F A C H H O C H S C H U L E W E D E L

Computeralgebra

Thema 1: Vorstellung des Computer-Algebra-Systems Mathematica

Informatik Seminararbeit

WS 2007/2008

von

Sigrun Reimitz

Wedel, den 13. Dezember 2007

Inhaltsverzeichnis

1 Einleitung **2**

2 Mathematica Grundlagen **3**
2.1 Bestandteile von Mathematica 3
2.2 Bedienung der Oberfläche 4
2.3 Regeln der Mathematica-Sprache 5
2.4 Hilfe 7

3 Arbeiten mit Mathematica **8**
3.1 Exakte Arithmetik 8
3.2 Symbolische Algebra 9
3.3 Graphische Darstellung 10
3.4 Polynome und Polynomfaktorisierung 12
3.5 Gleichungen und Gleichungssysteme 12
3.6 Listen, Vectoren und Matrizen 14

4 Programmieren mit Mathematica **16**
4.1 Funktionen programmieren 16
4.2 Verzweigungen und Schleifen 17
4.3 Mustererkennung 18
4.4 Rememberprogrammierung 20

5 Zusammenfassung **21**

1 Einleitung

Die vorliegende Seminararbeit bietet eine Einführung in Mathematica. Der Autor beschränkt sich dabei auf ausgewählte Gebiete der Mathematik in Anlehnung an Kapitel 1 und 2 von [KOEPF].

Der Abschnitt 2 auf Seite 3 stellt die Benutzerführung von Mathematica vor. Dem vorangestellt, geht der Autor kurz darauf ein, was Computeralgebra-Systeme sind und was von ihnen erwartet werden kann. In diesem Abschnitt erhält der Leser zudem einen Überblick über hilfreiche Eingabetechniken. Des Weiteren werden die Regeln der Mathematica-Sprache kurz umrissen und im letzten Unterabschnitt auf Programm-Hilfestellungen hingewiesen.

Abschnitt 3 auf Seite 8 umfasst ausgewählte Bereiche aus Mathematica: Arithmetik, symbolische Algebra, Polynome und Polynomfaktorisierung sowie Gleichungssysteme. In dem Unterabschnitt 3.3 erhält der Leser einen Eindruck von den Möglichkeiten zur graphischen Darstellung.

Einen kurzen Überblick über die Programmierung von Funktionen erhält der Leser im Abschnitt 4 auf Seite 16.

Die Verwendung von Mathematica ist damit nicht umfassend erläutert. Es wird auf die im Literaturverzeichnis aufgelisteten Bücher verwiesen.

Der Inhalt dieser Arbeit ist eine schriftliche Ergänzung zum Seminarvortrag[1]. Der Leser findet die in der Veranstaltung vorgeführten Beispiele wieder. Diese Beispiele sind mit der Mathematica Version 4.1. vorgeführt worden. Die dabei benutzten Tastaturkürzel sind für das Betriebssystem Windows verwendbar. Für eine Benutzung auf anderen Systemen wird auf die Literatur verwiesen (z. B. für Linux im [KOEPF][2]). Der Leser kann mit dem kostenlosen Mathematica Player[3] die Ergebnisse für die Beispiele einsehen. Es wird deshalb zum Teil auf die Darstellung der Ergebnisse verzichtet, bzw. Ergebnisse im nachfolgenden Text kommentiert.

[1] Siehe [VORTRAG]
[2] Vgl. S. 603
[3] Der kostenlose Mathematica Player ist erhältlich unter der Adresse [MATHEMATICA].

2 Mathematica Grundlagen

Mathematica ist ein Computeralgebra-System: Computeralgebra-Systeme entstehen zum einen aus der Integration verschiedener algebraischer Algorithmen, die in der Computeralgebra entwickelt wurden (und werden), zum anderen tragen sie zu der Entwicklung von Algorithmen bei.[4] Die Computeralgebra beschäftigt sich mit der symbolischen Manipulation von algebraischen Ausdrücken. Diese Ausdrücke werden trotz Anwendung von Algorithmen exakt im Hauptspeicher dargestellt, da eine Ersetzung von Symbolen mit Näherungswerten im System vermieden wird. Neben den algebraischen Funktionalitäten beherrschen Computeralgebra-Systeme auch andere Teile der Mathematik, z. B. Differential- und Integralrechnung oder Optimierungsverfahren.

Kommen wir nun zu Mathematica: Hier handelt es sich um ein Softwarepaket, das den Anwender bei der Lösung mathematischer Probleme unterstützt. Um diese Software im vollen Umfang nutzen zu können, werden Grundlagen der Mathematik und mathematische Konzepte beim Anwender vorausgesetzt.

Mathematica stellt dem Nutzer eine Entwicklungsumgebung zur Verfügung, in die eine ganze Reihe von Werkzeugen für numerische und algebraische Berechnungen sowie die symbolische Behandlung von mathematischen Funktionen integriert sind. Das Programm kann dabei auf eine umfangreiche Bibliothek zurückgreifen, die einen großen Teil des aktuellen mathematischen Wissens in Form von Formeln bereit hält. Ein weiterer Vorzug von Mathematica sind seine besonderen Fähigkeiten bei der Visualisierung von Funktionen und anderem Zahlenmaterial. Die Möglichkeiten reichen hier bis zur dreidimensionalen Darstellung (vgl. Abschnitt 3.3).

2.1 Bestandteile von Mathematica

Mathematica besteht aus zwei separaten Teilen: Frontend und Kernel. Das Frontend regelt die Interaktion mit dem Benutzer und der Kernel führt die Berechnung aus. Das Grundschema der Arbeit mit Mathematica ist die Eingabe des Inputs, dessen Weitergabe an den Kernel und die Rückgabe des Outputs vom Kernel.

