den bereits vorhandenen Lehrbüchern im Bereich der Lösung polynomialer Gleichungssysteme. Es sollte in keinem Regal fehlen.

Peter Bürgisser (Paderborn)

W. Koepf
Computeralgebra. Eine algorithmisch orientierte Einführung

Springer Verlag, Berlin, Heidelberg, New York, 2006, ISBN 3-540-29894-0, €39,95

Es fällt schon auf, dass der Springer-Verlag in recht kurzem zeitlichen Abstand zu dem Buch von M. Kaplan (ISBN 3-540-21379-1) ein weiteres einführendes Lehrbuch mit gleichem Titel und gleicher Aufmachung auf den ja insgesamt nicht sehr reichhaltig bedienten Markt deutschsprachiger Literatur bringt. Da drängt sich ein Vergleich auf, zumal es bei der Themenauswahl naturgemäß große Überschneidungen, aber auch deutliche Unterschiede gibt. Auch wenn dies keine vergleichende Rezension ist, werde ich im Folgenden auch andeuten, was diese beiden Texte voneinander unterscheidet.

Der Untertitel des Koepfschen Buches, *Eine algorithmisch orientierte Einführung*, erstaunte mich: Wie könnte eine Einführung in die Computeralgebra anders als algorithmisch orientiert sein? Nun, das Vorwort ist dazu ein wenig deutlicher („Die Präsentation ist mehr algorithmisch als algebraisch orientiert."), und beim Durchblättern des Textes wird dann schnell klar, was damit gemeint ist: Im Jargon amerikanischer Verlage würde man das als einen *hands-on approach* bezeichnen. Wie das Buch von Kaplan ist der vorliegende Text aus Vorlesungen des Verfassers hervorgegangen, wobei er zwei vierstündige Vorlesungen wiedergibt, eine einführende mit den „unverzichtbaren Grundlagen" (das macht die ersten neun Kapitel, vom Umfang her etwa zwei Drittel des Buches aus) und eine weiterführende, die in drei Kapiteln drei spezielle Themen darstellt. In diesen Vorlesungen und den dazugehörigen Übungen wird intensiv mit *Mathematica* gearbeitet, und dieses praktische Tun und Demonstrieren wurde in den Text eingewoben. Ein beträchtlicher Teil des Textes besteht aus *Mathematica*-Kommandos, -Programmen und -Sitzungen, die zum unmittelbaren Nachvollziehen, zum Experimentieren und zum Weiterentwickeln einladen. So sieht man, anders als bei Kaplan und anderen Texten, keinen Pseudocode für die behandelten Algorithmen. Auf die Wahl von *Mathematica* werde ich weiter unten noch einmal eingehen. Hier möchte ich nur darauf hinweisen, dass auf der Web-Seite des Autors diese Programme als Worksheets auch in *Maple* und *MuPAD* verfügbar sind.

Die genannten Vorlesungen, auf denen dieses Buch beruht, wenden sich an Studierende unterschiedlicher Studiengänge (Diplom-Mathematik und Lehramt Mathematik, Bachelor Computational Mathematics und Informatik), also einen Hörerkreis mit unterschiedlichen mathematischen Voraussetzungen, gerade auch im Bereich der Algebra. Daher müssen die benötigten Begriffe, Techniken und Resultate in diesen Vorlesungen bereitgestellt werden, und das ist auch in diesem Buch wiedergegeben. Auch wenn nicht viel vorausgesetzt wird – der Autor selbst stuft da Texte wie den von M. Kaplan und das bekannte Buch von J. v. z. Gathen und J. Gerhard als deutlich anspruchsvoller ein – und alle Beweise geführt werden, sollte man sich nicht täuschen: Dies ist kein stringenter Algebra- oder Arithmetik-Kurs mit algorithmischen Zutaten! Hier steht die Algorithmik der Computeralgebra im Vordergrund und der begriffliche Apparat wird auf das Nötige und Elementare beschränkt. Um das an Beispielen zu verdeutlichen: Selbst einfachste gruppentheoretische Fakten (Struktur zyklischer Gruppen, Satz von Lagrange, multiplikative Struktur der Restklassenringe der ganzen Zahlen, Eulersche φ-Funktion) werden nicht explizit erörtert. Der Begriff des Homomorphismus wird nicht instrumentalisiert, das „Rechnen mit homomorphen Bildern" (Geddes/Czapor/Labahn, v. z. Gathen/Gerhard, Kaplan) also auch nicht als ein Paradigma der Computeralgebra formuliert. So kommt die Technik der modularen Arithmetik nur in einer eher bescheidenen Version zum Zuge, und die begriffliche Einordnung, der „höhere Standpunkt", kommt zu kurz, wenn er z. B. bei Themen wie der diskreten Fouriertransformation oder dem Hensel-Lifting durchaus hilfreich wäre. Man kann es auch so sehen: Wer sich ernsthaft mit Computeralgebra auseinandersetzen möchte, sollte durch das vorliegende Buch zu intensiver Beschäftigung mit der algorithmischen Algebra motiviert werden.

