Erste Schritte mit MuPAD

Martin Weigt

19. April 2001

Zusammenfassung

Dieser Text gibt eine kleine Einführung in das Computer-Algebra-System *MuPAD*, das im Rahmen der Vorlesungen und Übungen zur Elektrodynamik genutzt werden soll. Er soll bei den ersten Schritten in diesem System helfen, gibt jedoch keinen umfassenden Überblick über *MuPAD*.

1 Was ist MuPAD?

MuPAD ist ein sogenanntes Computer-Algebra-System. Neben numerischen, d.h. Näherungslösungen ermöglichtes es symbolische, d.h. exakte Manipulationen von mathematischen Ausdrücken. Beispiele für das numerische Rechnen reichen von den Funktionen eines gewöhnlichen Taschenrechners über das Plotten von Funktionen und Flächen bis zu komplizierten numerischen Algorithmen. Beispiele für das symbolische Rechnen sind dagegen das Differenzieren und (unbestimmte) Integrieren von Funktionen, das Faktorisieren von Polynomen oder das Auflösen von Gleichungen nach einer Variablen. *MuPAD* enthält eine vollständige Programmiersprache, die nicht nur das einfache Nacheinanderausführen von Befehlen, sondern auch die Nutzung von Schleifen, Verzweigungen usw. bietet. Derartige Computer-Algebra-Systeme werden in den letzten Jahren verstärkt in der theoretisch-physikalischen Forschung eingesetzt, da sie die exakte Analyse von Ausdrücken erlauben, deren Bearbeitung mit Bleistift und Papier hoffnungslos kompliziert erscheint.

Die bekanntesten Computer-Algebra-Systeme sind *Mathematica* und *Maple*. Ihnen gegenüber hat das jüngere und noch weniger komfortable *MuPAD* für uns jedoch einen entscheidenden Vorteil: Es ist frei unter `http://www.mupad.de` verfügbar. Jede und jeder kann unter dieser Adresse eine freie Einzellizenz erhalten. Das System existiert für Linux, Windows und Macintosh. Im Rahmen der Vorlesungen und Übungen zur Elektrodynamik nutzen wir die zur Zeit neueste *MuPAD*-Version 2.0. Sie ist auf den Rechnern `ds9-p00.theorie.physik.uni-goettingen.de` - `ds9-p03.theorie.physik.uni-goettingen.de` verfügbar. Sie können sich von den Linux-PCs und den X-Terminals des Institutes für numerische und angewandte Mathematik aus auf speziell bereitgestellte Accounts einloggen. Der Aufruf von *MuPAD* erfolgt mit dem Befehl `xmupad`.

Diese Einführung ist wie folgt aufgebaut. Zuerst wird die Benutzung der Hilfeseiten beschrieben. Auf diese kann während der Arbeit mit *MuPAD* interaktiv zugegriffen werden, sie enthalten neben den Beschreibungen der einzelnen *MuPAD*-Befehle auch eine ausführliche Einführung in das System, die weit über diesen Text – und auch über die Anforderungen im Rahmen der Vorlesungen und Übungen zur Elektrodynamik – hinausgeht. Diese Hilfefunktion wird insbesondere in der ersten Zeit Ihr wichtigstes Hilfsmittel bei der Nutzung von *MuPAD* sein. Im dritten Abschnitt werden die Unterschiede von exaktem und numerischem Rechnen mit Zahlen anhand einfacher Beispiele verdeutlicht. Im Abschnitt 4 werden wir uns mit Funktionen, dem Differenzieren und Integrieren etc. beschäftigen. Danach folgen Abschnitte über die Plot-Funktion, über lineare Algebra und das Erstellen einfacher Programme.

Diese Einführung ist sehr beispielorientiert und berührt nur den kleinsten Teil der Möglichkeiten von *MuPAD*. Sie soll lediglich eine kleine Starthilfe für die ersten Schritte in die Computeralgebra sein. Ich empfehle daher, die angegebenen Beispiele in *MuPAD* auszuführen und bei Unklarheiten immer die systemeigenen Hilfeseiten zu Rate zu ziehen. Viel Spaß!