Die Sicherung der Dateien erfolgt in sog. Notebooks mit der Endung *.nb. Der Inhalt eines Notebooks lässt sich in Abschnitte einteilen (Zellen). Es stehen unterschiedliche Formatierungszellen (Input, Output, Text usw.) zur Verfügung, deren Anwendung u. a. eine kommentierte Weitergabe oder Vorführung einer mathematischen Theorie ermöglicht. Die Gliederung eines Notebooks ist erkennbar an den Zellenklammern am rechten Rand, die die Art und die Reichweite eines Abschnittes (Zelle) anzeigen. Um einen beliebigen Ausdruck in Mathematica auswerten zu können, muss dieser zunächst in ein Notebook eingegeben werden. Die Eingabe erfolgt in einer Zelle.

[4] Vgl. [SCHWARDMANN], S. 1

Mathematica ist ein erweiterbares System. Sollte der Anwender Funktionen für spezielle Gebiete vermissen, sind diese über Packages nachzuladen. Der Autor verweist hierzu auf [WOLFRAM][5], um die Einbindung der Pakete durchzuführen.

2.2 Bedienung der Oberfläche

Mathematica erzeugt in einem Notebook standardmäßig Input-Zellen für die Eingabe mathematischer Ausdrücke. Eine Zelle darf mehrere Zeilen umfassen.

Für die Eingabe gibt es folgende Alternativen:

- Eingabe über die Tastatur in eindimensionaler Form der Art: +, -, *, /, ^

  In[1]:= $(5\ \hat{}\ 2 - 1 * 30)/-5$

- Eingabe in einer optisch ansprechenden und übersichtlichen Form mit Hilfe von Tastaturbefehlen[6]:

STRG+/	Bruch	$\frac{a}{b}$
STRG+2	Quadratwurzel	$\sqrt{a}$
STRG+^	Exponentialausdruck	a^2
STRG+_	Index tiefgestellt	a_b
STRG+LEER	beendet die Eingabe	

  So könnte ein Beispielsausdruck aussehen:

  In[2]:= $\frac{x^2 + a * b}{c}$

- Mathematica bietet eine Eingabehilfe über Paletten an. Diese Paletten sind in dem Menüpunkt `File/Palettes` zu finden und sind ähnlich verwendbar wie ein Formeleditor[7]. Eine Eingabe in dieser Form bedingt Mausbenutzung.

- Die so genannte FullForm nutzt die Darstellung über Mathematica-Funktionen. Intern wird in dieser Darstellungsform gerechnet (siehe auch Abschnitt 4.3).

  In[3]:= $Divide[Plus[Power[x, 2], Times[a, b]], c]$

[5] Vgl. [WOLFRAM], S. 60 ff.
[6] Das ist nur eine kleine Auswahl von Möglichkeiten. Es wird auf [KOEPF], S. 69 verwiesen.
[7] Z. B. der Formeleditor bei MS Office.

Die Ergebnisse der Darstellungform In[2]:= und In[3]:= sind identisch. Um die vollständige (interne) Form eines Ausdruckes zu sehen, ist:

In[4]:= $FullForm[\%]$

anzuwenden. Das führt zur Ausgabe: `Times[Power[c,-1], Plus[Times[a,b], Power[x,2]]]` und gibt einen Einblick in die Art, wie Mathematica Ausdrücke verarbeitet.

Eine Berechnung wird über `Shift+Return` angewiesen. Erst dann erzeugt die Mathematica-Oberfläche `In[n]` und `Out[n]` zur Unterscheidung der Ein- und Ausgabezellen. Mathematica nummeriert alle Zellen entsprechend der Auswertungsreihenfolge. In dieser Ausarbeitung wird auf die Nummerierungsreihenfolge `In[n]` aus dem Vortrag Bezug genommen. Eine Berechnung kann über `Strg+.` oder `Shift+c` abgebrochen werden für den Fall einer Endlosschleife beim Programmieren oder einer zu langen Berechnung von Mathematica.

Auf die vorangegangenen Berechnungen kann mit dem Symbol % zugegriffen werden. Die Eingabe von Funktionen ist in der Standardform für $f[x, y]$, in der Präfixform für $f[x]$, in der Postfixform für $f[x]$ und in der Infixform für $f[x, y]$ möglich. In dieser Seminararbeit wird nur in der Standardform (z. B. `FullForm[`x^2`]`) oder Postfixform (z. B. x^2`//FullForm`) gearbeitet. Die Handhabung der anderen Darstellungsformen ist nachzulesen bei [WOLFRAM][8].

2.3 Regeln der Mathematica-Sprache

In Mathematica sind Großbuchstaben für Funktionen und Anweisungen reserviert. Besteht ein Befehl aus mehreren Wörtern, werden diese ohne Leerzeichen aneinander gereiht; neue Wörter beginnen wiederum mit einem Großbuchstaben. **Groß- und Kleinschreibung** ist signifikant. Die Argumente einer Funktion stehen in eckigen Klammern.

Beispiele: `Pi, FullForm, Sqrt[Pi]`

Der **Unterstrich** ist reserviert für formale Parameter in selbst definierten Funktionen (siehe hierzu Abschnitt 3.2 mit einem Beispiel zur Kurvendiskussion oder Abschnitt 4). Um sich in dieser Seminararbeit von den Systemfunktionen abzusetzen, beginnen die selbst definierten Variablen und Funktionen mit einem Kleinbuchstaben.

