Wolfram Koepf, Kassel

Computeralgebra in der universitären Lehre

Dieser Vortrag besteht aus zwei Teilen. Im ersten Teil erzähle ich, in welcher Form sich das Thema Computeralgebra im Rahmen der neuen Bachelor- und Master-Studiengänge positionieren kann. Im zweiten berichte ich über meine Vorlesungs-Erfahrungen zur Einführung der Computeralgebra. Natürlich haben derartige Vorlesungen ihre Bedeutung für den Bachelor-Studiengang.

Was verändert die Einführung von Bachelor und Master?

Die Einführung des Bachelors verkürzt die Ausbildungszeit bis zum ersten Abschluss. Der erste Abschluss (bei uns in Kassel eine dreimonatige Bachelorarbeit) muss bereits nach 5 Semestern anvisiert werden. Es ist *politisch gewollt*, dass der Bachelor *kein* wissenschaftlicher Abschluss ist. Die Studiengänge sind *modularisiert*. Die Module haben *studienbegleitende Prüfungen*. Abschlussprüfungen entfallen.
Die Lehramtsstudiengänge in Hessen sind bereits modularisiert. Der Verteilungsschlüssel Fach – Didaktik – Erziehungswissenschaften wurde vom hessischen Ministerium vorgegeben. Der verbliebene Anteil für die Fachausbildung ist insbesondere beim gymnasialen Lehramt ziemlich klein. Als Resultat gibt es bei uns nach den Anfängervorlesungen im ersten Studienjahr nur noch *kleine Häppchen*, d. h. 2V+1Ü-Vorlesungen, damit dennoch eine gewisse Breite erreicht werden kann. Natürlich geht dies auf Kosten der Tiefe.
Da wir (aus Kapazitätsgründen) für den gymnasialen Lehramtszweig keine eigenen Veranstaltungen anbieten können, sind auch die Veranstaltungen im Bachelor klein. Eine 4-stündige Analysis III oder eine Algebra I bzw. II gibt es also nicht mehr. Dafür gibt es nun jeweils einführende 2+1-Veranstaltungen wie z. B. *Einführung in die Funktionentheorie*. Schon eine zweite 2+1-Vorlesung ist damit eine Vertiefungsveranstaltung.

Wo ist die Computeralgebra verortet?

In Kassel haben wir auch die Anfängervorlesungen der Algebra umstrukturiert. Die Vorlesungen der ersten beiden Semester heißen nun *Algorithmische*

Lineare Algebra. Wir hoffen hierbei, die Studenten besser bei ihrem Schulwissen „abzuholen“, sowie durch den algorithmischen Aspekt auf ein besseres Verständnis. Die dann folgende Vorlesung *Grundlagen der Algebra und Computeralgebra* behandelt die Elemente der algorithmischen Zahlentheorie, welche Basis jedes Computeralgebrasystems sind.
Seit 2001 hatten wir bereits einen speziellen Bachelor-/Master-Studiengang *Computational Mathematics* mit Vertiefung in Computeralgebra und Kryptographie. Leider konnten wir hierfür nur wenige Studierende gewinnen. Als Resultat wird dieser Studiengang zugunsten eines gemeinsamen Studiengangs „Bachelor Mathematik“ aufgegeben. Ein Abschluss in Computational Mathematics scheint weniger attraktiv zu sein als ein Abschluss in Mathematik, vielleicht, weil letzteres breiter erscheint. Es hat sich aber gezeigt, dass die wenigen Studenten, die bei uns den Bachelor CM abschlossen, ihre Bachelorarbeit alle im Bereich Computeralgebra geschrieben haben.

Erfahrungen aus meinen Vorlesungen

Meine Vorlesungen sind als Buch erschienen [1]. Von der Homepage des Buchs `http://www.mathematik.uni-kassel.de/~koepf/CA` können die vorgestellten Programme heruntergeladen werden.

Ganzzahl-Arithmetik

Schon bei der internen Vereinfachung rationaler Zahlen ist die Berechnung größter gemeinsamer Teiler erforderlich. In der Schule werden größte gemeinsame Teiler in der Regel durch vollständige Faktorisierung gefunden. Aber: die Faktorisierung ganzer Zahlen ist zeitaufwendig! Der aktuelle „Weltrekord“ ist die Faktorisierung einer 200-stelligen Dezimalzahl mit zwei 100-stelligen Primfaktoren, wofür dann hunderte Rechner viele Monate lang mit den besten verfügbaren Algorithmen beschäftigt waren [2]. Die Berechnung des größten gemeinsamen Teilers mit dem euklidischen Algorithmus ist viel effizienter. Hiermit lassen sich größte gemeinsame Teiler tausendstelliger Zahlen in Sekundenbruchteilen bestimmen.
Den euklidischen Algorithmus formuliert man (*rekursiv*) so:

- $\mathrm{ggT}(a,b) = \mathrm{ggT}(|a|,|b|)$, falls $a < 0$ oder $b < 0$
- $\mathrm{ggT}(a,b) = \mathrm{ggT}(b,a)$, falls $a < b$
- $\mathrm{ggT}(a,0) = a$
- $\mathrm{ggT}(a,b) = \mathrm{ggT}(b, a \bmod b)$

und ein entsprechendes Programm in einem General-Purpose-Computeralgebrasystem wie Derive, Maple, *Mathematica*, Maxima, MuPAD oder Reduce sieht praktisch genauso aus.
Lässt man den Euklidischen Algorithmus *iterativ* durchlaufen, so liefert er zusätzliche Informationen. Für $a, b \in \mathbb{Z}$ liefert der *erweiterte euklidische Algorithmus* $g = \mathrm{ggT}(a,b)$ und Koeffizienten $s, t \in \mathbb{Z}$ derart, dass

$$g = s\,a + t\,b\ .$$

Umkehrung: Gilt für geeignete $s, t \in \mathbb{Z}$ die Beziehung $1 = s\,a + t\,b$, so sind a und b teilerfremd.

Ein Primzahltest

Für eine Primzahl $p \in \mathbb{P}$ und $a \in \mathbb{Z}$ gilt *der kleine Satz von Fermat*

$$a^p \equiv a \pmod{p}\ ,$$

welcher ganz elementar bewiesen werden kann. Ist diese Beziehung für eine Zahl $a \in \mathbb{Z}$ nicht erfüllt, so kann also p keine Primzahl sein! Dies ist der sogenannte *Fermat-Test*.
Um den Fermattest durchführen zu können, müssen also sehr effizient modulare Potenzen berechnet werden. Die modulare Potenz $a^n \pmod{p}$ berechnet man am besten durch Zurückführen auf Exponenten der Größe $n/2$. Ein derartiges Verfahren wird *Divide- und Conquer- Algorithmus* genannt.
Eine rekursive Formulierung dieses Algorithmus ist gegeben durch

- $a^n \bmod p = (a^{n/2} \bmod p)^2 \bmod p$ für gerade n
- $a^n \bmod p = (a^{n-1} \bmod p) \cdot a \bmod p$ für ungerade n
- $a^0 \bmod p = 1$

Dies liefert wieder eine sehr effiziente Implementierung in einem Computeralgebrasystem.

Carmichaelzahlen

Eine ganze Zahl p, die *keine* Primzahl ist und dennoch das Fermatkriterium erfüllt, heißt *Carmichaelzahl*. Es gibt unendlich viele Carmichaelzahlen. Carmichaelzahlen werden vom Fermattest nicht erkannt. Die Bestimmung der ersten Carmichaelzahlen 561, 1105, 1729, 2465, 2821, ... gemäß der Definition ist bereits relativ aufwendig.
Schneller geht es, wenn man die folgende Strukturaussage zu Hilfe nimmt: Eine Zahl $p \in \mathbb{N} \setminus \mathbb{P}$ ist genau dann eine Carmichaelzahl, wenn

- $p = p_1 \cdots p_n$ mit lauter paarweise verschiedenen Primzahlen $p_k \in \mathbb{P}$
- $p_k - 1 | p - 1$ für alle $k = 1, \ldots, n$.

Hiermit lassen sich auch hundertstellige Carmichaelzahlen finden.

Ich möchte erwähnen, dass das berühmte RSA-Verschlüsselungsverfahren sich mit den nun behandelten Algorithmen vollständig implementieren lässt [1].

Zusammenfassung

Ich hoffe, mein Vortrag hat Ihnen Folgendes gezeigt:

- Bachelor- und Masterprogramme bieten die Möglichkeit, Computeralgebramodule zu integrieren.
- Computeralgebra bietet Themen für Bachelorarbeiten an.
- Programmieren mit General-Purpose-Computeralgebrasystemen bietet Funktionalitäten auf Hochsprachenniveau.
- Algorithmen der Computeralgebra sind – vor allem auch für Lehramtsstudenten – bedeutsam.

Literatur

[1] Koepf, W.: Computeralgebra. Springer, Berlin/Heidelberg, 2006

[2] Weisstein, E.: RSA-200 Factored. MathWorld Headline News:
`http://mathworld.wolfram.com/news/2005-05-10/rsa-200`