2 Die Hilfefunktion

MuPAD wird unter Linux mit dem Befehl `xmupad` aufgerufen, und zunächst öffnet sich das *MuPAD* Session Window. Unter Linux erscheint als Prompt ≫, die Hilfeseiten können durch Eingabe von ? (und `Enter`) bzw. durch Anklicken des Büchleins in der Iconliste geöffnet werden. Kurzinformationen zu einzelnen Befehlen (hier z.B. zu `solve`) koennen jederzeit durch Eingabe von

≫ `info(solve)`

abgerufen werden, hier erscheint als Antwort:

```
solve -- solve equations and inequalities [try ?solve for options]
```

Befehle werden grundsätzlich durch Drücken der `Enter`-Taste ausgeführt. Zur übersichtlichen Darstellung kann es nötig sein, innnerhalb eines Kommandos oder eines Programmes einen neue Zeile zu beginnen ohne den Befehl auszuführen. Dieser Zeilenumbruch erfolgt mit `Shift-Enter`. Ausführliche Informationen erhält man durch Eingabe von

≫ `?solve`

In diesem Fall wird die entsprechende Handbuchseite im Hilfefenster geöffnet. Diese ist als Hypertext angelegt, enthält also Links zu ähnlichen Kommandos, und illustriert die Nutzung des erfragten Befehls durch mehrere Beispiele. Der Nachteil dieser beiden Funktionen ist jedoch, dass man den genauen Namen des Befehls vorher wissen muss.

Sehr hilfreich beim Erlernen ist daher das mitgelieferte Tutorium, das durch Eingabe von

≫ `?tutorium`

geöffnet wird. Es enthält auch ein umfangreiches Stichwortverzeichnis, das die Suche nach unbekannten Befehlen erleichtert. Tutorium und Stichwortverzeichnis sind ebenfalls als Hypertext realisiert.

3 Exaktes und numerisches Rechnen mit Zahlen

MuPAD kann wie ein besserer Taschenrechner benutzt werden:

≫ `1+5/3`

```
8/3
```

Wie man an diesem Beispiel sieht, werden die Ergebnisse als exakte rationale Zahl angegeben, nicht in Gleitkommadarstellung. Letztere stellt nur eine Rundung da, die mittels des Befehls `float` gefunden werden kann:

≫ `float(1+5/3)`

```
2.666666667
```

MuPAD rechnet automatisch in Gleitkommadarstellung, wenn ein Teil der Eingaben schon in Gleitkommadarstellung erfolgt:

≫ `1.0+5/3`

```
2.666666667
```

Noch deutlicher wird der Unterschied z.B. bei der Lösung einer quadratischen Gleichung $x^2 - 7x + 1 = 0$. Der Befehl

≫ `solve(x^2 - 7*x + 1 = 0, x)`

gibt die exakten Lösungen $7/2 \pm 3/2 \cdot \sqrt{5}$ an, während ein nachfolgendes

≫ `float(%)`

```
0.1458980338, 6.854101966
```

die Näherungslösung in Gleitkommadarstellung ermittelt. Das sehr nützliche Symbol % bezeichnet die Ausgabe des jeweils letzten Befehls, d.h. `float(%)` berechnet die Gleitkommadarstellung der letzten Ausgabe. Bei Anwendung von `float(solve)` auf nichtpolynomiale Gleichungen ist zu beachten, dass nur eine Lösung ausgegeben wird. Allerdings kann man den Suchbereich einschränken, siehe `?numeric::solve` oder Kapitel 8 im Tutorium.

Für die größtmögliche Genauigkeit der Ergebnisse ist es sinnvoll, so lange wie möglich mit den exakten Ausdrücken zu rechnen. Die numerischen Berechnungen im Gleitkommaformat sind jedoch schneller, so dass in praktischen Anwendungen meist eine geschickte Mischung beider am günstigsten ist.

MuPAD kennt die meisten Standardfunktionen und kann mit komplexen Zahlen umgehen:

```
≫ sqrt(-1)

                                    I

≫ exp(I*PI)

                                    -1

≫ sin(3)

                                  sin(3)
```

Hierbei steht `PI` für $\pi \simeq 3.14159...$, analog `E` für $e = \exp(1)$. Das dritte Beispiel zeigt insbesondere, dass die Funktionen ebenfalls exakt ausgewertet werden – für $\sin(3)$ gibt es keine einfachere Darstellung. Eine Näherung kann wieder mit `float` berechnet werden.

