

Einführung in Maple

Version 0.2

Tobias Müller

Ammerbuch, den 25. April 2005

Inhaltsverzeichnis

1	Einfaches Rechnen mit Maple	3
1.1	Grundlagen	3
1.2	Einfaches Rechnen mit Maple	3
1.3	Variablen	4
1.4	Variablen einschränken	4
1.5	Funktionen	5
1.6	Spezielle Funktionen	5
1.7	Ausdrücke vereinfachen	5
2	Zeichnen mit Maple	6
2.1	Einfache Funktionen zeichnen	6
2.2	Parametrische Darstellung	7
2.3	3-dimensionale Darstellungen	8
2.4	Vektorfelder visualisieren	9
3	Analysis	10
3.1	Differenzieren und Integrieren	10
3.2	Grenzwerte	10
3.3	Stückweise definierte Funktionen	10
4	Lineare Algebra	12
4.1	Vektoren und Matrizen	12
4.2	Addition und skalare Multiplikation	12
4.3	Skalarprodukt	12
4.4	Kreuzprodukt	13
4.5	Matrizenmultiplikation	13
4.6	Inverse einer Matrix	13
5	Statistik	14
5.1	Listen	14
5.2	Mittelwert	14
6	Typische Probleme/Fehler	15
6.1	Groß/Kleinschreibung	15
6.2	Verwendung des %-Operators	15
6.3	Seltsame Ergebnisse mit I	15

1 Einfaches Rechnen mit Maple

1.1 Grundlagen

Um mit Maple rechnen zu können, ist es wichtig ein paar Grundlagen über Maple zu wissen. In Maple werden alle Befehle in sogenannten Execution Groups (welcher an einer schwarzen eckigen Klammer erkannt werden können) eingegeben. Dort erscheinen sie in roter Farbe. Wenn Maple das Arbeitsblatt dann ausführt, werden die Ergebnisse der Befehle in der Execution Group dann am Ende der Execution Group in blau dargestellt. Alle Befehle müssen mit einem `;` abgeschlossen werden. Falls die Ausgabe der Befehle unterdrückt werden soll, muss ein `:` anstatt des `;` verwendet werden.

```
> 1+2*3;  
4*4;  
23-23;  
  
7  
0
```

Innerhalb dieser Execution Groups können mit `#` Kommentare eingegeben werden. Mit dem Bestätigen der Eingabetaste wird die komplette Execution Groups ausgewertet. Mit Umschalt+Eingabetaste kann eine neue Zeile innerhalb der Execution Group erzeugt werden. Einige Befehle sind in extra Paketen abgelegt. Diese können mit `with` nachgeladen werden.

1.2 Einfaches Rechnen mit Maple

Einfach aritmetische Operationen können mit Maple in einfacher Weise getätigt werden. Dazu wird einfach der zu berechnende Ausdruck in einer Execution Group eingegeben. Maple beachtet dabei die Potenz-vor-Punkt-vor-Strichregel und interpretiert Klammern `()` korrekt. Die Operatoren `+`, `-`, `*`, `/` können direkt eingegeben werden. Eine Potenz wird mit `^` oder `**` und eine (Quadrat-)Wurzel mit `sqrt` dargestellt. Der `*`-Operator darf allerdings nie weggelassen werden: `3x` oder `3 x` ist also kein gültiger Ausdruck!

```
> ((1+2*3)**2)/7-2;  
  
5
```

Maple rechnet dabei standardmäßig exakt. Ist eine Fließkomma-Berechnung erwünscht muss das mit `evalf` erwungen werden. Kommt allerdings eine Fließkommazahl (z.B. `1.5`) in einem Ausdruck vor, wird er automatisch als Fließkomma-Ausdruck ausgewertet. `evalf` kann noch ein zweiter Parameter zur Festlegen der Nachkommastellen übergeben werden.

```
> sqrt(2);  
3/2;  
3.0/2;  
sin(Pi/4);  
evalf(Pi,25);  
  
√2  
3  
2  
1.500000000  
1/2√2  
3.141592653589793238462643
```

Um mit einem vorherigen Ergebnis weiterzurechnen gibt es den `%`-Operator. Analog dazu gibt es die `%%`- und `%%%`-Operatoren um mit dem zweit bzw. drittletzten Ergebnisses weiterzurechnen. (Vgl. [Abschnitt 6.2](#))

```
> 1*2;
3+4;
%%;
%%;
%;

2
7
2
7
7
```

