*Koepf*
Computeralgebra

Wolfram Koepf

# Computeralgebra

Eine algorithmisch orientierte Einführung

Springer

*Prof. Dr. Wolfram Koepf*
Fachbereich 17 Mathematik/Informatik
Universität Kassel
Heinrich-Plett-Straße 40
34132 Kassel, Deutschland
e-mail: koepf@mathematik.uni-kassel.de
Internet: http://www.mathematik.uni-kassel.de/~koepf

Bibliografische Information der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.ddb.de abrufbar.

Mathematics Subject Classification (2000): 68W30, 11Y05, 11Y11, 11Y16, 11R04, 12D05, 12Y05, 13F20, 13F25, 33F10

ISBN-10 3-540-29894-0 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-29894-6 Springer Berlin Heidelberg New York



Springer ist ein Unternehmen von Springer Science+Business Media

springer.de



Umschlaggestaltung: *design & production* GmbH, Heidelberg
Herstellung: LE-TEX Jelonek, Schmidt & Vöckler GbR, Leipzig
Satz: Datenerstellung durch den Autor unter Verwendung eines Springer TEX-Makropakets
Gedruckt auf säurefreiem Papier 44/3100YL - 5 4 3 2 1 0

# Vorwort

Das vorliegende Lehrbuch über Computeralgebra gibt eine Einführung in dieses moderne Gebiet der Mathematik. Es entstand im Laufe der letzten fünf Jahre und umfaßt den Stoff zweier vierstündiger Vorlesungen, die ich mehrfach an der Universität Kassel durchgeführt habe. Als ich im Jahr 2000 an die Universität Kassel kam und im Sommersemester 2000 zum ersten Mal die Vorlesung Computeralgebra und im darauffolgenden Wintersemester die Vorlesung Computeralgebra 2 abhalten durfte, begann ich mit der Niederschrift. Ich habe diese Vorlesungen dann mehrfach an der Universität Kassel für Studenten der Studiengänge Diplom-Mathematik, gymnasiales Lehramt Mathematik, Bachelor Computational Mathematics sowie Bachelor Informatik ab dem dritten Studiensemester abgehalten.

Vorausgesetzt werden also lediglich einige Kenntnisse aus der linearen Algebra sowie der elementaren Analysis. Daher ist das Buch auch für interessierte Gymnasiallehrer als Quelle zum Verständnis für Algorithmen der Computeralgebra oder als Vertiefung in Leistungskursen sehr gut geeignet. Ich habe mich bemüht, die wichtigsten Prinzipien und Algorithmen der Computeralgebra aufzunehmen, deren Beweise ich allerdings so elementar wie möglich führe. Da ich den Besuch einer Algebra-Vorlesung nicht voraussetzen konnte, habe ich auf tiefliegende algebraische Argumente verzichtet. Die Präsentation ist mehr algorithmisch als algebraisch orientiert. Beispielsweise wird der Chinesische Restsatz (lediglich) als Algorithmus und nicht als Isomorphiesatz formuliert. Die vorliegende Vorlesung ersetzt keine Algebra-Vorlesung, sie wirbt vielmehr für eine algebraische Vertiefung.

In der einführenden Computeralgebra-Vorlesung habe ich in etwa den Stoff der ersten neun Kapitel durchgenommen. Dieser Kanon scheint mir die unverzichtbare Grundlage der Computeralgebra darzustellen. Jeder Dozent wird allerdings sicher eigene Akzente setzen. Ich habe beispielsweise aus Zeitmangel die Beweise einiger Sätze über endliche Körper weggelassen und auf den Beweis im Text verwiesen, ein anderer Dozent wird vielleicht lieber auf das Kapitel über Codierungstheorie und Kryptographie verzichten, obwohl nach meiner Erfahrung die Studierenden gerade dieses Kapitel sehr gerne annehmen. Eine weitere Variante besteht darin, die Algorithmen für ganze Zahlen und für Polynome (wie beispielsweise den euklidischen Algorithmus und seine Erweiterung) parallel anstatt hintereinander zu behandeln. Kann man höhere Algebrakenntnisse voraussetzen, werden einige Beweise kürzer. Dem jeweiligen Dozenten bleiben also genügend Auswahlmöglichkeiten, um seinen eigenen Stil zu bewahren.