Was ist nun in dem Buch? Im ersten Kapitel werden anhand von *Mathematica*-Kommandos einige typische Eigenschaften von Computeralgebrasystemen demonstriert. Das zweite Kapitel geht sehr knapp und exemplarisch auf einige Aspekte der *Mathematica*-Programmierung ein. Vorsicht – die durchgängige Verwendung von *Mathematica* bedeutet nicht, dass dieser Text eine Einführung in das System *Mathematica* oder dessen Programmierung bietet! Hierfür muss man sich unbedingt noch mit einschlägiger Literatur versorgen und/oder die Hilfefunktionen ausgiebig konsultieren. *Mathematica* als Programmiersprache erlaubt viele

verschiedene Programmierstile (prozedural, funktional, regelbasiert), die, vermischt angewendet, zusammen mit den vielen eingebauten „hohen“ Konstrukten von *Mathematica* schnell zu kompakten, aber schwer lesbaren Programmen führen können. Nun sind hier die Programme kaum je länger als 20 Zeilen, da ist das noch kein Problem. Einem attraktiven Aspekt von *Mathematica*s Programmiersprache, dem *pattern matching*, wird hier und in der Folge einige Aufmerksamkeit geschenkt. Die Übersetzung mit „Mustererkennung“ halte ich in diesem Zusammenhang allerdings für unglücklich; besser wäre es, von „Musterabgleich“ zu sprechen. Das dritte Kapitel behandelt einige grundlegende Aspekte der Arithmetik ganzer Zahlen, bis hin zur eindeutigen Faktorzerlegung. Das Faktorisierungsproblem wird, im Gegensatz zum Buch von M. Kaplan, nur am Rande erwähnt. Das folgende Kapitel ist mit „Modulare Arithmetik“ überschrieben, behandelt aber nur – schon etwas halbherzig, s. o. – die Restklassenringe von $\mathbb{Z}$ mit Chinesischem Restsatz, kleinem Satz von Fermat (aber nicht dem Satz von Euler) und Primzahltests bis zum Miller-Rabin-Verfahren. Etwas irritierend ist die Bezeichnung $\mathbb{Z}_p$ für den Restklassenring modulo p auch dann, wenn p keine Primzahl ist. Diesen hat der Autor aber für die Zwecke dieses Buchs nicht wirklich im Visier. So kommt es, dass in den ergänzenden Bemerkungen zum vierten Kapitel das p im ersten Absatz eine beliebige natürliche Zahl ist, im zweiten aber eine Primzahl, ohne dass das gesagt wird. Ein kleiner Lapsus, sicher, aber auch eine Andeutung, dass die algebraische Seite vielleicht doch etwas ausführlicher dargestellt werden sollte. Das fünfte Kapitel gibt einen ganz knappen Einblick in Fragestellungen der (Quellen- und Kanal-) Codierung und der Public-Key-Kryptographie – mit einem Spielbeispiel der Reed-Solomon-Codes und dem RSA-Verfahren als Hinweise für Anwendungsbereiche der Computeralgebra. Das ist übrigens die einzige Stelle in diesem Buch, an der über den engeren Bereich der Computeralgebra hinaus geblickt wird. Der Polynomarithmetik wird dann in den folgenden drei Kapiteln breiter Raum gewährt. Den Grundlagen im sechsten Kapitel folgt ein Kapitel zu algebraischen Zahlen und zur Konstruktion und Struktur endlicher Körper. Das wird dann fortgeführt und angewendet zur Beschreibung der Polynomfaktorisierung über $\mathbb{Z}$ mittels Polynomfaktorisierung über $\mathbb{Z}_p$ und Hensel-Lifting. Der erste, einführende Teil des Buches wird abgeschlossen durch ein Kapitel über Vereinfachung und Normalformen, das sich i. W. auf Normalformen für Polynome und trigonometrische Polynome beschränkt. Hier steht man vor dem Dilemma, dass die tatsächlichen Vereinfachungsstrategien der Computeralgebrasysteme in der Regel nicht offen zugänglich sind, sich also Theorie und Praxis nicht so ohne Weiteres in Bezug bringen lassen.