1.3 Variablen

Natürlich unterstützt Maple auch Variablen. Dabei ist immer zwischen gebundenen und ungebundenen Variablen zu unterscheiden. Eine Variable wird durch eine Zuweisung gebunden. Immer wenn die Variable nun vorkommt, ersetzt Maple sie durch den Ausdruck/Wert an den sie gebunden wurden. Das ist zum Beispiel sehr praktisch um einen Wert für später zwischenspeichern. Ungebundene Variablen werden immer dann gebraucht, wenn ein Wert noch unbekannt ist, also zum Beispiel in einer Gleichung mit unbekanntem oder in eine Funktionsvorschrift. Variablen werden können mit `:=` zuweisen werden. Um die Zuweisung wieder aufzugeben muss einfach der Name der Variablen in `'` der Variable zugewiesen werden.

```
> a:=5;
b:=10;
a*b;
a+c;
5*x+10*a;
a:='a';
5*x+10*a;

a := 5
b := 10
50
5 + c
5x + 50
a := a
5x + 10a
```

1.4 Variablen einschränken

Oft ist wichtig Variablen einzuschränken. Dazu dient der Befehl `assume`. Eine mit ihm beschränkte Variable taucht dann bei der Ausgabe immer gefolgt von einer `0` auf, damit man weiss das sie beschränkt ist. `assume` wird immer der Variablenname und die Einschränkung übergeben, wie z.B. $x < 5$. Alternativ sind kann man eine Variable z.B. auch auf ganze Zahlen (`integer`) einschränken.

```
> assume(s, integer);
sin(s*Pi);
assume(s<0);
assume(r>0);
is(s*r<0);

0
true
```

Falls man keine Kennzeichnung der eingeschränkten Variablen möchte, kann man dies mit `interface (showassumed = 0);` abschalten.

1.5 Funktionen

Um Funktionen in Maple zu definieren wird eine Funktionsvorschrift z.B. `f: x -> x**2` einer Variablen zugewiesen. Auf die Funktion kann dann mit `f(x)` zugegriffen werden, bzw. mit `f(2)` konkrete Werte berechnet werden.

```
> f:=x->x**2;
f(2);
f(x);
```

$$f := x \rightarrow x^2$$

1.6 Spezielle Funktionen

Natürlich sind in Maple schon sehr viele Funktionen vordefiniert. Eine kleine Auswahl der wichtigeren soll hier präsentiert werden:

Maplebefehl	Funktionsname	Mathematische Schreibweise
<code>abs</code>	Betragsfunktion	$ x $
<code>sqrt</code>	Quadratwurzel	\sqrt{x}
<code>sin</code>	Sinus	$\sin x$
<code>cos</code>	Kosinus	$\cos x$
<code>tan</code>	Tangens	$\tan x$
<code>arcsin</code>	Arkussinus	$\arcsin x = \sin^{-1} x$
<code>arccos</code>	Arkuskosinus	$\arccos x = \cos^{-1} x$
<code>arctan</code>	Arkustangens	$\arctan x = \tan^{-1} x$
<code>exp</code>	Exponentialfunktion	$e^x = \exp x$
<code>ln</code>	natürlich Logarithmus	$\ln x$
<code>log10</code>	Logarithmus zur Basis 10	$\log_1 0x$

Die hyperbolischen trigonometrischen Funktionen und deren Umkehrfunktionen sind auch definiert. Alle trigonometrischen Funktionen erwarten (wie üblich) ihre Parameter im Bogenmaß.