In den ersten beiden Kapiteln wird vorgestellt, was ein Computeralgebrasystem kann und wie man in einem solchen System programmiert. Danach wird die Ganzzahlarithmetik behandelt, die die Grundlage jedes Computeralgebrasystems darstellt. Nach dem modularen Rechnen mit dem chinesischen Restsatz, dem kleinen Satz von Fermat und

der Betrachtung von Primzahltests folgt dann zunächst ein Kapitel über Codierungstheorie und Kryptographie, in welchem die behandelten zahlentheoretischen Grundlagen angewandt werden. Hier wird u.a. das RSA-Verfahren behandelt.

Als nächstes werden Polynome betrachtet, wobei hier die Algorithmen der Ganzzahlarithmetik erneut ins Spiel kommen. Es folgt der Kronecker-Algorithmus zur Faktorisierung von Polynomen mit ganzzahligen Koeffizienten. Im Kapitel über algebraische Zahlen wir dann die modulare Arithmetik von einem neuen Standpunkt aus betrachtet. Nun können auch endliche Körper und Resultanten eingeführt werden. Im folgenden Kapitel über Polynomfaktorisierung werden moderne effizientere Algorithmen behandelt und implementiert. Es folgt ein Kapitel über Vereinfachung und Normalformen, welches den Kanon der ersten Vorlesung abschließt.

Die Kapitel 10-12 stellen die Themen der von mir durchgeführten weiterführenden Vorlesung bereit und sind naturgemäß stärker von meinen eigenen Interessen geprägt. Diese Auswahl ist nach meiner Überzeugung für eine weiterführende Vorlesung besonders gut geeignet, denn die Kapitel wenden das in der ersten Vorlesung erarbeitete Wissen auf Themen an, die jedem Mathematikstudenten während seines Studiums begegnen und die auch viele Mathematik-Anwender benötigen und interessieren: Potenzreihen, Summationsformeln und Integration. Während man Potenzreihen bereits in der Analysis kennenlernt und beispielsweise in der Physik und in der Wahrscheinlichkeitstheorie benötigt, treten Summationsformeln überall in Mathematik und in Anwendungen auf. Unbestimmte Integration schließlich wird im allgemeinen als ein schwieriges Problem der Analysis betrachtet, und es ist nicht unmittelbar klar, daß man dieses Problem mit Methoden der Algebra anpacken kann.

Naturgemäß konnte ich nicht alle relevanten Themen der Computeralgebra aufnehmen, auch weil das Buch eine direkte Vorlage zu zwei Vorlesungen liefern und daher nicht zu umfangreich sein sollte. Beispielsweise habe ich mich entschieden, die Theorie der Gröbnerbasen wegzulassen, obwohl diese eine sehr bedeutende Rolle in Computeralgebrasystemen spielen, insbesondere bei der Lösung polynomialer Gleichungssysteme, wo Gröbnerbasen inzwischen Resultanten den Rang abgelaufen haben. Will man dieses Thema allerdings gründlich behandeln, so muß man hierfür entweder höhere Algebrakenntnisse voraussetzen oder es wird daraus leicht ein eigenes Buch. Hier verweise ich also lieber auf die bereits vorhandene Literatur, beispielsweise das für eine Vorlesung oder ein Seminar sehr gut geeignete Buch von Cox, Little und O'Shea [CLO1997]. Ebenfalls hätte ich gerne den für viele Anwendungen wichtigen LLL-Algorithmus von Lenstra, Lenstra und Lovász [LLL1982] behandelt, mußte diesen aus Platzgründen aber ebenfalls weglassen. Bei einer weiteren Vertiefung ist eine Behandlung dieser beiden Themenkreise allerdings zwingend.