4 Rechnen mit Funktionen

MuPAD ist in der Lage, Formeln mit (mehreren) Variablen zu verarbeiten. So kann man Funktionen definieren, hier z.B. $f(x) = 2x^2 + \sin(x)$:

```
≫ f := x -> 2*x*x + sin(x)

                            x -> 2*x*x + sin(x)
```

Zuweisungen erfolgen immer mit `:=` (`a:=2` ordnet a den Wert 2 zu), f wird also als Abbildung $x \mapsto 2x^2 + \sin(x)$ eingeführt. Diese Funktion kann bei beliebigen Werten von x ausgewertet werden, z.B.

```
≫ f(0.9)

                                2.40332691
```

oder auch nach x abgeleitet werden:

```
≫ diff(f(x), x)

                               4 x + cos(x)
```

Die zweite Ableitung kann auf verschieden Wegen berechnet werden:

```
≫ diff(%, x); diff(f(x), x, x); diff(f(x), x$2)

                                4 - sin(x)
                                4 - sin(x)
                                4 - sin(x)
```

Das Semikolon trennt zwei Befehle in einer Eingabe, sie werden nacheinander ausgeführt. *MuPAD* kann sowohl unbestimmte als auch bestimmte Integrale berechnen, die entsprechenden Befehle sind z.B. `int(f(x),x)` für $\int f(x)\ dx$, und `int(f(x),x=0..PI)` für $\int_0^\pi f(x)\ dx$.

Völlig analog werden mehrdimensionale Funktionen behandelt:

$\gg$ `g := (x,y) -> x^2*y^2`

```
(x,y) -> x^2*y^2
```

definiert die Funktion $g(x,y) = x^2y^2$. Integration und partielle Ableitungen erfolgen mit denselben Befehlen, z.B.

$\gg$ `diff( g(x,y), x$2, y)`

```
4 y
```

berechnet $\frac{\partial^3 g(x,y)}{\partial x^2 \partial y}$.

MuPAD kann jedoch auch deutlich kompliziertere Manipulationen vornehmen, wie z.B. das Lösen von Differentialgleichungen. Diese Darstellung geht jedoch über den Rahmen dieser Kurzeinführung hinaus. Ich verweise erneut auf das wesentlich umfassende Tutorium, das von den Hilfeseiten von *MuPAD* aus geöffnet werden kann.

5 Plotten

Um Vorstellungen vom Verlauf einer Kurve zu bekommen, ist es wichtig, sie graphisch darzustellen. *MuPAD* verfügt über ein sehr mächtiges, aber auch kompliziertes Instrumentarium zum Anfertigen von Plots. Die einfachsten und wohl meistgenutzten Befehle ergeben 2- und 3-dimensionale Funktions-Plots:

$\gg$ `plotfunc2d(f(x), x=-2..2)`

$\gg$ `plotfunc3d(g(x,y), x=-2..2, y=-3..3)`

stellen $f(x)$ und $g(x,y)$ über den jeweiligen Intervallen dar. Man kann auch parametrische Plots anfertigen:

$\gg$ `plot2d(Scaling = Constrained, [Mode = Curve, [cos(u), sin(u)], u = [0, 2*PI]])`

stellt z.B. den Einheitskreis dar. Die Option `Scaling = Constrained` garantiert, dass der Kreis nicht zu einer Ellipse verzerrt wird. Näheres ist wieder unter `?plot2d` zu finden. Es gibt eine analoge Funktion `plot3d`, die dreidimensionale parametrische Plots anfertigt. Der mächtigste Befehl (bzw. eine Bibliothek von Befehlen) allerdings ist `plot`, `info(plot)` gibt eine Liste der Unterbefehle. Diese ermöglichen z.B. die Darstellung von Vektorfeldern.