Mathematica kennt eine Reihe vordefinierter **Konstanten**. Eine andere Darstellungsform wird erreicht durch die Tastenkombination `ESC<Konstante>ESC`. Die folgende Übersicht zeigt eine Auswahl von Konstanten und ihre Darstellungsform mithilfe von `ESC`.

[8] Vgl. [WOLFRAM], S. 243

Pi	ESC p ESC	π Kreisteilungszahl
E	ESC e3 ESC	$\mathfrak{e}$ Eulersche Zahl
I	ESC i ESC	i imaginäre Einheit
Infinity	ESC inf ESC	∞ unendlich

Selbst definierte Konstanten sind mit `Protect` zu vereinbaren. Im folgenden Beispiel wird π konstant auf den Wert $3,14$ festgelegt:

Beispiel: `p=3.14; Protect[p]`

Variablenvereinbarungen sind immer global. Daher ist die Verwendung von Variablen kritisch und erzeugt evtl. ungewolltes Verhalten. Mit `Remove["Global`* "]` werden alle global vereinbarten Variablen gelöscht. `Clear[p]` entfernt nur eine Variable, hier also die eben vereinbarte Variable `p`.

Nachfolgend sind Beispiele zur Variablenvereinbarung aufgelistet:

- $f1 := x^2 + tx + 1$
  $f1$ ist eine Wertvariable, in der ein symbolischer Ausdruck mit den Symbolvariablen x und t gespeichert ist.
- $f2[x\_] := x^2 + tx + 1$
  $f2$ ist eine einstellige Funktion, die von einer weiteren Symbolvariablen t abhängt.
- $f3[x\_, t\_] := x^2 + tx + 1$
  $f3$ ist eine zweistellige Funktion.

Nicht immer ist die globale Verwendung von Variablen gewünscht (z. B. bei der Erstellung eigener Programme). Sollen Variablen lokal Anwendung finden, ist folgendes zu vereinbaren: `Module[{a,b,...}, proc]` bedeutet "Funktion `proc` mit den lokalen Variablen a, b ...". Für weitere Informationen verweist der Autor auf [WOLFRAM][9].

Verschiedene Objekte lassen sich mit dem Klammerpaar {} zu **Listen** zusammenfassen. Die Elemente der Liste werden mit Komma separiert (siehe auch Abschnitt 3.6). Berechnungen lassen sich in einer Folge von Schritten - getrennt durch ein **Semikolon** - zusammenfassen. Endet eine Anweisung mit einem Semikolon, unterdrückt dieses die Anzeige im Output.

[9] Vgl. [WOLFRAM], S. 113

In Mathematica wird zwischen den **Operatoren** =, := und == unterschieden:

- Der Zuweisungsoperator = weist dem Symbol auf der linken Seite den vorher ausgewerteten Ausdruck auf der rechten Seite zu.
- Der Operator := heißt "verzögerter Zuweisungsoperator". Er weist dem Symbol auf der linken Seite den Ausdruck der rechten Seite zu. Eine Auswertung des Ausdruckes erfolgt erst bei Verwendung des Symbols.
- Der Operator == ermittelt, ob der Ausdruck auf der linken Seite und der Ausdruck auf der rechten Seite gleich sind, wobei die Ausdrücke vor dem Vergleich ausgewertet werden.

2.4 Hilfe

Mathematica stellt eine umfassende Online-Hilfe über den Menüpunkt HELP zur Verfügung. Darüber hinaus ist zu jeder Funktion ein Hilfetext verfügbar. Sollte also eine Erklärung zu `Times` gewünscht sein, erhalten wir mit

In[5]:= $?Times$

die Ausgabe: "x*y*z or x y z represents a product of terms".

Für die Verwendung der umfangreichen Sammlung an Funktionen ist die Tastaturkombination `STRG+K` zu empfehlen. Das erzeugt eine Liste mit Funktionen, die mit den bereits eingegebenen Buchstaben beginnen.

3 Arbeiten mit Mathematica

3.1 Exakte Arithmetik

Mathematica führt alle Rechnungen aus, die auch ein klassischer Taschenrechner beherrscht, allerdings mit hoher Genauigkeit. Es wird erst dann eine Ersetzung von symbolischen Ausdrücken vorgenommen, wenn ein Muster (siehe auch Abschnitt 4.3) gefunden wurde.

Dazu ein Beispiel. Der Ausdruck:

In[6]:= 2^{150}

wird genau ermittelt. Die Berechnung von:

In[7]:= $2 * \pi$

wird jedoch nicht mit einem Näherungswert von π durchgeführt, um Rundungsfehler zu vermeiden. Sollte dennoch eine numerisches Berechnung gewünscht sein, liefert die Funktion `N` ein angenähertes numerisches Ergebnis. Die Anzahl der Stellen beeinflusst das zweite Argument der Funktion. So ergibt folgendes Beispiel die Ausgabe von 30 Stellen:

In[8]:= $N[\pi, 30]$

Mathematica stellt eine große Auswahl mathematischer Funktionen zur Verfügung. Als erstes Beispiel sei hier:

In[9]:= $FactorInteger[12]$

genannt für die Bestimmung der Primfaktoren einer Zahl n. Wir erhalten das Resultat:

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

was bedeutet, dass der Teiler 2 zweimal vorkommt und der Teiler 3 einmal.

