Symbolisches Rechnen*

Das Computeralgebrasystem Singular[1]

Vorausgesetzt wird, dass das genannte System (das in der Version 2.04 zur Verfügung stand) auf unserem Rechner installiert ist. Eine freie Kopie mit Installationsanleitung und Dokumentation ist unter der zuvor angegebenen www-Adresse erhältlich. 2/6/2
Der Aufruf von Singular unter Verwendung eines Dateinamens als Parameter führt dazu, dass die in der Datei aufgeführten Befehle der Reihe nach abgearbeitet werden. Bei Eingabe von Befehlen über die interaktive Shell werden diese ausgeführt und ihre Resultate auf dem Bildschirm ausgegeben.

```
int n=7;
```

definiert eine ganze Zahl `n`, der der Wert 7 zugewiesen ist. Mit dem Befehl

```
n;
```

wird der Wert von `n` (nach Betätigen der Eingabetaste) angezeigt; alternativ dazu können wir `n` in eine Datei schreiben, z.B. durch

```
write(“Resultat“,n);
```

in die Datei mit dem Namen „Resultat“.

```
help int;
```

ruft die Online-Hilfe zum Befehl „int“ auf. Dort sind auch Beispiele und präzise Angaben zur Syntax enthalten, so dass wir uns hier kurz fassen können.
Für die meisten Rechnungen ist ein Grundkörper erforderlich, der stets zusammen mit einem Polynomring über diesem eingeführt wird. Wir rechnen vorzugsweise über dem Primkörper (der bei Bedarf erweitert wird – beispielsweise dann, wenn nicht alle Koeffizienten der gegebenen Polynome darin enthalten sind).

```
ring r=5,X,lp;
```

definiert den Polynomring $\mathbb{F}_5[X]$ mit der lexikographischen Ordnung `lp`; die Primzahl 5 legt die Charakteristik des Grundkörpers fest. Ein Polynom lässt sich nun z.B. durch

```
poly f=X99-1;
```

definieren; dies ist $\mathtt{X}^{99}-1$, wobei für indizierte Variable stets der Exponent durch das Zeichen „ ˆ “ abzutrennen ist, im Zweifelsfall wird also

```
poly f=Xˆ99-1;
```

eingegeben. Offensichtlich gilt `f(1)=0`, wir sehen das auch unter Verwendung des Substitutionsbefehls

```
subst(f,X,1);
```

Einfache Operationen mit Polynomen werden auf nahe liegende Weise ausgeführt, z.B. durch

```
poly g=2*X+3;
f+g; f*g; 2*f-g; gˆ12;
```

Wir fragen nach der Faktorzerlegung für das zuvor eingegebene Polynom f. Als Teiler kommen nur Polynome vom Grad ≤ 99 infrage. Da der Grundkörper endlich ist, ergibt sich durch Unterscheidung einer endlichen Zahl von Fällen die Zerlegung in irreduzible Faktoren. Die Idee aus Beispiel

[1] Greuel, G.-M., Pfister, G., Schönemann, H., Singular *Reference Manual*. Reports On Computer Algebra 12, May 1997, Center for Computer Algebra, Universität Kaiserslautern

3.* in 2/4/10 ergibt auch über dem Körper der rationalen Zahlen eine prinzipielle Möglichkeit der Faktorzerlegung. Nun soll nicht behauptet werden, dass moderne CAS auf diese Weise rechnen - das Auffinden schneller Algorithmen zur Faktorisierung erfordert bereits richtig interessante Mathematik. Singular liefert das Resultat mit dem Befehl

```
factorize(f);
```

Dabei wird zuerst eine Liste der irreduziblen Faktoren und weiter die Liste ihrer Vielfachheiten ausgegeben (wobei der konstante Faktor am Anfang in unserem Sinn zu ignorieren ist).

Der $\mathbb{Q}$-Algebrahomomorphismus $\mathtt{h:S}\to\mathtt{R}$, $\mathtt{S}:=\mathbb{Q}\mathtt{[Y,Z]}$, $\mathtt{R}=\mathbb{Q}\mathtt{[X]}$ mit `h(Y)= X-1` und $\mathtt{h(Z)=X}^2$ lässt sich so einführen:

```
ring S=0,(Y,Z),lp; ring R=0,X,lp;

map h=S,X-1,X2;
```

Damit ergibt sich nach Wahl des jeweils aktuellen Ringes

```
setring S; poly g=YZ; setring R;

h(g);
```

das erwartete Resultat $\mathtt{h(g)=X}^3-\mathtt{X}^2$. ❑

Für Matrizen-Operationen stehen umfangreiche Hilfsmittel zur Verfügung. Zunächst definieren wir die Matrix

$$\mathtt{A}=\begin{pmatrix}\mathtt{X}^2 & 2\mathtt{X} & 3\\ 4 & 5 & 6\end{pmatrix}\in \mathrm{M}(2,3;\mathbb{Q}[\mathtt{X}])$$

durch den Singular-Befehl

```
matrix A[2][3]=X2,2*X,3,4,5,6;
```

(wobei die Einträge fortlaufend zeilenweise einzugeben sind). Eine weitere Matrix erklären wir durch

```
matrix B[2][2]=1,1,1,2;
```

und erhalten das erwartete Resultat für das Produkt mit

```
A*B;
```

oder in gefälligerer Form mittels

```
print(A*B);
```

Manche Funktionen erfordern das Laden zusätzlicher Bibliotheken, beispielsweise zur linearen Algebra, durch

```
LIB “linalg.lib“;
```