Die letzten drei Kapitel des Buches geben den zweiten Teil des Vorlesungszyklus wieder, bei dem der Autor auch seinen eigenen Forschungsinteressen Raum gibt. Verdienstvoll ist ein Kapitel über formale Potenzreihen, ein in der (Lehrbuch-)Literatur eher vernachlässigtes Thema. Hier steht der sogenannte holonome Fall im Mittelpunkt, also Funktionen/Folgen, die durch lineare Differentialgleichungen/Differenzengleichungen mit rationalen Funktionen als Koeffizienten dargestellt werden können. Die Stichworte algebraische, hypergeometrische, implizite Funktionen sollen andeuten, dass sich hier ein reichhaltiges und aktuelles Anwendungsgebiet für Methoden der Computeralgebra in der Analysis auftut. Ähnliches gilt auch für die algorithmische Summation und Integration, denen die beiden letzten Kapitel gewidmet sind. Hier ist die Computeralgebra nicht nur Mittel zum Zweck, sondern hat auch durch ihre Möglichkeiten und Methoden der mathematischen Duchdringung wesentliche Impulse verliehen. Bei der Summation wird der hypergeometrische Fall, mit den Namen von Sister Celine, B. Gosper und D. Zeilberger verbunden, einigermaßen ausführlich dargestellt. Dies geht wesentlich weiter als der entsprechende Abschnitt im Buch von Kaplan. Der interessierte Leser sollte damit in der Lage sein, in aktuelle Publikationen einzusteigen. Bei der Integration wird der klassische Fall der Integration rationaler Funktionen bis hin zu den Methoden von Rothstein/Trager und Rioboo behandelt, also etwa wie bei Geddes/Czapor/Labahn (dieses Thema fehlt bei Kaplan völlig). Klarerweise stellt dieser zweite Teil an das mathematische Verständnis höhere Anforderungen als der erste, nicht nur von der rein technischen Seite, sondern der Leser sollte auch einen gewissen analytischen Erfahrungshintergrund (Funktionentheorie, spezielle Funktionen, konkrete Summations- und Integrationsprobleme) mitbringen, um die dargestellte Entwicklung schätzen zu können.

Was in diesem Buch nicht zu finden ist? Auf die Behandlung von Themen wie Gröbnerbasen, Gitterreduktion, elliptische Kurven, ganzzahlige Faktorisierung, Differentialgleichungen, ... hat der Autor wohlweislich verzichtet; auch so ist der Umfang zweier Semester gut ausgefüllt. Die mathematischen Anforderungen müsste man von Anfang an höher ansiedeln, und es gibt ausgezeichnete spezielle Literatur dazu.

Noch ein paar Bemerkungen zu speziellen Aspekten des Buchs: Die Darstellung mathematischer Sachverhalte ist, bis auf einige wenige „Ausrutscher“, korrekt und sollte für das Zielpublikum durchweg gut verständlich sein. Mathematisch Interessierte werden hier und da Zusammenhänge vermissen – so stehen in den Übungen zum dritten Kapitel euklidischer Algorithmus und Kettenbruchentwicklung nebeneinander, ohne dass die Verbindung gezogen wird. Oder im zweiten Kapitel: Da fällt eine „schnelle“ Rekursion für die Fibonacci-Zahlen quasi vom Himmel; dass sich dahinter das allgemeine Prinzip der schnellen Exponentiation (iteriertes Quadrieren und Multiplizieren, alias russische Bauernexponentiation) verbirgt, anwendbar auf alle Folgen, die einer linearen Rekursion mit konstanten Koeffizienten genügen, bleibt verborgen. An solchen Stellen ist der Dozent gefragt.

Algorithmen werden, wie schon betont, nur als *Mathematica*-Code dargestellt. In den meisten Fällen werden auch Aussagen zur Komplexität (Laufzeit) gemacht, wobei nur uniforme (aber nicht auch logarithmi-

sche) Komplexität betrachtet wird, also die Anzahl der Elementar-Operationen, unabhängig von der Größe der Operanden. Das kann beim Rechnen über $\mathbb{Z}$ oder Polynomringen irreführende Aussagen ergeben. Gerade der beliebte Algorithmus für die „schnelle“ Exponentiation liefert hierfür ein schlagendes Beispiel. Über Aspekte des Aufbaus von Computeralgebrasystemen (Datenstrukturen, Speicherung, Auswertungs- und Simplifikationsstrategien) erfährt man nur durch gelegentliche Andeutungen etwas, aber das wäre auch eher Gegenstand anderer Lehrveranstaltungen.