Das zweite Beispiel ist eine boolesche Funktion, mit der eine Primzahl überprüft werden kann:

In[10]:= $PrimeQ[2^{31} - 1]$

Das Ergebnis liefert `True` und bestätigt damit die von Euler 1772 gefundene Primzahl.[10]

[10] Vgl. [WIKIPEDIA]

3.2 Symbolische Algebra

Symbolische Berechnungen in der Mathematik sind Umformungen eines Ausdrucks nach gewissen vorher festgelegten oder bewiesenen Regeln in einen anderen Ausdruck. Mathematica wendet so lange die bekannten Transformationsregeln auf einen Ausdruck an, bis eine Stufe der Vereinfachung erreicht wurde, die als Zwischenschritt verwendet werden kann.[11]

In[11]:= $Sqrt[z]\hat{}\,2(3x-y-2x)(y+x)z\hat{}-1$

Dieser Ausdruck wird intern vereinfacht zu $(x-y)(x+y)$. Ein weiteres Ausmultiplizieren unterbleibt.

Eine solche Umformung wird nicht in allen Fällen vollzogen, deshalb lassen sich Umformungen vom Benutzer steuern, z. B. mit `Simplify`. Im folgenden wird diese Mathematica-Funktion in Postfix-Notation[12] angewendet.

In[12]:= $\frac{a}{(a-b)(a-c)}+\frac{b}{(b-a)(b-c)}+\frac{c}{(c-a)(c-b)}//Simplify$

Mithilfe der Funktion `Simplify` wird der Ausdruck zu Null vereinfacht. Mathematica wendet dabei dieselben Umformungen wie beim Rechnen mit rationalen Zahlen an.

Der Leser möge sich davon überzeugen, indem die Funktion `PolynomialLCM`[13] auf den Nenner des Ausdruckes In[12]:= angewendet wird.

In[13]:= $PolynomialLCM[(a-b)(a-c),(b-a)(b-c),(c-a)(c-b)]$

Wir erhalten: $(a-b)(a-c)(b-c)$ als neuen Nenner. Und damit ergibt sich durch weitere Umformungsschritte der Wert Null.

Für viele Aufgaben aus der Analysis sind Computeralgebra-Systeme wie Mathematica ein nützliches und zeitsparendes Hilfsmittel.[14] Nachfolgend werden Teile der Kurvendiskussion demonstriert.

Zu betrachten sei die Funktion[15]:

In[14]:= $f[x\_]=\frac{x^2}{x^2-4}$

Die Nullstellen lassen sich mit der Mathematica-Funktion `Solve` bestimmen:

In[15]:= $Solve[f[x]==0,x]$

[11] Vgl. [WOLFRAM], S. 66
[12] Die Postfix-Notation kann bei allen einstelligen Funktionen (mit einem Argument) eingesetzt werden.
[13] PolynomialLCM liefert das kleinste gemeinsame Vielfache bei Polynomen.
[14] Vgl. [BURKHARDT], S. 73
[15] Wie bereits in Abschnitt 2.3 erläutert, dient der Unterstrich zur Kennzeichnung von formalen Parametern einer Funktion. Beim späteren Aufrufen der Funktion entfällt der Unterstrich.

Nicht unerwartet gibt Mathematica die Nullstelle `x->0` als Ergebnis aus. Da es sich um eine quadratische Funktion im Zähler handelt, wird die Ausgabe um eine zweite Nullstelle ergänzt: hier also eine Wiederholung von `x->0`.

Für die Ermittlung von Polstellen wird der Nenner von In[14]:= betrachtet mit:

In[16]:= $Solve[Denominator[f[x] == 0, x]]$

Die Ordinate der horizontalen Asymptote wird als Grenzwert der Funktion bestimmt:

In[17]:= $Limit[f[x], x- > \infty]$

Für die weitere Analyse ist die erste Ableitung:

In[18]:= $f1[x\_] = f'[x]$

und zweite Ableitung zu errechnen.

In[19]:= $f2[x\_] = f''[x]$

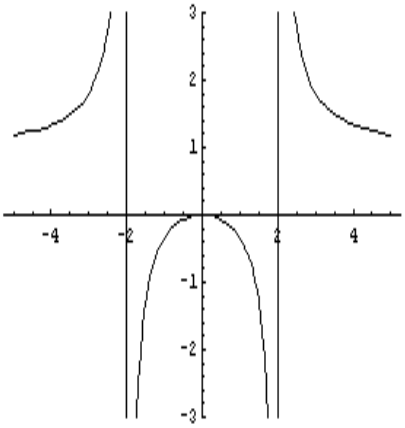
Mit den bisher ermittelten Ergebnissen ist die Kurve im nächsten Abschnitt 3.3 darstellbar.

3.3 Graphische Darstellung

Eine große Stärke von Mathematica ist das Erstellen von Grafiken. Diese werden vom Mathematica-Kernel als symbolische Ausdrücke geführt, die sich durch spezielle Kommandos (z. B. `Plot`) darstellen lassen. Dabei werden diese Grafikobjekte in ein PostScript-Format umgewandelt, auf dessen Basis die Anzeige erfolgt.[16]

Die Darstellung der Kurven aus Abschnitt 3.2 ist wie folgt möglich:

In[20]:= $Plot[f[x], \{x, -5, 5\}, PlotRange- > \{-3, 3\}];$

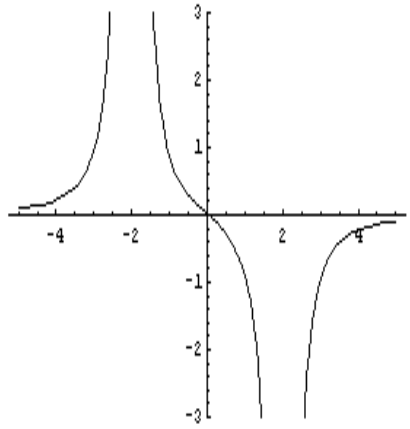


[16] Vgl. [KOEPF], S. 43

Im ersten Argument ist die zu zeichnende Funktion angegeben. Das zweite Argument enthält eine Liste mit der (einen) Variablen und den Grenzen des darzustellenden Wertebereichs. `PlotRange` beeinflusst den anzuzeigenden x- und y-Achsenbereich.