1.7 Ausdrücke vereinfachen

Um algebraische Ausdrücke zu vereinfachen stellt Maple viele Befehle zur Verfügung. Die zwei wichtigsten seien hier vorgestellt: `simplify` ist wohl der wichtigste Befehl um Ausdrücke zu vereinfachen. Hierbei versucht Maple Vereinfachungsregeln anzuwenden, Funktionsaufrufe (wenn möglich) zu erledigen, etc. Beim anderen Befehl `factor` wird versucht Polynome möglichst faktorisiert darzustellen.

```
> simplify((x+2)*(x-2)*(x+3)*(x-3));
factor(%);
```

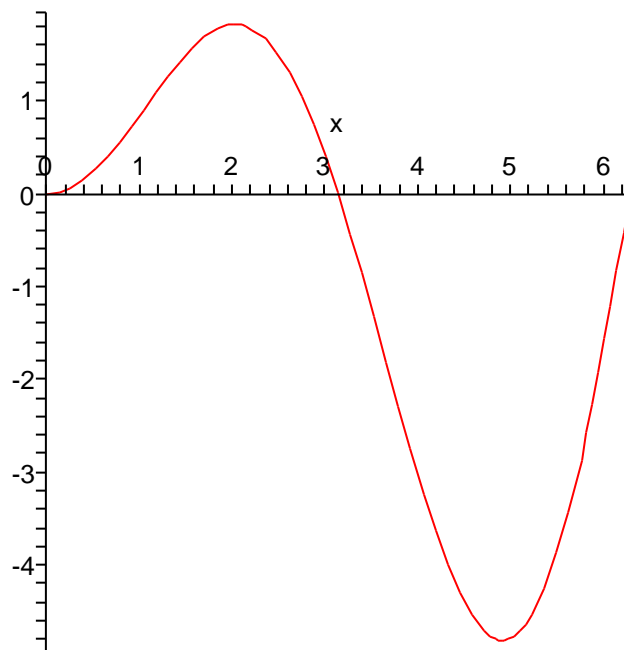
$$(x^2 - 4)(x^2 - 9)$$
$$(x + 2)(x - 2)(x + 3)(x - 3)$$

2 Zeichnen mit Maple

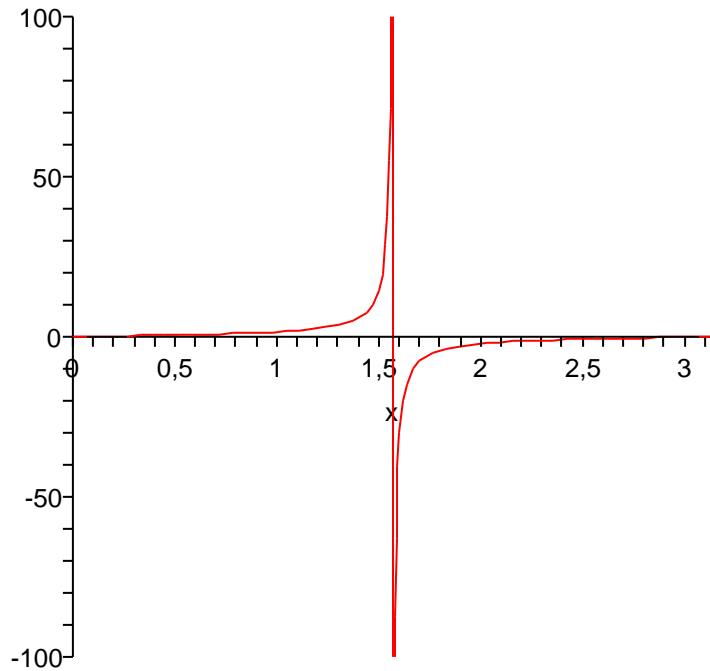
2.1 Einfache Funktionen zeichnen

Die einfachste Art Funktionen darzustellen ist der 2 dimensionale Plot. Dabei wird einfach der Laufparameter auf der x -Achse und der Funktionswert auf der y -Achse dargestellt. Dem `plot`-Befehl wird dazu einfach die zu zeichnende Funktion und der Funktionsparameter übergeben. Falls gewünscht, kann der Funktionsparameter durch ein Intervall eingeschränkt werden. Ist eine qualitativer Funktionsverlauf gewünscht darf das Intervall auf `infinity` (∞) als Grenze beinhalten. Wenn eine Beschränkung der y -Achse durch ein Intervall gewünscht ist, kann dies als dritter Parameter übergeben werden. Alle nachfolgenden Parameter dienen dann nur noch der Darstellung. Die wichtigsten sind `scaling=CONSTRAINED` für eine gleiche Einteilung der x und y -Achse sorgt, `style=` um zwischen Linien- (`line`) oder Punktdarstellung (`point`) umzustellen.