Für die invertierbare Matrix `B` erhalten wir nun $\mathtt{B}^{-1}$ mittels

```
print(inverse(B));
```
❑

Natürlich erwarten wir, dass auch das Rechnen mit Näherungen möglich ist. 2/6/3
Dafür werden Datentypen eingeführt, die eine gewisse Ähnlichkeit mit Polynomalgebren über den reellen Zahlen aufweisen. So wird durch die Vereinbarung

```
ring r=(real,30),U,lp;
```

das Rechnen mit Zahlen in 30-stelliger Gleitkommadarstellung möglich, wobei intern nochmals dieselbe Stellenzahl hinzugenommen wird, um (möglichst) die Stabilität der Operationen zu gewährleisten. Obwohl Rechnungen dieser Art nicht exakt sind und in die Irre führen können, lässt sich bei sorgfältiger Fehlerabschätzung meist ein brauchbares Resultat gewinnen.

Vgl. auch das erste Beispiel in 2/3/2.

Beispiel. (*Iterationsverfahren für lineare Gleichungssysteme*)

$$(*) \qquad \mathtt{A\ x\ =\ b} \text{ mit } \mathtt{A} \in \mathrm{M}(n;\mathbb{R}),\ \mathtt{b} \in \mathrm{M}(n,1;\mathbb{R})$$

sei ein lineares Gleichungssystem mit einer regulären Koeffizientenmatrix A. Wir wissen bereits, dass dann eine eindeutig bestimmte Lösung $x \in \mathrm{M}(n,1;\mathbb{R})$ existiert. Ist die Zahl n groß, dann kann das System aufgrund des hohen Rechenaufwands und zahlreicher Rundungsfehler „praktisch unlösbar“ werden. Daher ist es ratsam so zu rechnen, dass nicht zu viele Zwischenschritte erforderlich sind und eventuelle Fehler sich möglichst automatisch korrigieren; die folgende Methode kann dabei unter Umständen helfen.

Es sei $\mathtt{A} = \mathtt{B} + \mathtt{C}$ mit einer ebenfalls regulären Matrix $\mathtt{B} \in \mathrm{M}(n;\mathbb{R})$, die jedoch „leichter“ zu invertieren ist als die ursprüngliche Matrix A. Dann ist die Bedingung $(*)$ offensichtlich äquivalent zum System

$$\binom{*}{*} \qquad \mathtt{x} = -\mathtt{B}^{-1}\mathtt{C}\mathtt{x} + \mathtt{B}^{-1}\mathtt{b},$$

das zunächst viel schlechter aussieht als das erste. Wir betrachten die Abbildung

$$\mathtt{f} : \mathrm{M}(n,1;\mathbb{R}) \to \mathrm{M}(n,1;\mathbb{R}),\quad \mathtt{x} \mapsto -\mathtt{B}^{-1}\mathtt{C}\mathtt{x} + \mathtt{B}^{-1}\mathtt{b}.$$

x ist genau dann eine Lösung von $\binom{*}{*}$, wenn $\mathtt{f}(\mathtt{x}) = \mathtt{x}$. Unter einer geeigneten Beschränktheitsbedingung für $\mathtt{B}^{-1}\mathtt{C}$ folgt aus einem Satz der Analysis (*banachscher Fixpunktsatz*), dass jede Folge $(\mathtt{x}_i)_{i\in\mathbb{N}}$, $\mathtt{x}_i \in \mathrm{M}(n,1;\mathbb{R})$, für die $\mathtt{f}(\mathtt{x}_i) = \mathtt{x}_{i+1}$ ist, gegen die eindeutig bestimmte Lösung x des gegebenen Systems konvergiert. Wir werden das hier nicht weiter erläutern, sondern begnügen uns mit einem numerischen Experiment, das durch das Auffinden von x mit $\mathtt{f}(\mathtt{x}) = \mathtt{x}$ gerechtfertigt wird. Wir wählen

Idee: Nehmen wir an, wir hätten eine konvergente Folge (x_i) mit $x_{i+1} = f(x_i)$.
Dann folgt nach Wahl von f zunächst $x_{i+1} = -B^{-1}Cx_i + B^{-1}b$.
Sofern die Matrizen-Operationen mit der Grenzwertbildung für $i \to \infty$ vertauschbar sind, ergibt die letztere Gleichung mit dem Grenzwert $x = \lim_{i\to\infty} x_i$ offenbar $x = f(x)$.

