Kommentierte Literaturliste zur Vorlesung

Computeralgebra

von WOLFGANG K. SEILER

Wintersemester 2001/2002, Universität Mannheim

Eine gute erste Übersicht über das Gesamtgebiet, die allerdings für weniger triviale Sätze oft auf die Literatur verweist, ist

J.H. DAVENPORT, Y. SIRET, E. TOURNIER: Computer Algebra. Systems and algorithms for algebraic computation, *Academic Press,* 1988, ^{2}1993

Einen Überblick in unabhängig voneinander lesbaren Einzelkapiteln verschiedener Autoren, die meist ohne zu sehr in die Tiefe zu gehen zumindest die wichtigsten Aspekte ihres Themas vollständig behandeln, ist

ARJEH M. COHEN, HANS CUYPERS, HANS STERK [HRSG.]: Some Tapas of Computer Algebra, *Springer,* 1999

Ähnlich aufgebaut, wenn auch deutlich weniger homogen ist

B. BUCHBERGER, G.E. COLLINS, R. LOOS: Computer Algebra. Symbolic and Algebraic Computation, *Springer Wien,* 1982, ^{2}1983,

dessen Artikel trotz ihres hohen Alters teilweise immer noch sehr interessant zu lesen sind.

Deutlich umfangreicher, aber ebenfalls um eine Einführung in das Gesamtgebiet bemüht, sind die Bücher

JOACHIM VAN ZUR GATHEN, JÜRGEN GERHARD: Modern Computer Algebra, *Cambridge University Press,* 1999,

bei dem man sich nicht von den motivierenden Beispielen abschrecken lassen sollte, und

K.O. GEDDES, S.R. CZAPOR, G. LABAHN: Algorithms for Computer Algebra, *Kluwer,* 1992

Die drei folgenden Bücher legen das Schwergewicht auf das Rechnen mit Polynomen, behandeln also beispielsweise keine symbolische Integration:

FRANZ WINKLER: Polynomial Algorithms in Computer Algebra, *Springer Wien,* 1996

BHUBANESWAR MISHRA: Algorithmic Algebra, *Springer,* 1993

RICHARD ZIPPEL: Effective Polynomial Computation, KLUWER, 1993

Von ähnlicher Natur, aber mit einem deutlichen Übergewicht der (teilweise vom Autor stammenden) Subresultantenalgorithmen ist

ALKIVAIDIS G. AKRITAS: Elements of Computer Algebra with Applications, *Wiley,* 1989

Die mathematischen Grundlagen des EUKLID*ischen Algorithmus sowie der Algorithmen zur Polynomfaktorisierung sind einfach und sehr gut lesbar dargestellt in*

MAURICE MIGNOTTE: Mathematics for Computer Algebra, *Springer,* 1992

Um Anwendungen der Computeralgebra in Algebra und Zahlentheorie geht es in

BERNADETTE PERRIN-RIOU: Algèbre, arithmétique et maple, *Cassini,* 2000

Eine ausführliche Darstellung der symbolischen Integration transzendenter Funktionen findet man in

MANUEL BRONSTEIN: Symbolic Integration I, *Springer,* 1997

Da der zweite Band noch nicht erschienen ist, muß man für die Integration algebraischer Funktionen weiterhin zurückgreifen auf die deutlich schwieriger zu lesenden lecture notes

JAMES HAROLD DAVENPORT: On the Integration of Algebraic Functions, *Springer Lecture Notes in Computer Science* **102**, 1979

Zwei Bücher, die sich speziell mit GRÖBNER*-Basen in allen ihren Varianten befassen, insbesondere also auch mit nichtlinearen Gleichungssystemen, sind*

BERND STURMFELS: Gröbner Bases and Convex Polytopes, *American Mathematical Society,* 1996

WILLIAM W. ADAMS, PHILIPPE LOUSTAUNAU: An Introduction to Gröbner Bases, *American Mathematical Society,* 1994

