (Berlin), `mirza@berlin.snafu.de`

Unterrichtsmaterialien im Internet

Das Internet ist bekanntlich ein ziemlich chaotisches Sammelsurium von Materialien, welche nicht alle einen gleich hohen Qualitätsstandard haben. Ferner sind einige Internetseiten recht kurzlebig. Aus diesem Grund haben wir nun damit begonnen – wie im letzten Rundbrief bereits angekündigt – Materialien, welche im Internet zur Verfügung gestellt werden und welche sich mit dem Einsatz von Computeralgebra im Unterricht beschäftigen, zu begutachten.

Bei positiver Begutachtung werden wir diese Materialien in einer regelmäßigen Rubrik des Computeralgebra-Rundbriefs mit einer kurzen Inhaltsangabe veröffentlichen. Diese Veröffentlichungen werden mit dem Datum der Annahme versehen, da wir natürlich nicht garantieren können, wie lange diese Materialien unter der jeweils veröffentlichten Adresse zur Verfügung stehen.

Ich möchte Sie nochmals ausdrücklich auffordern, mir Internetseiten vorzuschlagen, welche für Schüler, Lehrer, Studenten oder Hochschullehrer interessant sein könnten. Scheuen Sie sich auch nicht davor, eigene Seiten zu benennen. Wir wollen gerne die besten Seiten unseren Mitgliedern empfehlen.

Heute möchte ich Sie zunächst auf die (teilweise offiziellen, teilweise inoffiziellen) Internetseiten der Ministerien und Landesinstitute hinweisen, welche in meinem Artikel über die Thurnau-Tagung auf S. 7–9 angegeben sind. Diese Seiten enthalten viele Informationen über den Einsatz von Computeralgebra bzw. von Computern im allgemeinen in den einzelnen Bundesländern, einige auch noch sehr viel mehr.

Weitere offizielle Adressen der Bundesländer wie beispielsweise Bildungsserver finden Sie bei der Adresse `http://www.mathematik.uni-kassel.de/~koepf/bundeslaender.html`, welche auch von unserer Homepage `http://www.gwdg.de/~cais` aus erreicht werden kann.

Als Modell einer Veröffentlichung im Rundbrief stelle ich Ihnen heute die Seite des österreichischen TI-DERIVE-Projektes vor.

`http://www.acdca.ac.at` 26. September 2000

Dies ist die Homepage des Vereins ACDCA (*Austrian Center for Didactics of Computer Algebra*). Hier sind alle Materialien, welche im Zusammenhang mit den drei offiziellen Schulprojekten in Österreich (CAS I (Derive) 1992/93, CAS II (TI) 1997/98, CAS III (TI) 1999/2000) entstanden, gesammelt und stehen zum Download bereit.
Zu diesen Materialien gehören Projektevaluierungen und -berichte genauso wie eine nach Klassenstufen sowie nach Themen abrufbare Sammlung von Unterrichtsmaterialien. Diese Datenbank umfaßt sowohl die Materialien zum Herunterladen als auch eine Sammlung von 75 Büchern und 22 Handreichungen zum Thema.
Diese Seite ist für jeden hilfreich, der mit einem der TI-Computeralgebrarechner oder mit Derive arbeitet. Andere Computeralgebrasysteme spielen keine Rolle.

Wolfram Koepf (Kassel)

Berichte über Arbeitsgruppen

Das SINGULAR-Projekt an der Universität Kaiserlautern

Computeralgebra wird an der Universität Kaiserslautern an den Fachbereichen Mathematik und Informatik betrieben. Fachbereichsübergreifend existiert das Zentrum fuer Computeralgebra der Fachbereiche Mathematik, Informatik und Elektrotechnik, über das zu einem späteren Zeitpunkt gesondert berichtet wird.

Im Beitrag auf Seite 11 unter der Rubrik "Neues über System und Hardware" wird das SINGULAR-Projekt beschrieben.

Kurze Geschichte von SINGULAR
Die ersten Ueberlegungen, ein Paket fuer Standardbasen in lokalen Ringen zu implementieren, stammt aus dem Jahr 1984, als Greuel und Pfister in einer gemeinsanmen Arbeit die Verallgemeinerung einer Vermutung von K. Saito numerisch so formulierten, dass sie rechnerisch widerlegbar wurde.

- 1987 implementierten Pfister und Schoenemann in Berlin das Programm Buchmora in Modula-2 für Atari. Widerlegung obiger Vermutung.
- 1989 Umbenennung von Buchmora in SINGULAR und Weiterentwicklung als gemeinsames Projekt von Greuel (Kaiserslautern) und Neuendorf/Pfister/Schönemann (Berlin).
- 1990 Portierung nach Unix, erstes User Manual, "Heimatstadt" Kaiserslautern.
- 1993 Neuimplementierung in C/C++, SINGULAR-Programmiersprache, Bibliotheken.
- 1996 Uni- und multivariate Polynomfaktorisierung, gcd.
- 1997 SINGULAR 1.0: Polynomfaktorisierung, Auflösungen, communication links
- 1998 SINGULAR 1.2: schneller, Primärzerlegung, Ring-Normalisierung
- 2000 Singular 2.0: viel schneller, numerische Datentypen und Algorithmen.

Mitarbeiter:
Verantwortliche Leiter: G.-M. Greuel, G. Pfister, H. Schönemann
Aktuelle Mitarbeiter: O. Bachmann, W. Pohl
Weitere Mitarbeiter:
Kaiserslautern: T. Bayer, A. Frühbis-Krüger, C. Gorzel, K. Krüger, M. Lamm, C. Lossen, T. Nüssler, J. Schmidt, M. Schulze, R. Stobbe, M. Wenk
Saarbrücken: W. Decker, A. Heydtmann,. M. Mesollen, C. Theiss

Cottbus: B. Martin, T. Siebert
Berlin: H. Grassmann, W. Neumann
Mainz: S. Endrass
Dortmund: D. Hillebrand
Valladolid: P. Gimenz, I. Faran
La Laguna: I. Bermejo
Darüberhinaus werden in Diplomarbeiten ständig neue Bibliotheken sowie Teile des Kerns zu SINGULAR hinzugefügt.