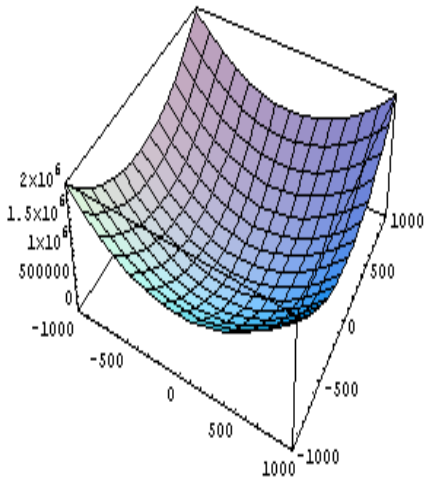
Und ebenso wird die zweite Ableitung dargestellt:

In[21]:= $Plot[f'[x], \{x, -5, 5\}, PlotRange-> \{-3, 3\}];$



Eine dreidimensionale Darstellung ist mit dem Befehl:

In[22]:= $Plot3D[x^2 + y^2, \{x, -1000, 1000\}, \{y, -1000, 1000\}];$



möglich. Diesem Befehl müssen ein Funktionsterm mit zwei Variablen sowie die Bereiche der beiden Variablen übergeben werden.

3.4 Polynome und Polynomfaktorisierung

Algebraische Ausdrücke können auf verschiedene Weise dargestellt werden. Zum Beispiel kann der Ausdruck $(a+b)^2$ entsprechend binomischer Formel auch als $(a^2+2ab+b^2)$ geschrieben werden. Im folgenden werden zwei Mathematica-Funktionen vorgestellt, die algebraische Ausdrücke ineinander überführen.

Die Funktion `Expand` multipliziert Produkte und Potenzen aus und schreibt das Ergebnis als Summe von Termen. `Factor` versucht, den Ausdruck in ein Produkt von minimalen Faktoren zu zerlegen.

In[23]:= $Expand[(x+y+z)^3]$

Der Ausdruck In[23]:= gibt die ausmultiplizierte Form an:
$x^3+3yx^2+3zx^2+3y^2x+3z^2x+6yzx+y^3+z^3+3yz^2+3y^2z$

Der Ausdruck In[24]:= stellt die ursprüngliche Form $(x+y+z)^3$ wieder her:[17]

In[24]:= $Factor[\%]$

Mathematica kann mit Polynomen umgehen, deren Koeffizienten im endlichen Körper Z_p der ganzen Zahlen modulo einer Primzahl p liegen. Folgender Ausdruck ergibt keine weitere Zerlegung:[18]

In[25]:= $Factor[x^6+x^2+1]$

Die Zerlegung ist hingegen im Polynomring $Z_{13}[x]$ möglich:

In[26]:= $Factor[x^6+x^2+1, Modulus->13]$

und ergibt: $(x^2+6)(x^4+7x^2+11)$.

3.5 Gleichungen und Gleichungssysteme

In diesem Abschnitt wird gezeigt, wie Lösungen für eine Gleichung, für ein Gleichungssystem ohne Freiheitsgrade und für ein Gleichungssystem mit Freiheitsgraden ermittelt werden können.

Die Mathematica-Funktion `Solve` löst eine Gleichung und gibt eine Liste von Lösungen für x aus.

In[27]:= $Solve[x^2-3x-1==0, x]$

[17] Hier wird mit % auf das Ergebnis von Expand zugegriffen.
[18] Vgl. [KOEPF], S. 9

Die Gleichung In[27]:= ist eindeutig nach x lösbar und ergibt zwei Lösungen:
$\{\{x- > \frac{1}{2}(3-\sqrt{13})\}, \{x- > \frac{1}{2}(3+\sqrt{13})\}\}$

Das Gleichungssystem In[28]:= soll nach x und y aufgelöst werden. Auch hier kann eine eindeutige Lösung bestimmt werden.

In[28]:= $Solve[\{2x+3y == 5, 3x+4y == 11\}, \{x, y\}]$

Die Lösung des 3. Beispiels In[29]:= ist von der Variablen a abhängig.

In[29]:= $Solve[\{y-3a == x, 2x-a == y\}, \{x, y\}]$

Der Versuch, die Variable a zu eliminieren mit:

In[30]:= $Solve[\{y-3a == x, 2x-a == y\}, \{x\}, \{a\}]$

wird erfolgreich sein, wie folgendes Ersetzungsschema zeigt.

I	$x-y$	$+3a = 0$
II	$2x-y$	$-a = 0$
III	$7x-4y$	$= 0$

Die Gleichung III lässt sich berechnen aus I+3∗II.

Mathematica lässt sich auch für das Lösen von Optimierungsaufgaben einsetzen. Eine typische Aufgabe ist dem Material[19] von Herrn Prof. Gerhardt aus Operation Research entnommen. Inhalt der Aufgabe ist es, bei einem Transport durch die Wüste mit einer bestimmten Anzahl von Kamelen und Dromedaren minimale Leihgebühren für die Tiere zu zahlen. Die Restriktionen sind in der folgenden Tabelle zusammengefasst.