Eher um die Anwendung von GRÖBNER*-Basen und ähnlichen Methoden geht es in den folgenden drei Büchern, von denen insbesondere das erste sehr einfach zu lesen ist:*

DAVID COX, JOHN LITTLE, DONAT O'SHEA: Ideals, Varieties, and Algorithms. An Introduction to Computational Algebraic Geometry and Commutative Algebra, *Springer,* 1992

DAVID COX, JOHN LITTLE, DONAT O'SHEA: Using Algebraic Geometry, *Springer,* 1998

THOMAS BECKER, VOLKER WEISPFENNING, HEINZ KREDEL: Gröbner Bases. A Computational Approach to Commutative Algebra, *Springer,* 1993

Die theoretischen Grundlagen für das exakte Rechnen mit reellen Zahlen mit Anwendungen in der rechnerischen Geometrie findet man in

RICCARDO BENEDETTI, JEAN-JACQUES RISLER: Real algebraic and semi-algebraic sets, *Hermann,* 1990

Die folgen Bücher behandeln Computeralgebra mit Universalprogrammiersprachen. Im ersten geht es um die Implementierung einiger grundlegender Algorithmen in C++. Die aufgeführten Algorithmen verdeutlichen in erster Linie das Prinzip, was zu teilweise dramatischen Effizienzverlusten führt.

TAN KIAT SHI/WILLI-HANS STEEB: SymbolicC++: An introduction to Computer Algebra Using Object-Oriented Programming, *Springer Singapore,* 1998, [2]2000

Das nächste Buch benutzt ADA, behandelt allerdings nur Polynome in einer Veränderlichen:

PATRICE NAUDIN, CLAUDE QUITTÉ: Algorithmique algébrique, *Masson,* 1992

Die Listenverarbeitungssprache LISP *war die Grundlage mehrerer früher Computeralgebrasysteme wie* REDUCE *und* Macsyma; *der dreistufige Aufbau von* REDUCE *aus einem Computeralgebrasystem, einem „algebraischen“* LISP *und dem tatsächlichen* LISP *sowie das Wechseln zwischen diesen Modi ist Gegenstand der beiden Bücher*

F. BRACKX, D. CONSTALES: Computer Algebra with LISP and REDUCE. An Introduction to Computer-aided Pure Mathematics, *Kluwer,* 1991

MALCOLM MACCALLUM, FRANCIS WRIGHT: Algebraic Computing with REDUCE, *Clarendon Press, Oxford,* 1991

Einen nicht mehr ganz aktuellen Überblick über verschiedene gebräuchliche Computeralgebrasysteme gibt

ULRICH SCHWARDMANN: Computeralgebra-Systeme, *Addison-Wesley,* 1995

Die „offiziellen“ Beschreibungen der gängigen Computeralgebrasysteme sind

M.B. MONAGAN, K.O. GEDDES, K.M. HEAL, G. LABAHN, S.M. VORKOETTER: Programmieren mit Maple V, *Springer,* 1996

STEPHEN WOLFRAM: The Mathematica Book, *Cambridge,* 1996

B. FUCHSSTEINER, K. GOTTHEIL, A. KEMPER, O. KLUGE, K. MORISSE, H. NAUNDORF, G. OEVEL, T. SCHULZE: MuPAD, *Birkhäuser,* 1993

Daneben gibt es, insbesondere für Maple und Mathematica, eine ganze Reihe von Büchern über Analysis, Lineare Algebra, Differentialgleichungen und ähnliche Themen, die eines dieser Computeralgebrasysteme benutzen (und teilweise auch auf CD in veralteter/eingeschränkter) Form mitliefern; diese sind hier nicht aufgeführt, da dort nicht die Computeralgebra im Vordergrund steht. Anhand des elektronischen UB-Katalogs lassen sich die hier vorhandenen Bände aber natürlich leicht finden durch die Angabe von Maple oder Mathematica als Titelwort.