Förderung:
Das SINGULAR-Projekt wurde durch Mittel der Volkswagen-Stiftung, durch zwei DFG Schwerpunkte sowie durch Mittel der Stiftung Rheinland-Pfalz für Innovation gefördert.

Die SINGULAR-Gruppe unterstützt die Philosophie der freien Software und deshalb wird SINGULAR nicht-kommerziell und frei erhältlich bleiben.

Arbeitsbereich *Computational Mathematics* an der Universität Gesamthochschule Kassel

Unsere Arbeitsgruppe `<http://www.mathematik.uni-kassel.de/~compmath>`, welche im letzten Rundbrief vorgestellt wurde, hat sich – wie bereits angekündigt – zum Wintersemester 2000/2001 durch eine Neuberufung vergrößert.
Neue Mitglieder der Arbeitsgruppe sind

- Prof. Dr. Hans-Georg Rück (Zahlentheorie, elliptische Kurven, Abelsche Varietäten, Drinfeld-Moduln, Public-Key-Kryptographie)
- Dr. Markus Wessler (Kryptographie mit elliptischen und hyperelliptischen Kurven).

Wir möchten ferner auf eine freie Mitarbeiterstelle (1/2 BAT IIA) aufmerksam machen. Details zu der zu besetzenden Stelle gibt es bei `http://www.uni-kassel.de/pvabt3/stellen/extern/841.ghk`.

Computeralgebra am IWR Heidelberg

Seit der erstmaligen Vorstellung im Jahr 1994 hat sich die Zusammensetzung wie auch die Ausrichtung der Arbeitsgruppe Computeralgebra am IWR (Interdisziplinäres Zentrum für Wissenschaftliches Rechnen) an der Universität Heidelberg erheblich gewandelt. Ursache hierfür war die Wegberufung der beiden Dozenten G. Hiß und G. Malle nach Aachen bzw. Kassel und das Hinzustoßen neuer Mitglieder zu der Gruppe. Daher soll in diesem Heft der alte Bericht durch eine aktuelle Information ersetzt werden.
Mitglieder der Arbeitsgruppe:
Leitung: Prof. Dr. B. H. Matzat, PD Dr. G. Kemper, PD Dr. P. Müller
Weitere Mitglieder: Dr. M. Dettweiler, J. Hartmann, Dr. J. Klüners, T. Oberlies, Dr. S. Reiter
Homepage: `http://www.iwr.uni-heidelberg.de/iwr/compalg`
Arbeitsgebiete:

- Konstruktive inverse Galoistheorie. Dabei Erstellung einer Datenbank für Polynome mit gegebener Galoisgruppe (in Zusammenarbeit mit G. Malle in Kassel)
- Arithmetische Anwendungen endlicher Gruppentheorie, z.B. auf Hilbertmengen
- Algorithmische Invariantentheorie. Dabei Erstellung einer Datenbank für Invarianten endlicher Gruppen
- Konstruktive Differential-Galoistheorie: Algorithmen und inverses Problem
- Konstruktive Idealtheorie und nichtlineare Gleichungssysteme

Literatur:
[1] H. Derksen, G. Kemper: Computational Invariant Theory. Springer-Verlag (in Vorbereitung)
[2] M. Dettweiler, S. Reiter: An algorithm of Katz and its application to the inverse Galois problem (in [6])
[3] K. Geissler, J. Klüners: Galois group computation for rational polynomials (in [6])
[4] G. Malle, B. H. Matzat: Inverse Galois Theory. Springer-Verlag, Heidelberg 1999
[5] B. H. Matzat, G. M. Greuel, G. Hiß (Eds.): Algorithmic Algebra and Number Theory. Springer-Verlag, Berlin 1999
[6] B. H. Matzat, J. McKay, K. Yokoyama (Eds.): Algorithmic Methods in Galois Theory. Special issue of JSC (erscheint Dez. 2000)
[7] B. H. Matzat, M. van der Put: Iterative differential equations and the Abhyankar conjecture. Preprint 2000
[8] P. Müller: Hilbert's irreducibility theorem for prime degree and general polynomials. Israel J. Math. 109 (1999), 319-337.

Publikationen über Computeralgebra

- Erk, K., Priese, L., *Theoretische Informatik*, Springer Verlag, Berlin, Heidelberg, New York, ISBN 3-540-66192-1, 2000, pp. 433 S., DM 59,00.
- Gramlich, G., Werner, W., *Numerische Mathematik mit Matlab*, dpunkt Verlag, Heidelberg, ISBN 3-932588-55-X, 2000, pp. 456 S., DM 89,00.

  Das Buch wird in diesem Rundbrief auf Seite 36 besprochen.
- Hibbard, A., Levasseur, K., *Exploring Abstract Algebra with Mathematica*, Springer Verlag, Berlin, Heidelberg, New York, ISBN 0-387-98619-7, 1999, pp. 495, DM 8 4,-.
- Kreuzer, M., Robbiano, L., *Computational Commutative Algebra 1*, Springer Verlag, Berlin, Heidelberg, New York, ISBN 3-540-67733-X, 2000,
- Nowottny, D., *Mathematik am Computer*, Springer Verlag, Berlin, Heidelberg, New York, ISBN 3-540-88058-5, 1999, pp. 259, DM 49,90.

  Das Buch wird in diesem Rundbrief auf Seite 37 besprochen.

Besprechungen zu Büchern der Computeralgebra

J. Buchmann, Einführung in die Kryptographie

Springer-Lehrbuch, Springer-Verlag Berlin-Heidelberg-New York 1999,
ISBN 3-540-66059-3, pp.229, DM 49,90.