	Dromedare	Kamele	Restriktion
Steine	50	100	≥1000
Heu	4	3	≤60
Wasser	80	100	≤1600
Kosten	5	11	Minimum

Die Schreibweise für dieses Minimierungsproblem in Mathematica ist:[20]

In[31]:=
$ConstrainedMin[5x+11y,$
$\{50x+100y >= 1000,$
$4x+3y <= 60,$
$80x+100y <= 1600\},$
$\{x, y\}]$

[19] Vgl. [GERHARDT], S. 2-19
[20] Ab Mathematica Version 5.0 lautet die Syntax: `NMinimize` bzw. `NMaximize`

Der erste Teilausdruck $5x + 11y$ entspricht der zu minimierenden Zielfunktion. Die Restriktionen werden in geschweiften Klammern dargestellt. Die Lösungsvariablen sind in dem letzten Klammerausdruck zu finden.[21]

Die Lösung des Problems: $\{104, \{x- > 12, y- > 4\}\}$ bedeutet für die Wüsten-Karawane einen Einsatz von 12 Dromedaren (x) und 4 Kamelen (y) bei minimalen Kosten von 104 Geldeinheiten.

3.6 Listen, Vectoren und Matrizen

Listen bieten die Möglichkeit, Objekte zusammenzufassen. Mengen, Vektoren, Matrizen werden in Mathematica als Listen dargestellt. Sie haben folgende Eigenschaften:

- In Listen stehen die Elemente zwischen zwei geschwungenen Klammern.
- Eine Liste kann einer Variablen zugewiesen werden.
- Listen sind in Operationen verwendbar.
- Listen können Argumente von Funktionen sein.

Eine Liste mit 3 Elementen sieht bspw. so aus:

In[32]:= $v = \{a, b, c\}$

Listen lassen sich für die Bildung von Vektoren (die Variable v kann als Vektor interpretiert werden) und Matrizen verwenden:

In[33]:= $m1 = \{\{a, b\}, \{c, d\}, \{e, f\}\}$

$$\begin{pmatrix} a & b \\ c & d \\ e & f \end{pmatrix}$$

In[34]:= $m2 = \{\{1, 2, a\}, \{4, 5, 6\}\}$

$$\begin{pmatrix} 1 & 2 & a \\ 4 & 5 & 6 \end{pmatrix}$$

[21] Für das weitergehende Verständnis zum Lösen einer Optimierungsaufgabe wird auf das Material von Prof. Gerhard verwiesen.

Um Matrizen in Mathematica zu multiplizieren, verbindet man die Matrizen mit einem Punkt. Diese Multiplikation entspricht dem in der Mathematik gebräuchlichen Verfahren.

In[35]:= $m3 = m1.m2$

$$\begin{pmatrix} a+4b & 2a+5b & a^2+6b \\ c+4d & 2c+5d & ac+6d \\ e+4f & 2e+5f & ae+6f \end{pmatrix}$$

Der Zugriff auf die Elemente einer Liste ist über die Angabe des Index möglich:

In[36]:= $m1[[3,2]]$

ergibt das zweite Element der Spalte drei von Matrix $m1$.

Die Anwendung der Funktion

In[37]:= $Transpose[m3]$

führt zur transponierten[22] Matrix:

$$\begin{pmatrix} a+4b & c+4d & e+4f \\ 2a+5b & 2c+5d & 2e+5f \\ a^2+6b & ac+6d & ae+6f \end{pmatrix}$$

[22] Transponieren bedeute das Vertauschen der Zeilen und Spalten der Ausgangsmatrix. Es entsteht eine veränderte Matrix.

4 Programmieren mit Mathematica

Mit Mathematica können eigene Programme erstellt werden. Die folgenden Beispiele setzen ein Grundverständnis über lokale und globale Variablen, Programmiertechniken wie Verzweigungen sowie Iteration und Rekursion voraus. Die ersten beiden Abschnitte dieses Kapitels beschreiben den Aufbau von Programmen und die Verwendung von Schleifenkonstrukten, die aus der Informatik bekannt sein dürften.

Mathematica setzt keine Datentypen voraus. Trotzdem können die Argumente der Funktionen auf bestimmte Typen beschränkt werden, wie wir im Abschnitt 4.3 erfahren werden.

Die bei [KOEPF] so bezeichnete Rememberprogrammierung[23] wird im letzten Abschnitt anhand der Fibonaccizahlen erläutert.

Etwas umfangreicher beschreibt [KAUFMANN] Programmierung mit Mathematica.

4.1 Funktionen programmieren

Im Abschnitt 2.3 wurde bereits darauf hingewiesen, dass Variablen in der Regel global vereinbart werden. Daher lassen wir uns zunächst alle bisher verwendeten Variablen anzeigen:

In[38]:= $?Global`*$

Die Ausgabe aller global vereinbarten Variablen für das Beispiel-Notebook [VORTRAG] sieht so aus:

```
Global`
a  b  c  d  e  f  f1  f2  m1  m2  m3  v1  x  y  z
```

Und diese werden gelöscht mit:

In[39]:= $Remove["Global`*"]$

Selbst definierte Funktionen weisen einen oder mehrere Parameter auf, die (zur Vereinbarung mit einem Unterstrich versehen) in eckigen Klammern angegeben werden. Der Parameter in dieser Funktionsdefinition ist nur noch ein Platzhalter für ein beliebiges Muster. Das folgende Programm demonstriert, wie Funktionen zu vereinbaren sind.

In[40]:= $f[x\_] := x^2;$

In[41]:= $g[x\_, y\_] := \frac{x}{1+y^2}$

[23] Äquivalente Begriffe wie "Funktionen mit Gedächtnis"finden wir in [KOFLER], S. 146 oder "Definitionen mit Gedächtnis"bei [KAUFMANN].

Die Verwendung der Funktionen erfolgt ohne Unterstrich:

In[42]:= $f[t] - g[8, v]$

Die Ersetzung der formalen Parameter erfolgt und wir erhalten als Ergebnis: $t^2 - \frac{8}{v^2+1}$.