$$\mathtt{A} = \begin{pmatrix} 1 & 0 & 0.1 \\ 0 & 1 & 0.1 \\ 0.1 & 0.1 & 1 \end{pmatrix},\quad \mathtt{b} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix};$$

dies entspricht der Eingabe

```
matrix A[3][3]=1,0,0.1,0,1,0.1,0.1,0.1,1;
matrix b[3][1]=1,0,1;
```

Nun wird eine reguläre Matrix B so gewählt, dass die Differenz $\mathtt{C} = \mathtt{A} - \mathtt{B}$ „klein“ ist; für die Einheitsmatrix $\mathtt{B} = \mathrm{E}_3$ scheint dies plausibel:

```
matrix B[3][3]; B=B+1;
```

(Singular-Definition einer Diagonalmatrix, deren Diagonalelemente 1 sind).

```
matrix C[3][3]=A-B;
```

Zur Bequemlichkeit beschreiben wir die oben definierte Abbildung f durch eine Prozedur

```
proc f (x) {return(-C*x+b);}
```

und definieren x als

```
matrix x[3][1];
```

Ersetzen des Anfangswerts von x (hier ist das die Nullspalte) durch f(x) drückt sich im Befehl

```
x=f(x);
```

aus. In der Hoffnung, dass die nach diesem Muster gebildete Folge konvergiert, lassen wir die Substitution wiederholen, solange `x` und `f(x)` verschieden sind,

```
while (x!=f(x)) {x=f(x);}
```

(`!=` bedeutet $\neq$). Was geschieht hier? Zunächst wird geprüft, ob $\mathtt{x} \neq \mathtt{f(x)}$. Ist dies der Fall, so wird `x` durch `f(x)` ersetzt und der erste Schritt wiederholt; anderenfalls ist die Rechnung beendet.
Das schnell erhaltene Resultat lässt sich durch

```
print(x);
```

anzeigen; es ist $\mathtt{x} = \begin{pmatrix} 0.90816326530612244897959183673 5 \\ -0.0918367346938775510204081632653 \\ 0.918367346938775510204081632653 \end{pmatrix}$.

Wer Lust dazu hat, kann leicht mit dem gaußschen Algorithmus die exakte Lösung des Systems bestimmen und sich davon überzeugen, dass sie bis auf die angegebene Stellenzahl mit x übereinstimmt.

Dieses Verfahren funktioniert in vielen Fällen. Ist die Matrix `B` in $\binom{*}{*}$ eine Diagonalmatrix, so wird die hier erläuterte Methode als *Jacobi-Verfahren* bezeichnet, bei Wahl einer unteren Dreiecksmatrix für `B` als *Gauß-Seidel-Verfahren.* ❏

Der zentrale Algorithmus in Singular dient der Bestimmung von *Standardbasen* für Ideale in verschiedenen Typen von Ringen; wir haben sie für Polynomringe unter dem Namen Gröbnerbasen kennen gelernt. Die Syntax ist einfach. 2/6/4

```
ring r=0,(X1,X2,X3),lp;

ideal i=X1^5+X2^5-X3^5,X1^7+X2^7-X3^7;

option(redSB); std(i);
```

Wir erhalten eine reduzierte Gröbnerbasis aus 21 Polynomen. Auch die Entscheidung, ob das Polynom $\mathtt{f} = \mathtt{X}_1^9 + \mathtt{X}_2^9 - \mathtt{X}_3^9$ in dem gegebenen Ideal liegt, wird uns leicht gemacht:

```
poly f=X1^9+X2^9-X3^9; reduce(f,std(i));
```

Der so erhaltene Rest bei Division durch `std(i)` ist von Null verschieden, d.h. `f` liegt nicht in dem gegebenen Ideal.
Nun kann es interessant sein, anstelle der lexikographischen Ordnung `lp` dieselbe Rechnung mit anderen Monomordnungen auszuführen.

```
ring r=0,(X1,X2,X3),M(1,1,1,1,0,1,1,2,3);
```

definiert einen Polynomring mit der Monomordnung, die durch die reguläre Matrix

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 2 & 3 \end{pmatrix}$$

gegeben ist. Wie zuvor erhalten wir eine Standardbasis, die in diesem Fall aus nur 6 Polynomen besteht.

```
ideal i=X1^5+X2^5-X3^5,X1^7+X2^7-X3^7;

std(i);
```

Eine häufig verwendete Monomordnung ist die graduiert invers-lexikographische Ordnung, die in Singular mit `dp` bezeichnet wird; Rechnungen mit

dieser Ordnung erweisen sich oft als besonders effektiv, sie ist jedoch nicht direkt für Elimination von Unbestimmten verwendbar.
Unter Verwendung von `dp` im angegebenen Beispiel entsteht eine weitere Gröbnerbasis für das gegebene Ideal, ebenfalls aus 6 Polynomen. ❏

Sachverzeichnis