Das Buch enthält die folgenden Abschnitte:
Ganze Zahlen — Kongruenzen und Restklassenringe — Verschlüsselung — Wahrscheinlichkeit und perfekte Sicherheit — Der DES-Algorithmus — Primzahlerzeugung — Public-Key Verschlüsselung — Faktorisierung — Diskrete Logarithmen — Kryptographische Hash-Funktionen — Digitale Signaturen — Andere Gruppen — Identifikation — Public-Key-Infrastrukturen.
Aus dem Vorwort : „ Ich wende mich in diesem Buch an Leser, die moderne kryptographische Techniken und ihre mathematischen Fundamente kennenlernen wollen, aber nicht über die entsprechenden mathematischen Spezialkenntnisse verfügen. Mein Ziel ist es, in die Basistechniken der modernen Kryptographie einzuführen. Ich setze dabei zwar mathematische Vorbildung voraus, führe aber in die Grundlagen von

linearer Algebra, Algebra, Zahlentheorie und Wahrscheinlichkeitstheorie ein, soweit diese Gebiete für die behandelten kryptographischen Verfahren relevant sind.“
Dem Verfasser ist es gelungen, dieser Zielsetzung gerecht zu werden. Alle zum Verständnis moderner krytopraphischen Techniken erforderlichen Begriffsbildungen werden in einer sehr klaren Sprache erläutert und, wenn möglich, sogleich an kleinen Beispielen verdeutlicht. Besonderer Wert wird auf die Beschreibung der zugrundeliegenden Ideen gelegt. Der Autor beschränkt sich auf die Darstellung derjenigen mathematischen Hilfsmittel, deren Beweis nicht mehr als eine Druckseite beansprucht. In der Summe ergibt sich ein Text von hoher Lesbarkeit.
Wer jedoch zum Beispiel Genaueres über Primzahlerzeugung oder Faktorisierung erfahren möchte, sei auf die Skripten des Autors (mit Volker Müller) unter
`ftp://ftp.informatik.tu-darmstadt/pub/TI/lecture_notes/` verwiesen.
Die Übungsaufgaben (mit Lösungen) sind genau am Text orientiert. Neben Aufgaben, die ein Überdenken der im Text gebrachten Argumente und Beweise erfordern, finden sich viele, die den Leser auffordern, die vorgestellten Techniken an moderat großen Beispielen mit Hilfe eines Computer-Algebra Systems zu erproben.
(Die im Buch erwähnte Programm-Bibliothek LiDIA findet sich unter
`http://www.informatik.tu-darmstadt.de/TI/LiDIA/`.)
Das Buch hat ein ausreichend detailliertes Sachverzeichnis und ein Literaturverzeichnis, in dem u.a. alle wichtigen, weiterführenden Werke über moderne Kryptographie zu finden sind.
Nochmals aus dem Vorwort: „ Es ist nötig, daß die Anwender einschätzen können, ob die benutzten kryptographischen Methoden effizient und sicher genug sind. Dazu müssen sie nicht nur wissen, wie die kryptographischen Verfahren funktionieren, sondern sie müssen auch deren mathematischen Grundlagen verstehen.“
Dies allen zur Beherzigung, die sich der praktischen Seite der Kryptographie zuwenden wollen!
Ein lehrreiches und preiswertes Buch.

H. Meyn (Erlangen)

J. von zur Gathen, J. Gerhard, Modern Computer Algebra

Cambridge University Press, Cambridge, 1999, ISBN 0-521-641764, XIV + 754 Seiten

Das Buch Modern Computer Algebra von J. von zur Gathen und J. Gerhard wird sich sicher zu einem Standardwerk über Computeralgebra entwickeln. Es ist sehr liebevoll aufbereitet mit historischen Einleitungen und Anmerkungen, vielen Beispielen, Bildern und Anwendungen sowie einer Vielzahl von Übungsaufgaben. Das Buch beginnt mit einer Sammlung motivierender Anwendungen, u.a. Modellierung von Cyclohexan-Molekülen und RSA-Verschlüsselung, die später nach der Entwicklung der nötigen Theorie wieder aufgegriffen werden.
Die Kapitel sind nach Wegbereitern der Computeralgebra benannt: Euklid, Newton, Gauß, Fermat und Hilbert. Das erste startet mit den grundlegenden Algorithmen zur Multiplikation und Division (mit Rest). Hauptthema aber ist der Euklidische Algorithmus in verschiedenen Spielarten und Anwendungen wie Kettenbrüche, Partialbruchzerlegung, Resultanten und Subresultanten etc. Alle diese Algorithmen wie auch die späteren sind durch sorgfältige Aufwandsanalysen begleitet. Flankiert wird dieses Kapitel durch Ausflüge in die Kalenderkorrektur, Tonsysteme und die Codierung mit BCH-Codes.
Hauptthema des zweiten Kapitels Newton ist die schnelle Arithmetik. Es beginnt mit einer Darstellung des Karatsuba-Algorithmus, der diskreten und schnellen Fouriertransformation bis zum Algorithmus von Schönhage und Strassen. Danach folgt eine Einführung des Newtonverfahrens mit Konvergenzverhalten (Julia-Mengen), Polynomwertberechnung und Interpolation. Anschließend wird mit dem schnellen Euklidischen Algorithmus das Hauptthema des ersten Kapitels weitergeführt. Es folgt ein Kapitel über schnelle Matrixmultiplikation mit dem (ersten) Algorithmus von Strassen sowie Verfahren für spezielle Matrizen, z. B. dünn besetzte (Wiedemann). Für die fortgeschritteneren und für die Komplexitätstheorie interessanten Algorithmen von Strassen und von Winograd wird auf die Literatur verwiesen. Beschlossen wird das Kapitel mit der stetigen Fouriertransformation mit Anwendung auf Bildkompression.
Das Kapitel Gauß ist der Faktorisierung von Polynomen gewidmet. Es enthält zunächst Algorithmen zur Faktorisierung von Polynomen über endlichen Körpern, z. B. die Algorithmen von Berlekamp und Cantor-Zassenhaus. Weiter die Henselsche Liftungsmethode in Polynomringe über p-adischen Zahlen,