Ein Beispiel für ein rekursives Programm ist [KOEPF][24] entnommen worden:

In[43]:= $nextPrime[n\_] := n + 1/; PrimeQ[n + 1]$

In[44]:= $nextPrime[n\_] := nextPrime[n + 1]$

Dieses Programm berechnet die nächste Primzahl oberhalb einer Zahl n. Die Abbruchbedingung ist in der ersten Zeile In[43]:= bei `PrimeQ` zu finden. `PrimeQ` liefert einen booleschen Wert. Falls die nächste Zahl keine Primzahl ist, wird n solange um 1 erhöht bis die Abbruchbedingung der ersten Zeile erfüllt ist. Der Operator `/;` drückt aus, dass es sich um eine bedingte Definition handelt, die nur ausgeführt wird, wenn die folgende Bedingung wahr ist.

Prüfen wir mit der neuen Funktion, welche Primzahl nach 1234567 ermittelt werden kann:

In[45]:= $nextPrime[1234567]$

so erhalten wir einen Wert (1234577), den wir im zweiten Schritt auf Korrektheit überprüfen können.

In[46]:= $PrimeQ[\%]$

Das Ergebnis ist $True$.

4.2 Verzweigungen und Schleifen

Für die Programmierung sind Bedingungen, Rekursionen und Schleifen notwendige Hilfsmittel. Das folgende Programm[25] zeigt die Anwendung einer IF-Verzweigung.

In[47]:= $f[x\_] := If[x > 0, "positiv", "nichtpositiv", "unklar"]$

Das Programm wertet folgenden Aufruf:

In[48]:= $f[-6]$

den Wert -6 zu "nichtpositiv"und

In[49]:= $f[w]$

[24] Vgl. [KOEPF] S. 6
[25] Vgl. [KOEPF], S. 30

zu "unklar"aus.

Eine Schleife[26] sieht wie folgt aus:

In[50]:= $y = 1; Do[y = k * x, \{k, 1, 100\}]; y$

Der sog. Iterator in geschweiften Klammern enthält die Laufvariable k, den Start- und Endwert (hier 1 bis 100)[27], und evtl. die Schrittweite (hier nicht angegeben, da die Schrittweite 1 ist).

Bei genauerer Betrachtung haben wir

In[51]:= 100!

programmiert. Der Vergleich der Ergebnisse[28] beider Ausdrücke bestätigt uns dies.

4.3 Mustererkennung

Die Arbeitsweise von Mathematica beruht auf Mustererkennung. Z. B. gibt es eine eingebaute Regel für:

In[52]:= $Sin[\pi]$

die besagt, dass der Ausdruck zu Null ausgewertet werden kann.

Unverändert bleibt hingegen:

In[53]:= $Sin[x]$

wenn x nicht näher bestimmt ist.

Die Position eines bestimmten Muster kann ermittelt werden mit:

In[54]:= $Position[1 + Sin[3] - Sin[x] + Sin[Pi], Sin[\_]]$

Wir suchen hier "irgendeinen Sinus"`Sin[\_]` in dem Ausdruck (`1+Sin[3]-Sin[x]+Sin[Pi]`). Mathematica kann zunächst eine Ersetzungsregel auf `Sin[Pi]` anwenden. Der verbleibenden Ausdruck sieht in der bereits erwähnten FullForm[29] so aus:

In[55]:= $Plus[1, Sin[3], Times[-1, Sin[x]]]$

Nun ist auch das Ergebnis: {{2},{3,2}} zu interpretieren: Wir finden an zweiter Stelle bei `Sin[3]` und an dritter Stelle als zweiten Teil von `Times[..., Sin[x]]` einen "Sinus von irgendetwas".

[26] Vgl. [KOEPF], S. 31
[27] Wenn der Anfangswert 1 ist, kann er weggelassen werden.
[28] Im Beispiel-Notebook [VORTRAG] nachzuvollziehen.
[29] Zur Erinnerung: Mathematica rechnet intern in dieser FullForm.

Viele mathematische Probleme lassen sich durch die Anwendung fester Regeln lösen. Mit dem nächsten Beispiel "meineDiff"soll die Mustererkennung für die Programmierung der Ableitung einer ganzrationalen Funktion erläutert werden. Es wird sich auf folgende Ableitungsregeln bezogen:[30]

- $(f+g)' = f' + g'$
- $(c*f)' = c*f'$
  falls c eine Konstante ist,
- $(c)' = 0$
  falls c eine Konstante ist,
- $(x^n)' = n*x^{n-1}$

Die Umsetzung der Regeln erfolgt wie bei der Definition von Funktionen. Somit ergibt sich für die erste Regel:

In[56]:= $meineDiff[f\_ + g\_, x\_] := meineDiff[f, x] + meineDiff[g, x]$

Um die zweite Regel zu programmieren, wird der Befehl `FreeQ` benötigt. Dieser überprüft, ob c eine Konstante ist. Dies ist der Fall, wenn c kein x enthält.

In[57]:= $meineDiff[c\_f\_, x\_] := c * meineDiff[f, x] /; FreeQ[c, x]$

Die dritte Regel wird unter der Bedingung formuliert, dass nur die Konstante c ohne x abzuleiten ist.

In[58]:= $meineDiff[c\_, x\_] := 0 /; FreeQ[c, x]$

Möchten wir Funktionen der Art $f(x) = 2x^3+7x^2+1$ nach x ableiten, erhalten wir das Ergebnis $7 * meineDiff[x^2, x] + 2 * meineDiff[x^3, x]$.[31] Uns fehlt jetzt noch ein geeignetes Muster für die 4. Regel:

In[59]:= $meineDiff[x\_^{n\_}, x\_] := n * x^{n-1} /; FreeQ[n, x]$

Und nun erhalten wir mit dem Aufruf unserer Funktion:

In[60]:= $meineDiff[2x^3 + 7x^2 + 1, x]$

die korrekte Ableitung $6x^2 + 14x$.