Alle betrachteten Algorithmen des Buchs werden in Sitzungen mit dem Computeralgebrasystem *Mathematica* programmiert und getestet. Dies geht auf eine persönliche

Präferenz des Autors zurück.[1] Dennoch ist der jeweilige Dozent keineswegs an dieses System gebunden, denn die Definitionen und Sätze des Buchs sind selbstverständlich völlig unabhängig von einem speziellen Computeralgebrasystem.

Um das Buch vollends systemunabhängig zu gestalten, habe ich im Internet auf der Webseite `http://www.mathematik.uni-kassel.de/~koepf/CA` alle Computeralgebra-Sitzungen des Buchs auch als *Maple*-Worksheets und *MuPAD*-Notebooks bereitgestellt. Die drei verwendeten Systeme sind allesamt sogenannte General Purpose-Systeme und beherrschen die Mathematik in einer Breite, die dem Themenumfang des Buchs entspricht. Sie besitzen ferner die Möglichkeit, Sitzungen in einem internen Format (*Mathematica*: Notebooks `.nb`; *Maple*: Worksheets `.mws` und `.mw`; *MuPAD*: Notebooks `.mnb` und `.mn`) abzuspeichern.

Die in einem derartigen Notebook oder Worksheet abgespeicherten Befehle können jederzeit erneut vom System berechnet werden, was eine Vorführung durch den Dozenten ermöglicht. Ich habe die Computeralgebra-Sitzungen in meinen Veranstaltungen mit Laptop und Beamer vorgeführt und die Rechenergebnisse jeweils während der Vorlesung erzeugt. Dies gibt den Studenten die Möglichkeit, die Algorithmen direkt nachzuvollziehen, und es können auch immer sofort weitere Beispiele betrachtet werden, die von den Studierenden vorgeschlagen werden. Ferner können die Algorithmen im Sinne experimenteller Mathematik ohne Umweg zum Testen benutzt werden.

Aus diesem Grund war mir auch wichtig, die Algorithmen nicht – wie in vielen anderen Büchern zum Thema – in Pseudocode darzustellen, sondern als lauffähige Programme. Im Rahmen dieses Buchs bzw. der im Internet verfügbaren *Maple*- und *MuPAD*-Quellen liegen alle betrachteten Algorithmen in den drei Systemen *Mathematica*, *Maple* und *MuPAD* vor. Ich denke, daß dies für viele Leser nützlich sein dürfte. Was den Übungsbetrieb betrifft: Mit allen drei Systemen können die Studenten sehr einfach ihre bearbeiteten Übungsblätter als Notebooks oder Worksheets digital zum Korrigieren einreichen, und der Übungsleiter kann diese Dateien dann korrigiert zurückgeben.

Als Sprecher der Fachgruppe Computeralgebra habe ich einige Tagungen zum Thema *Computeralgebra in Lehre, Ausbildung und Weiterbildung* organisiert. Auf diesen Tagungen bin ich des öfteren von Gymnasiallehrern nach Literatur zu den Algorithmen der Computeralgebra gefragt worden. Mit diesem Buch liegt auch für diesen Leserkreis eine leicht verständliche Beschreibung der algorithmischen Grundlagen von Computeralgebrasystemen vor. Bislang gibt es – bis auf das kürzlich ebenfalls beim

[1] Ich halte *Mathematica* für die Lehre am besten geeignet, denn die Oberfläche ist ausgereifter als die der Konkurrenz, sie ist schneller, leichter zu bedienen und hat mehr Eingabemöglichkeiten. Für die Forschung allerdings ist *Mathematica* nur sehr begrenzt geeignet, da sich Ergebnisse in der Regel nicht verifizieren lassen, denn *Mathematica* verfügt über keine Möglichkeit, den Rechenablauf und die verwendeten Algorithmen zu hinterfragen. Hier haben *Maple* und *MuPAD* deutliche Vorteile.