die zum Zassenhaus-Algorithmus zur Zerlegung ganzzahliger Polynome führt. Schließlich wird die Gitterbasisreduktion nach Lenstra-Lenstra-Lovasz beschrieben und mit deren Hilfe die Polynomialität der Primzerlegung ganzzahliger Polynome gezeigt. Als weitere Anwendungen werden u. a. simultane diophantische Approximationen behandelt und Angriffsmöglichkeiten auf Cryptosysteme vorgeführt.
Im Kapitel Fermat werden Primzahltest und Primzerlegungsalgorithmus für ganze Zahlen behandelt. Der erste Paragraph führt über das Studium von Carmichael-Zahlen zum Primzahltest von Solovay und Strassen. Der folgende enthält elementare Faktorisierungsalgorithmen wie Pollards ϱ - und (p-1)-Methode, Dixon's random square Methode (leider ohne Verfeinerung zum quadratischen Sieb) sowie eine Beschreibung der elliptischen Kurven-Methode von Lenstra. Beschlossen wird dieses Kapitel mit der Vorstellung von diversen Verschlüsselungssystemen, insbesondere dem auf obigen Resultaten aufbauenden RSA-Schema.
Das letzte Kapitel ist nichtlinearen algebraischen Gleichungssystemen gewidmet. Nach Einführung von Gröbnerbasen wird der Buchbergersche Algorithmus behandelt, und es werden geometrische Anwendungen gezeigt. Bei der Komplexitätsuntersuchung wird festgestellt, daß dieser grundlegende Algorithmus der Klasse der doppelt-exponentiellen Algorithmen angehört, wodurch seine Reichweite im allgemeinen sehr eingeschränkt ist. Es folgen Abschnitte über symbolische Integration, u. a. mit dem Algorithmus von Rothstein und Trager, und symbolische Summation mit dem Gosper-Algorithmus. Unter den Anwendungen sind u. a. Petri-Netze und automatisches Beweisen (von Formeln) zu finden. Schließlich wird zum Abschluß das Cyclohexan-Beispiel aus der Einführung ausführlich behandelt.
Das Buch deckt den Standardstoff einer ein- bis zweisemestrigen Vorlesung über Computeralgebra ab. Unter anderem wegen der vielen Beispiele, historischen Anmerkungen und Verweise ist es als Begleitlektüre sehr zu empfehlen. Da insbesondere die Themen der ersten beiden Kapitel sehr ausführlich behandelt und illustriert werden, ist dieses Buch auch zum Selbststudium geeignet. Es kann weiter Impulse für Vorlesungen an Fachhochschulen sowie für Arbeitsgemeinschaften an Schulen geben. Ich wünsche diesem Buch einen breiten Leserkreis.

B. H. Matzat (Heidelberg)

G. Gramlich, W. Werner, Numerische Mathematik mit Matlab
Eine Einführung für Naturwissenschaftler und Ingenieure

dpunkt Verlag, Heidelberg, 2000, ISBN 3-932588-55-X, 456 S., DM 89,00.

MATLAB erfreut sich in Wissenschaft und Industrie immer größerer Beliebtheit für numerische Berechnungen aller Art. Zur Ergänzung des Basisprogramms existieren zahlreiche sogenannte *Toolboxes* für speziellere Aufgaben wie z.B. der Simulation dynamischer Systeme. Die *Symbolic Math Toolbox* stellt eine Schnittstelle zu dem Computeralgebrasystem MAPLE zur Verfügung und gestattet damit auch symbolische Berechnungen. Sie gehört zum Lieferumfang der Studentenversion von MATLAB, ansonsten muß sie separat gekauft werden.

Wie der Titel klar anzeigt, möchte das Buch Naturwissenschaftler und Ingenieure einführen in die numerische Mathematik – was hier vorrangig numerische Lineare Algebra bedeutet – unter Verwendung von MATLAB. Dies ist im wesentlichen auch sehr gut gelungen. Im Gegensatz zu vielen Büchern mit ähnlicher Zielsetzung erschöpft sich das vorliegende nicht darin, einfach ein paar Kochrezepte für Standardaufgaben zusammenzustellen, sondern (im Rahmen des auf diesem Niveau Möglichen) wird auch immer wieder auf die numerischen Stabilität von Verfahren oder die Konditionierung von Problemen eingegangen.

Von einer kurzen Einleitung abgesehen gliedert sich das Buch in neun Kapitel und drei Anhänge. Zunächst werden die Grundlagen der Anwendung von MATLAB eingeführt, danach Grundbegriffe der Linearen Algebra bzw. des Numerischen Rechnens. Die verbleibenden sechs Kapitel behandeln nacheinander lineare Gleichungssysteme, lineare Ausgleichsrechnung, Eigenwertprobleme, Interpolation, nichtlineare Gleichungen und die nichtlineare Ausgleichsrechnung. Jedes Kapitel enthält eine Reihe von Übungsaufgaben sowohl zu den behandelten theoretischen Konzepten als auch zu deren praktischen Anwendung in MATLAB. Beweise sind kaum zu finden, dafür aber zahlreiche Verweise auf weiterführende Literatur.

Von den drei Anhängen ist der erste dem symbolischen Rechnen mit der *Symbolic Math Toolbox* gewidmet; der zweite gibt einen Überblick über die Eigenschaften der wichtigsten Matrixfaktorisierungen und der dritte enthält zahlreiche Internetadressen zu mathematischer Software, Fachverbänden (unter anderem auch die Fachgruppe Computeralgebra!) oder Zeitschriften und Programmbibliotheken zur Numerischen Mathematik.

Dem Buch liegt eine CD bei, die zu sämtlichen im Buch besprochenen Programmen den Sourcecode enthält. Die Benutzung der CD setzt allerdings eine vorhandene MATLAB-Installation voraus.