[30] Vgl. [BURKHARDT], S. 87 ff.
[31] Mathematica ordnet die Terme nach steigenden Potenzen von x.

4.4 Rememberprogrammierung

Die Berechnung der Fibonaccizahlen dienen dem Autor als Grundlage, um die Rememberprogrammierung zu erläutern.

Eine Fibonaccizahl wird bekanntlich aus der Summe ihrer zwei Vorgänger gebildet. Die erste und zweite Zahl stehen per Definition fest. Das rekursive Programm[32] dazu sieht so aus:

In[61]:= $fib[0] = 0;$

In[62]:= $fib[1] = 1;$

In[63]:= $fib[n\_] := fib[n-1] + fib[n-2]$

Die Berechnung von `fib[30]` benötigte in der Vorführung dieses Beispiels 7.75 Sekunden. Die Zeitermittlung erfolgt mit der Mathematica-Funktion: `Timing[fib[30]]`. Der Grund für dieses Verhalten liegt in dem rekursiven Verzweigen für jede Berechnung: Die Berechnung von `fib[30]` benötigt `fib[29]` und `fib[28]`. Um `fib[29]` zu berechnen bedarf es wiederum `fib[28]` und `fib[27]`.[33]

Um eine effiziente Berechnung der Fibonaccizahlen zu erreichen, ist die Verwendung einer zusätzlichen Variable sinnvoll. Der Ausdruck in In[63]:= wird verändert zu
$fib1[n\_] := \underline{fib1[n] =} fib1[n-1] + fib1[n-2]$. Das Programm rechnet nun zwar deutlich schneller, es werden aber alle berechneten Werte von `fib1` gespeichert, wie wir uns mit:

In[64]:= $?fib1$

überzeugen können. Die Ausgabe ist dem Beispiel-Notebook [VORTRAG] zu entnehmen.

[32] Vgl. [KOEPF], S. 36

[33] Entsprechend weiterführen für alle Vorgänger.

5 Zusammenfassung

Mathematica löst eine Vielzahl von mathematischen Problemen. Die Entscheidung, was zu rechnen ist und wie die Ergebnisse zu interpretieren sind, ist Aufgabe des Anwenders. Diese Software ist demnach nur dann ein mächtiges Werkzeug, wenn ausreichende mathematische Grundlagen vorhanden sind.

Die Möglichkeiten von Mathematica sind beeindruckend. Es stellt sich jedoch die Frage, ob andere Computeralgebra-Systeme ähnliche Funktionalitäten aufweisen. Einen Überblick in die Handhabung verschiedener universell einsetzbarer Computeralgebra-Systeme (z. B. Axiom, Maxcyma, Maple, Reduce, Derive und Mathematica) ist in dem Buch von [SCHWARDMANN] zu finden. Eine Gegenüberstellung verschiedener System ist nicht nur aus Kostengründen interessant, sondern auch aufgrund kritischer Stimmen, die der Autor in der betrachteten Literatur gefunden hat. So stellt [KOFLER][34] fest, dass Mathematica zu langsam für numerische Probleme sei. Hier sollte auf Spezialprogramme zurückgegriffen werden. Interessant für Informatiker wäre außerdem ein Blick in die verwendeten Algorithmen der Software. Mathematica gestattet uns dieses aus kommerziellen Gründen nicht.

Der Umfang dieser Arbeit ließ leider weitergehende Recherchen zu Mathematica-Tools nicht zu. Deshalb wird an dieser Stelle nur erwähnt, dass auf Basis von Mathematica webMathematica entwickelt wurde. Damit ist es möglich geworden, interaktives Lehrmaterial mit mathematisch-technischem oder physikalischem Inhalt im Internet zu präsentieren, Lehrveranstaltungen über Internet zu halten oder Vorlesungsskripte verfügbar zu machen. Vielleicht ist das ein Ausblick in die Art des Lernens von morgen!

[34] Vgl. [KOFLER], S. 130

Literatur

[KOEPF] Koepf, Prof. Dr. Wolfram (2006) *Computeralgebra: eine algorithmisch orientierte Einführung*
1. Aufl., Berlin: Springer-Verlag

[KOFLER] Kofler, Michael und Gräbe, Hans Gert (2002) *Mathematica Einführung, Anwendung, Referenz*
4. vollständig überarbeitete und erw. Auflage, München: Addison-Wesley Verlag

[WOLFRAM] Wolfram, Stephen (1997) *Das Mathematica-Buch: Mathematica Version 3*
3. Auflage, Bonn: Addoson Longman Verlag GmbH

[BURKHARDT] Burkhardt, Werner (1996) *Erste Schritte mit Mathematica*
2. überarb. Auflage, Berlin: Springer-Verlag

[SCHWARDMANN] Schwardmann, Ulrich (1995) *Computeralgebra-Systeme: Programme für Mathematik mit dem Combuter*
Bonn: Addiason-Wesley GmbH

[KAUFMANN] Kaufmann, Stephan (1992) *Mathematica als Werkzeug: eine Einführung mit Anwendungsbeispielen*
Berlin: Birkhäuser Verlag Basel

[GERHARDT] LoopBeispiel.pdf aus LM-OR1.pdf

[VORTRAG] PresentationMathematica.nb oder PresentationMathematica.pdf

[WIKIPEDIA] http://de.wikipedia.org/wiki/Primzahl

[MATHEMATICA] http://www.additive-net.de/software/mathematica/mma.player.shtml