Springer-Verlag erschienene Buch von Michael Kaplan [Kap2004], welches allerdings höhere Algebrakenntnisse voraussetzt – keine derartige Quelle in deutscher Sprache. Da das Buch elementar gehalten ist, ist es auch als Nachschlagewerk über Algorithmen der Computeralgebra gut geeignet, denn es besitzt einen sehr ausführlichen Index.

Als Grundlage für die ersten 9 Kapitel des vorliegenden Buchs konnte ich auf die englischsprachigen Bücher [Chi2000], [GCL1992] und [GG1999] zurückgreifen. Das Buch von Lindsay Childs entstand bereits 1979 und war sicher eines der ersten Bücher über Algorithmen der Computeralgebra, obwohl dies der Titel *A Concrete Introduction to Higher Algebra* nicht unbedingt suggeriert. Ebenfalls eine sehr wertvolle Quelle ist das Buch von Geddes, Czapor und Labahn aus dem Jahr 1992, das im wesentlichen eine sehr ausführliche und gut lesbare Beschreibung der Fähigkeiten von *Maple* darstellt. Schließlich erschien 1999 das Buch der beiden Autoren von zur Gathen und Gerhard, das man durchaus als Computeralgebra-Bibel bezeichnen kann. Aufgrund seiner Fülle und seiner höheren Algebra-Voraussetzungen scheint dieses Buch als Grundlage für eine einführende Vorlesung allerdings nicht direkt geeignet zu sein.

Das vorliegende Buch wäre nicht entstanden ohne die Hilfe zahlreicher Kollegen, Mitarbeiter und Studenten, die mir mit Rat und Tat zur Seite standen, den Text an etlichen Stellen abrunden und viele große und kleine Fehler der Erstfassung korrigieren halfen. Ganz besonders bedanken möche ich mich bei meinem langjährigen Forschungskollegen Dr. Dieter Schmersau, mit dem ich unzählige Gespräche über das gesamte Material geführt habe sowie bei Dr. Reinhard Oldenburg, der ebenfalls sehr ausführlich Korrektur gelesen hat. Schließlich geht mein Dank an meine Mitarbeiter, Studenten und Freunde Imran Hafeez, Peter Horn, Detlef Müller, Torsten Sprenger und Jonas Wolf. Jeder der Genannten hat mir wertvolle Hinweise zum Text gegeben und jedem Einzelnen ist zu verdanken, die Anzahl schlecht verständlicher Textteile, Druckfehler etc. deutlich vermindern zu helfen. Eine besonders unangenehme Fehlerquelle war die mehrfache Anpassung des Textes auf neue *Mathematica*-Versionen im Laufe der Jahre. Ich hoffe und bin einigermaßen zuversichtlich, daß das nun veröffentlichte Buch keine sinnentstellenden Fehler mehr enthält. Ich kann allerdings selbstverständlich nicht dafür garantieren, ob sich neue Versionen von *Mathematica* – bzw. in den ausgearbeiteten Sitzungen *Maple* und *MuPAD* – wie beschrieben verhalten werden.

Ein herzlicher Dank geht auch an die Universität Kassel für die Genehmigung zweier Forschungssemester im Zeitraum des Entstehens dieses Buchs, ohne die eine Fertigstellung nicht möglich gewesen wäre. Ebenfalls bedanke ich mich beim Konrad-Zuse-Zentrum Berlin für die Möglichkeit, dort in den letzten 5 Jahren an meinem Buch arbeiten zu können. Schließlich möchte ich mich beim Springer-Verlag bedanken, der mein Projekt von Anfang an begleitet und unterstützt hat.

Kassel, 20. November 2005 Wolfram Koepf

# Inhaltsverzeichnis