Etwas störend sind die zahllosen Fußnoten, die mitunter den Lesefluß doch ziemlich hemmen.[3] An einigen Stellen haben sich die beiden Koautoren wohl nicht genügend abgestimmt. So ist es sicherlich nicht nötig, innerhalb von fünf Seiten zweimal zu erläutern, was A^H bedeutet (beide Male selbstverständlich in einer Fußnote...). Von solchen Kleinigkeiten abgesehen kann das Buch ohne Einschränkungen für entsprechende Vorlesungen empfohlen werden. Ein Student, der es komplett durchgearbeitet hat, sollte sowohl über vernünftige Grundkenntnisse in Numerischer Mathematik als auch über einige praktische Erfahrung im Umgang mit MATLAB verfügen.

Werner M. Seiler (Mannheim)

D. Nowottny, Mathematik am Computer

Springer-Verlag, 1999, ISBN 3-540-66058-5, 250 p., DM 49.90.

Mathematik am Computer versucht, eine kurze Einführung in drei wichtige und typische Programmsysteme, nämlich LaTeX, MATLAB und Maple zu geben. Das Buch eignet sich z.B. als Begleittext für Kurse in Hochschulrechenzentren, die Studierende in die Benutzung dieser mathematischen Software einführen. Es gibt keine, wie heute meist üblich, dem Buch beigefügte CD, die den Programmcode der zahlreichen Beispiele enthält; diese können jedoch von einem im Buch angegebenen FTP-Server heruntergeladen werden. Für den ernsthaft Interessierten wird diese kleine zusätzliche Mühe kein Hindernis darstellen. Besonders positiv hervorzuheben sind die vielen interessanten, konkreten Aufgabenstellungen und Visualisierungen, die behandelt werden; diese allein sind die Anschaffung des Buches wert, auch wenn etliche dieser Beispiele bereits -verstreut- an anderer Stelle publiziert sind. Es soll aber nicht verschwiegen werden, dass es für jedes der drei Programme etliche bessere und ausführlichere -und in der Regel aber auch wesentlich teuerere- Einführungen gibt. Wer sich gezielt für LaTeX, MATLAB oder Maple und deren Feinheiten interessiert, sollte zu einer anderen Quelle greifen. Deutliche Schwächen hat das Buch, wenn es um Erklärungen von Phänomenen geht, auf die man z.B. in der Computeralgebra stößt. So wird (auf S. 111) recht laienhaft darüber spekuliert, warum Maple als Lösung der Gleichung

$$\ln(x) = \ln(\frac{2}{x+1}) \tag{1}$$

neben $x = 1$ auch $x = -2$ liefert. Der Hinweis, dass die von Maple berechnete Lösung -2 korrekt ist, wenn man den Logarithmus als Funktion auf den komplexen Zahlen betrachtet, fehlt. Pikant ist, dass der Autor unter der treffenden Überschrift "5.5.4 Achtung Fehler" einen eigenen Eingabefehler der Maple-Integration anlastet: „Gelegentlich kommt es vor, daß Maple falsche Ergebnisse liefert ...". Den Term $\sqrt{\frac{-1}{x^2-1}}$ realisiert er in Maple durch den Befehl `sqrt(-1/x^2-1)` anstelle von `sqrt(-1/(x^2-1))`(beachte die Klammerung!) und wundert sich, warum das Integral über diesen Term nicht -wie erwartet- arcsin(x) ergibt. Es ist ja durchaus bekannt, dass Maple wie auch andere Computeralgebrasysteme gewisse Integrale falsch berechnen; das hier zur Demonstration verwendete Beispiel ist jedoch lediglich ein Syntaxfehler bei der Eingabe. Der ausgesprochen positive Eindruck, den die vielen schönen Beispiele des Buches hinterlassen, wird etwas getrübt durch derartige inhaltliche Mängel.

Wilhelm Werner (Heilbronn)

[3] *Authors who are interested in good exposition should avoid footnotes whenever possible, since footnotes tend to be disctracting;* Donald E. Knuth, The TeXbook.

Lehrveranstaltungen zu Computeralgebra im WS 2000/2001

- **Rheinisch–Westfälische Technische Hochschule Aachen**
  Einführungspraktikum in das Formelmanipulationssystem Maple, G. Hiß, U. Klein, V. Dietrich, P2
  Praktikum: Programmieren in Maple, G. Hiß, U. Klein, P4
  Arbeitsgemeinschaft zu speziellen Problemen mit Maple, V. Dietrich, U. Klein, E. Görlich, Ü2

- **Universität Bayreuth**
  Diskrete Strukturen, A. Kerber, V4+Ü2
  Kodierungstheorie, A. Kerber, V2+Ü2

- **Freie Universität Berlin**
  Computeralgebra, K. Gatermann, V2

- **Technische Universität Berlin**
  Konstruktive Zahlentheorie II, M. Pohst, V4
  Seminar Algorithmische Zahlentheorie und Algebra, M. Pohst, S2

- **Technische Universität Darmstadt**
  Algorithmische Zahlentheorie, J. Buchmann, N.N., V2+Ü2
  Kryptographie II, I. Biehl, V4+Ü2
  Praktikum Public Key Infrastruktur und Anwendungen, J. Buchmann, M. Ruppert, P4
  Praktikum Weiterentwicklung von LiDIA, J. Buchmann, N.N., P4
  Praktikum Elektronische Zahlungssysteme, D. Kuegler, H. Vogt, P4
  Praktikum Implementation und Design von kryptographischen und zahlentheoretischen Algorithmen, die elliptische Kurven verwenden, J. Buchmann, H. Baier, B. Henhapl, P3

- **Universität Dortmund**
  Seminar symbolisches und numerisches Lösen von Gleichungssystemen, H. M. Möller, S2