6 Vektoren und Matrizen

Den einfachsten Weg zur Erzeugung von Matrizen bietet der Befehl `matrix(m,n,[[a11,a12,...a1n],[a21,...a2n],... [am1,...amn]])`, der die $m \times n$-Matrix mit den Einträgen $a_{ij},\ i = 1,...,m,\ j = 1,...,n$, definiert. m und n können in diesem Befehl auch weggelassen werden, die Einträge können sowohl (exakte) rationale Zahlen, Zahlen in Gleitkommadarstellung als auch arithmetische Ausdrücke wie die oben genannten Funktionen sein. Entsprechend werden n-dimensionale Spaltenvektoren durch `matrix(n,1,[a1,...,an]`, Zeilenvektoren durch `matrix(1,n,[a1,...,an]` gegeben. Haben wir zwei Matrizen A und B gegeben, z.B. als

$\gg$ `A := matrix([[2,1], [4,9]]); B := matrix([[2*x,1], [4/7,9*sin(x)]])`

dann ergeben die Kommandos

$\gg$ `A*B; A^3; 1/A; A^(-1)`

das Matrixprodukt $A \cdot B$, die dritte Potenz A^3 und die inverse Matrix A^{-1}. Einzelne Einträge der Matrix A können über `A[i,j]` ausgegeben werden:

$\gg$ `A[1,2]; B[2,2]`

```
1
9*sin(x)
```

Die wichtigsten Befehle der linearen Algebra sind in der Bibliothek `linalg` zusammengefasst, eine Liste wird durch `info(linalg)` ausgegeben. Darin finden wir insbesondere `linalg::charpoly`, `linalg::det`, `linalg::eigenvalues`, `linalg::eigenvectors`, `linalg::tr`, `linalg::transpose` für das charakteristische Polynom, die Determinante, Eigenwerte und -vektoren, die Spur und die Transponierte. Das Kommando `export(linalg)` exportiert die Funktionen der Bibliothek `linalg`, diese können dann ohne den Zusatz `linalg::` aufgerufen werden, also reicht `det(A)` anstelle von `linalg::det(A)` für die Berechnung der Determinante von A aus.

Für die Elektrodynamik sind partielle Differentialoperatoren von zentraler Bedeutung, hierzu gehören Rotation, Gradient und Divergenz. Diese sind ebenfalls in der Bibliothek `linalg` enthalten und können auf Vektoren/Skalare angewendet werden, deren Einträge mehrdimensionale Funktionen sind.

7 Einfache Programme

Mit Hilfe von *MuPAD* können auch Programme geschrieben werden. Eines der wichtigsten Instrumente dafür sind Schleifen:

```
≫ for i from 1 to 7 step 3 do print(i,i*i) end_for

1, 1
4, 16
7, 49
```

```
≫ s:= 1; repeat s:=s+2; print(s, float(int(sin(x)/x,x=0..s))) until s>4 end_repeat

1
3, 1.848652528
5, 1.549931245
```

Ähnlich funktioniert die `while`-Schleife, Verzweigungen im Programm sind mit `if-then-else` und `case` möglich. Unterprogramme (Prozeduren) werden mit `proc` definiert, genaueres ist den Hilfeseiten zu entnehmen. Ich möchte noch einmal daran erinnern, dass Zeilen innerhalb eines Programmes mit Shift-Enter umgebrochen werden.

8 Ausblick

Diese Kurzeinfassung hat einen kurzen Einblick in das Computer-Algebra-System *MuPAD* gegeben. Wie wir schon am Beispiel von `linalg` und `plot` gesehen haben, sind viele der komplizierteren Funktionen von *MuPAD* in Bibliotheken eingeordnet. Einige der wichtigsten sind

`combinat`	Kombinatorik
`detools`	Differentialgleichungen
`intlib`	Integration
`numeric`	numerische Algorithmen
`solvelib`	Lösen von Gleichungen
`transform`	Integraltransformationen

Wie üblich lassen sich die wichtigsten Informationen durch Eingabe von `info(Bibliotheksname)` oder `?Bibliotheksname` finden.

Zum tieferen Einblick in *MuPAD* empfehle ich das schon mehrfach genannte Tutorium, bei kleineren Unsicherheiten die Hilfeseiten zu benutzen.