Jedes Kapitel enthält einen Abschnitt „Ergänzende Bemerkungen“. Gerade im Hinblick auf die Tatsache, dass eine Vorlesung naturgemäß nur ein gerafftes Bild geben kann, empfinde ich diese Ergänzungen als zu „mager“ geraten. Hier könnte man mit ein wenig mehr Platz mehr attraktive Hinweise für interessierte Leser geben. Zwei Ergänzungen meinerseits zu den Kapiteln 10 und 11: Zum holonomen und hypergeometrischen Themenkreis gibt es außer der erwähnten eigenen des Autors eine ganze Reihe weiterer Implementierungen, über die man sich auf der Seite *Combinatorial Software and Databases* unter `http://igd.univ-lyon1.fr/~slc/` orientieren kann. Insbesondere die Arbeiten von F. Chyzak (INRIA) mit `Mgfun` wären hier einen Hinweis wert. Was das Summations-Analogon des Risch-Algorithmus betrifft, also den Algorithmus von Karr, der ganz am Ende von Kapitel 11 kurz angesprochen wird, so hat sich hier in den letzten Jahren durch die Arbeiten von C. Schneider (RISC Linz) enorm viel getan – gerade auch im Hinblick auf die Implementierung.

Insgesamt bezieht das vorgelegte Buch seinen Reiz aus dem unmittelbaren Arbeiten mit einem Computeralgebrasystem. Wenn dieser praktische Zugang dazu führt, dass die Studierenden animiert und neugierig werden auf ein tieferes Verständnis, hat er seine Berechtigung, auch wenn einige problematische Seiten gesehen werden sollten. Die ausschließliche Verwendung von *Mathematica* zur Formulierung von Algorithmen kann den Blick für eine abstraktere Sicht von Algorithmen erschweren. Abstraktion kann (oder sollte wenigstens) ein Mittel der effizienteren Durchdringung komplexer Sachverhalte sein. Das gilt auch für die algebraische Seite der Medaille. Was nun die Verwendung von *Mathematica* im Speziellen angeht, so begründet der Autor seine Entscheidung dafür mit der besseren Benutzer-Oberfläche im Vergleich zur Konkurrenz. Das mag mit den neuen Versionen von *Maple* und *MuPAD* kein ganz so kräftiges Argument mehr sein, aber „Oberfläche“ hat in diesem Zusammenhang noch eine andere Bedeutung: Es ist bei *Mathematica* in der Regel nicht sichtbar oder erfahrbar, was unter der Oberfläche eigentlich abläuft – ganz im Gegensatz zu *Maple*. Der Autor ist sich dieses Nachteils durchaus bewusst, vieles bleibt eben „Betriebsgeheimnis“. Ich selbst ziehe in dieser Situation die Offenheit von *Maple* der Oberfläche von *Mathematica* vor, aber das ist eine Entscheidung, die auch noch von anderen Parametern (Verfügbarkeit, Lizenzen) beeinflusst wird.

Volker Strehl (Erlangen)

H. Li, P. J. Olver, G. Sommer (Hrsg.)
Computer Algebra and Geometric Algebra with Applications

Springer Verlag, Berlin, Heidelberg, New York, 2005, ISBN 3-540-26296-2, €64,20

Dieser Sammelband enthält die Proceedings des 6. Internationalen Workshops über „Mathematics Mechanization“ (IWMM). Diese Workshopreihe wurde 1992 von Wen-tsun Wu initiiert und dient der Propagierung des Ansatzes der *Mechanisierung der Mathematik*, mit dem Wu seit Jahren den integrativen Einsatz algorithmischer und rechentechnischer Hilfsmittel an allen Fronten der Mathematik gegen eine Geringschätzung der algorithmischen gegenüber deduktiven Ansätzen betont. Ähnliche Überlegungen finden sich bei verschiedenen Autoren, exemplarisch etwa im Aufsatz *Computeralgebra – eine Säule des Wissenschaftlichen Rechnens* von Johannes Grabmeier (*it+ti* **6** (1995), 5 – 20), der von einer weitergehenden Symbiose von Kalkül und Technologie als Gegenstand eines Faches zwischen Mathematik und Informatik spricht und diesem den provisorischen Namen *Computermathematik* gibt. Wu hat eine wohl noch stärkere Hinwendung zu Anwendungen der Mathematik im Sinne des Selbstverständnisses des wissenschaftlichen Rechnens im Auge, wenn er fordert: „make algorithmic studies and applications of mathematics the major trend of mathematics development in the information age“.

Ein solches Mathematikverständnis ist der Ausgangspunkt für den Zuschnitt aller Workshops über „Mathematics Mechanization“. Es geht über das Selbstverständnis der Computeralgebra als *Teil*-Disziplin (u. A.) der Mathematik hinaus, das für die großen Computeralgebra-Tagungen wie ISSAC konstitutiv ist, und hat die Mathematik als Ganzes im Blick, in jedem einzelnen Workshop natürlich thematisch begrenzt. In der Einleitung zum vorliegenden Sammelband wird das für 2004 wie folgt beschrieben: „At each workshop several experts are invited to deliver plenary lectures on