- **Universität Erlangen-Nürnberg**
  Computeralgebra, V. Strehl, V3 + Ü1
  Kryptografie I, H. Meyn, V2
  Hauptseminar Kryptografie, M. Bauer, S2
  Differentialalgebra, W.-D. Geyer, V2
  Codes, H. Kurzweil, V4 + Ü2

- **Fachhochschule Flensburg**
  Angewandte Mathematik mit Maple V fuer VerfahrenstechnikerInnen im Hauptstudium, P. Thieler, V2+P2
  Iterationsverfahren mit Maple 6 fuer MathematikerInnen im Hauptstudium, P. Thieler, V2+P2
  Mathematik IV - Angewandte Mathematik mit Maple 6 fuer Studierende der Medieninformatik und der Technischen Informatik, P. Thieler, V2+P2
  DV-Anwendungssysteme Teil II - Maple 6 fuer MathematikerInnen im Grundstudium, P. Thieler, V2+P2

- **Universität Hamburg**
  Einführung in die Computeralgebra, K. Gatermann, Blockkurs

- **Technische Hochschule Hamburg-Harburg**
  Diskrete Mathematik Ia, K.-H. Zimmermann, V2+Ü1
  Diskrete Mathematik II, K.-H. Zimmermann, V3+Ü2
  Diskrete Mathematik III, P. Batra, V2

- **Martin-Luther-Universität Halle(Saale)**
  Mathematik mit Mathcad, H. Benker, S2
  Statistik mit Mathematica und Mathcad, H. Benker, H .Dietrich,H. Fritsch, S2

- **Universität Heidelberg**
  Codierungstheorie, P. Müller , V2

- **Universität Kaiserslautern**
  Einführung in die Kryptographie, A. Guthmann, V2
  Seminar Kryptographie , A. Guthmann, S2
  Seminar Algebraische Geometrie und Computeralgebra, G.-M. Greuel, G. Pfister, S2

- **Pädagogische Hochschule Karlsruhe**
  Algorithmen - von Hammurapi bis Gödel, J. Ziegenbalg, V2
  Codierung und Kryptographie, J. Ziegenbalg, V2

- **Universität-Gesamthochschule Kassel**
  Ausgewählte Kapitel der Computeralgebra, W. Koepf, V4+Ü2
  Seminar: Mathematik für Grund-, Haupt- und Realschule, W. Koepf, S2
  Oberseminar: Computational Mathematics, W. Koepf, G. Malle, G. Rück, OS1
  Einführung in Computeralgebrasysteme, R. Schaper, V2+Ü2

- **Universität Köln**
  Primzahlen und Faktorisierung, N. Klingen S2

- **Universität Leipzig**
  Einführung in das Symbolische Rechnen, H.-G. Gräbe, V2 + Ü1
  Praktikum Symbolisches Rechnen, H.-G. Gräbe, Blockpraktikum
  Geometrie mit dem Computer, H.-G. Gräbe, V2

- **Universität Linz, Research Institute for Symbolic Computation**
  Einführung in die Computer Algebra, F. Winkler, V2+Ü2
  Computer Analysis, P. Paule, V2
  Algorithmische Kombinatorik, P. Paule, V2
  Computeralgebrasysteme (DERIVE, TI-89/92) als didaktische Werkzeuge im Mathematikunterricht, B. Kutzler, V2
  Projektseminar Computer Algebra, F. Winkler, S2

- **Technische Universität München**
  Computeralgebra 1, M. Kaplan, V4

- **Universität Oldenburg**
  Seminar zur Computeralgebra, W. Schmale, S2

- **Universität Tübingen**
  Gröbnerbasen, C. Schwarzweller, V2 + Ü1
  Seminar Computeralgebra, R. Loos, S2

- **Universität Ulm**
  Computer-Algebra Praktikum für Physiker, G. Baumann, V2+P4

- **Universität Würzburg**
  Physik per Computer, Programmieren physikalischer Probleme mit C und Mathematica, W. Kinzel, V4

- **ETH Zürich**
  Computer Algebra I, T. Mulders, V2 + U1

Kurze Mitteilungen

- **Personelle Veränderungen im ISSAC Steering Committee**

  Auf der ISSAC 2000 in St. Andrews wurde Keith Geddes (Univ. London/Ontario), einer der Väter des Computeralgebrasystems MAPLE, in das Steering Committee gewählt. Turnusmäßig schied dafür D. Saunders aus. Der Vorsitzende des Komitees, Robert Corless (Univ. London/Ontario), trat zurück zugunsten seines Stellvertreters, W. Küchlin (Univ. Tübingen). Damit sind jetzt im Komitee neben den beiden genannten Mitgliedern noch Barry Trager (IBM, Yorktown Heights), Marc Giusti (Palaiseaux, Frankreich), H.M.Möller (Univ. Dortmund) und George Labahn (Univ. London/Ontario).

- Folgende Stelle ist zu besetzen:

  Im Fachbereich Mathematik/Informatik baldmöglichst
  Wiss. Mitarbeiter/in (BAT II a)
  Computational Mathematics

  halbtags, befristet für zunächst 3 Jahre mit der Möglichkeit der Verlängerung um weitere 2 Jahre (Qualifikationsstelle gem. 77 HHG i.V.m 57 b (2) Nr. 1 HRG; Promotionsmöglichkeit).
  Aufgaben: Entwicklung und Programmierung von Algorithmen für Fragestellungen aus den Bereichen der symbolischen Summation bzw. Integration, der orthogonalen Polynome und speziellen Funktionen, der Funktionentheorie oder verwandter Gebiete und die Betreuung von Lehrveranstaltungen in der Computational Mathematics. Bereitschaft zur Promotion in diesem Schwerpunkt wird vorausgesetzt. Voraussetzungen: Dipl.-Mathematiker/in mit guten Kenntnissen im Bereich der computergestützten Mathematik. Nähere Auskünfte erhalten Sie bei Prof. Dr. Koepf, koepf@mathematik.uni-kassel.de. Bewerbungsfrist: 31.10.2000
  Der Frauenförderplan der Universitt Gesamthochschule Kassel verpflichtet zur Erhöhung des Frauenanteils. Frauen werden deshalb nachdrücklich zur Bewerbung aufgefordert. Schwerbehinderte erhalten bei gleicher Eignung den Vorzug. Vollzeitstellen sind grundsätzlich teilbar. Bewerbungen mit den üblichen Unterlagen sind unter Angabe der Kennziffer innerhalb o. g. Frist(en) nach Erscheinen an den Präsidenten der Universitt Gesamthochschule Kassel, 34109 Kassel, zu richten.

- Folgende Stelle ist zu besetzen:

  Wissenschaftlicher Mitarbeiter gesucht

  Im Fachbereich Mathematik der Universität Dortmund ist bei Prof. Dr. H. M. Möller eine Stelle für einen wissenschaftlichen Mitarbeiter bzw. eine wissenschaftliche Mitarbeiterin zum 1.1.2001 zu besetzen. Die Stelle ist auf vier Jahre befristet und bietet Gelegenheit zur Promotion. Der offizielle Ausschreibungstext ist im Dekanat des Fachbereichs Mathematik der Universität Dortmund, 44221 Dormund, erhältlich. Von dem Mitarbeiter/der Mitarbeiterin wird Vertrautheit mit einem Computeralgebrasystem erwartet, gute Numerikkenntnisse und die Bereitschaft, bei der Betreuung von Lehrveranstaltungen auf dem Gebiet des symbolischen und numerischen Rechnens mitzuwirken.

Werbeseite der Fa. Visual Analysis

Aufnahmeantrag für Mitgliedschaft in der Fachgruppe Computeralgebra

(Im folgenden jeweils Zutreffendes bitte im entsprechenden Feld [ ] ankreuzen bzw. ________ ausfüllen.)

Name: ______________________ Vorname: ______________

Akademischer Grad/Titel: ______________________

Privatadresse

Straße/Postfach: ______________________

PLZ/Ort: ______________________ Telefon: ______________

e-mail: ______________________ Telefax: ______________

Dienstanschrift

Firma/Institution: ______________________

Straße/Postfach: ______________________

PLZ/Ort: ______________________ Telefon: ______________

e-mail: ______________________ Telefax: ______________

Gewünschte Postanschrift: [ ] Privatadresse [ ] Dienstanschrift

1. Hiermit beantrage ich zum 1. Januar 200__ die Aufnahme als Mitglied in die Fachgruppe

   Computeralgebra (CA) (bei der GI: 2.2.1).

2. Der Jahresbeitrag beträgt DM 15,00 bzw. DM 18,00. Ich ordne mich folgender Beitragsklasse zu:

   [ ] **15,00 DM**. für Mitglieder einer der drei Trägergesellschaften
   - [ ] GI Mitgliedsnummer: ______________
   - [ ] DMV Mitgliedsnummer: ______________
   - [ ] GAMM Mitgliedsnummer: ______________

   Der Beitrag zur Fachgruppe Computeralgebra wird mit der Beitragsrechnung der Trägergesellschaft in Rechnung gestellt. (Bei Mitgliedschaft bei mehreren Trägergesellschaften wird dies von derjenigen durchgeführt, zu der Sie diesen Antrag schicken.) [ ] Ich habe dafür bereits eine Einzugsvollmacht erteilt. Diese wird hiermit für den Beitrag für die Fachgruppe Computeralgebra erweitert.

   [ ] **15,00 DM**. Ich bin aber noch nicht Mitglied einer der drei Trägergesellschaften. Deshalb beantrage ich gleichzeitig die Mitgliedschaft in der

   [ ] GI [ ] DMV [ ] GAMM.

   und bitte um Übersendung der entsprechenden Unterlagen.

   [ ] **18,00 DM** für Nichtmitglieder der drei Trägergesellschaften. [ ] Gleichzeitig bitte ich um Zusendung von Informationen über die Mitgliedschaft in folgenden Gesellschaften:

   [ ] GI [ ] DMV [ ] GAMM.

3. Die in dieses Formular eingetragenen Angaben werden elektronisch gespeichert. Ich bin damit einverstanden, daß meine Postanschrift durch die Trägergesellschaften oder durch Dritte nach Weitergabe durch eine Trägergesellschaft wie folgt genutzt werden kann (ist nichts angekreuzt wird c. angenommen).

   [ ] a. Zusendungen aller Art mit Bezug zur Informatik, Mathematik bzw. Mechanik.

   [ ] b. Zusendungen durch wissenschaftliche Institutionen mit Bezug zur Informatik, Mathematik bzw. Mechanik.

   [ ] c. Nur Zusendungen interner Art von GI, DMV bzw. GAMM.

Ort, Datum: ______________ Unterschrift: ______________________

Zurück an:	Gesellschaft für Informatik e.V. (GI) Wissenschaftszentrum Ahrstraße 45 53175 Bonn Tel.: 0228-302-149, Fax.: -167 e-mail: gs@gi-ev.de	oder	Deutsche Mathematiker-Vereinigung e.V. (DMV) Mohrenstraße 39, 10117 Berlin Tel.: 030-20377-306, Fax.: -307 e-mail: dmv@wias-berlin.de	oder	Gesellschaft für Angewandte Mathematik und Mechanik e.V. (GAMM) NWF I – Mathematik, Univ. Regensburg Universitätsstr. 31, 96053 Regensburg

Fachgruppenleitung Computeralgebra 1999-2002

Dr. Joachim Apel
Math. Inst. d. Uni. Leipzig
Augustusplatz 10-11
D-04109 Leipzig
0341-97-32239, -32199(Fax)
`apel@mathematik.uni-leipzig.de`
`http://www.mathematik.uni-leipzig.de/MI/apel/apel.html`

Vertreter der GI:
Prof. Dr. Karl Hantzschmann
FB Informatik d. Uni. Rostock
Albert-Einstein-Straße 21
18059 Rostock
Postanschrift: 18051 Rostock
0381-498-3400,-3399(Fax)
`hantzschmann@informatik.uni-rostock.de`

Referent Lehre & Didaktik:
Prof. Dr. Wolfram Koepf
Fachbereich Mathematik/Informatik
Universität Gesamthochschule Kassel
Heinrich-Plett-Str. 40
34132 Kassel
0561/804-4207, Fax.: -4646
`koepf@mathematik.uni-kassel.de`
`http://www.mathematik.uni-kassel.de/~koepf`

Stellv. Sprecher:
Prof. Dr. M. Pohst
FB 3 Mathematik MA 8-1, TU Berlin
Straße des 17. Juni 136
10623 Berlin
030-314-25772,-24015(Sekr.),-21604(Fax)
`pohst@math.tu-berlin.de`

Referent Computational Engineering:
Prof. Dr. Volker Strehl
c/o Lehrstuhl Informatik 8 (Künstl. Intelligenz)
Universität Erlangen-Nürnberg
Am Weichselgarten 9
D-91058 Erlangen
09131-29914,-29907(Sekr.),-29905(Fax)
`strehl@informatik.uni-erlangen.de`

Dr. Johannes Grabmeier
IBM Deutschland
Informationssysteme GmbH
Vangerowstr. 18, Postfach 10 30 68
69020 Heidelberg
06221-59-4329,-4254(Sekr.),-3500(Fax)
`grabm@de.ibm.com`

Referent Chemieanwendungen:
Prof. Dr. A. Kerber
Lehrstuhl II f. Mathematik
Univ. Bayreuth, 95440 Bayreuth
0921-553387, -553385(Fax)
`kerber@uni-bayreuth.de`
`http://www.mathe2.uni-bayreuth.de`

Vertreter der DMV:
Prof. Dr. B. Heinrich Matzat
IWR, Univ. Heidelberg,
Im Neuenheimer Feld 368
69120 Heidelberg
06221-54-8242,-8318(Sekr.),-8850(Fax)
`matzat@iwr.uni-heidelberg.de`

Vertreter der GAMM:
Prof. Dr. Siegfried M. Rump
Informatik III, TU Hamburg-Harburg
Eissendorfer Str. 38
21071 Hamburg
040-42878-3027
`rump@tu-harburg.de`
`http://www.ti3.tu-harburg.de/rump/`

Fachexperte Physik:
Dr. Georg Weiglein
CERN – TH Division
CH-1211 Geneva 23
Schweiz
0041-22-767-2427,-3850(Fax)
`Georg.Weiglein@cern.ch`
`http://home.cern.ch/w/weiglein/www`

Referent Benchmarks:
Prof. Dr. G.-M. Greuel
FB Math. d. Uni. Kaiserslautern
Postfach 3049
D-67653 Kaiserslautern
0631-205-2850,-2339(Sekr.),-3052(Fax)
`greuel@mathematik.uni-kl.de`
`http://www.mathematik.uni-kl.de/~wwwagag/D/Greuel`

Fachexperte Schule:
Heiko Knechtel
An der Tränke 2a
31675 Bückeburg
05722-23628
`HKnechtel@aol.com`

Sprecher:
Prof. Dr. H. Michael Möller
Fachbereich Mathematik
Universität Dortmund
44221 Dortmund
0231-755-3077
`Moeller@math.uni-dortmund.de`

Referent CAIS:
Prof. Dr. Gerhard Schneider
GWDG, Am Faßberg
37077 Göttingen
0551-201-1545,-21119(Fax)
`Gerhard.Schneider@gwdg.de`

Fachexperte Fachhochschulen:
Prof. Dr. Wilhelm Werner
FB TWK der FH Heilbronn
74653 Künzelsau
Daimlerstr.35
07940-1306-21(Sekr.), -20(Fax)
`werner@fh-heilbronn.de`

Verwaltungen der Fachgruppe Computeralgebra

Mitgliederverwaltung der GI, Anzeigenverwaltung:
Gesellschaft für Informatik e.V.
Wissenschaftszentrum
Ahrstr. 45
53175 Bonn
Telefon 0228-302-145 (-164 Anzeigen)
Telefax 0228-302-167
el.Adr.: `gs@gi-ev.de`

Mitgliederverwaltung der DMV:
Deutsche Mathematiker
-Vereinigung, Geschäftsstelle
Mohrenstraße 39
10117 Berlin
Telefon 030-20377-306
Telefax 030-20377-307, el.Adr.:
`dmv@wias-berlin.de`

Mitgliederverwaltung der GAMM:
Gesellschaft für Angewandte
Mathematik und Mechanik e.V.
NWF I – Mathematik,
Univ. Regensburg
Universitätsstr. 31
96053 Regensburg
`http://www.uni-regensburg.de/Fakultaeten/nat_Fak_I/Mennicken/gamm/vorstand.html`

Impressum

Computeralgebra–Rundbrief Herausgegeben von der Fachgruppe Computeralgebra der GI (2.2.1), DMV und GAMM, Redaktionsschluß 28.02 und 30.09. Anschrift: Dr. Ulrich Schwardmann, Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG), Am Fassberg, 37077 Göttingen, Telefax: 0551-21119, Telefon: 0551-201-1542, Internet: `uschwar1@gwdg.de` , ISSN 0933-5994.
Mitglieder der Fachgruppe Computeralgebra erhalten je ein Exemplar dieses Rundbriefs im Rahmen ihrer Mitgliedschaft. Exemplare darüber hinaus bzw. außerhalb der Mitgliedschaft können über die GI bezogen werden.
WWW-Server der Fachgruppe Computeralgebra mit URL: `http://www.gwdg.de/~cais`,
Konferenzankündigungen, Mitteilungen und einzurichtende Links bitte an: `cais@gwdg.de`
CA-Diskussionsliste der Fachgruppe: `cais-l@rz.uni-karlsruhe.de` (Anm.: Subskriptionswunsch an `cais@gwdg.de`)