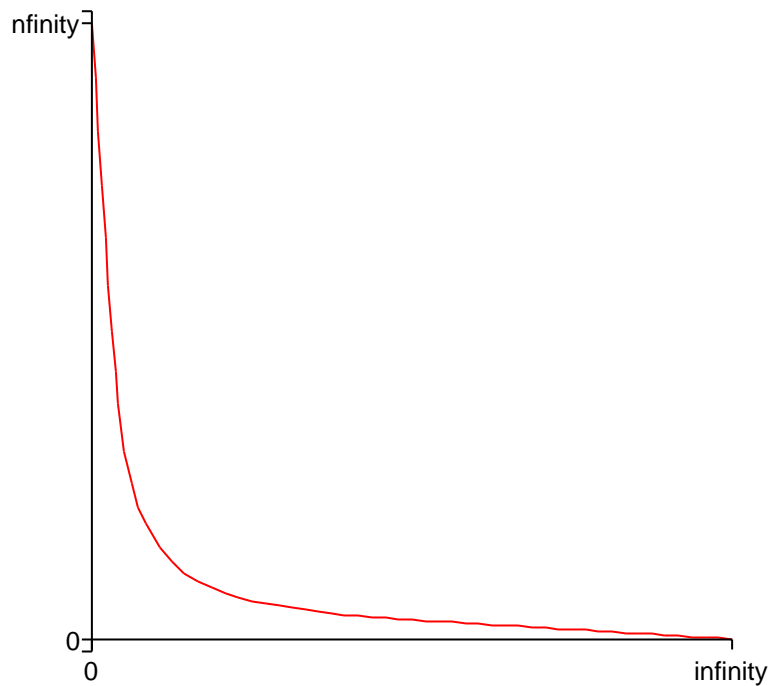
```
> plot(x*sin(x),x=0..2*Pi,scaling=CONSTRAINED);
```



```
> plot(tan(x),x=0..Pi,y=-100..100);
```



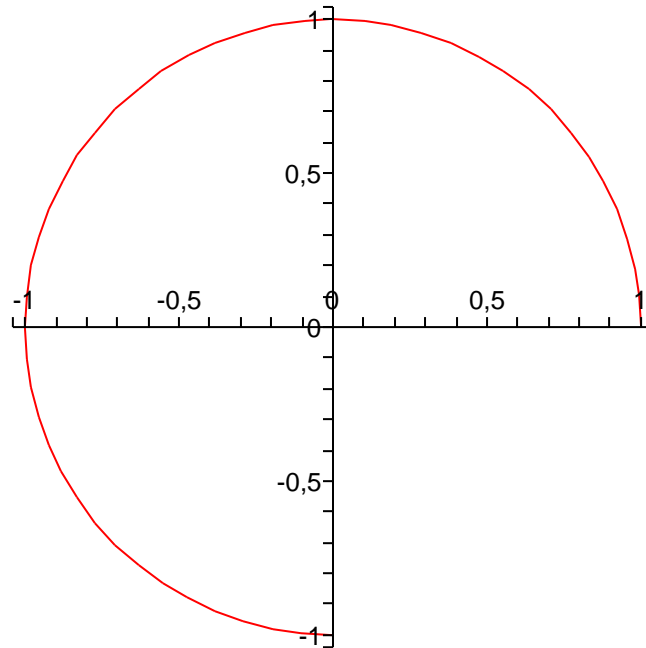
```
> plot(1/x,x=0..infinity);
```



2.2 Parametrische Darstellung

Oft wird auch eine parametrische Darstellung gewünscht. Dabei werden 2 Funktionen dem `plot`-Befehl übergeben. Die erste gibt die y -Koordinate und die zweite die x -Koordinate an. Der Laufparameter wird hier nicht mehr sichtbar. Die „normale“ Darstellung ist aber eigentlich nur ein Spezialfall der parametrische Darstellung, bei dem die zweite Funktion einfach $f(x) = x$ ist. Damit Maple weißt, das es sich um eine parametrische Darstellung handelt, wird ein Array (z.B. `[[cos(x),sin(x),x=0..Pi]]`) aus beiden Funktionen und Laufparameter an `plot` übergeben. Danach können die selben weiteren Parameter wie in [Abschnitt 2.1](#) übergeben werden.

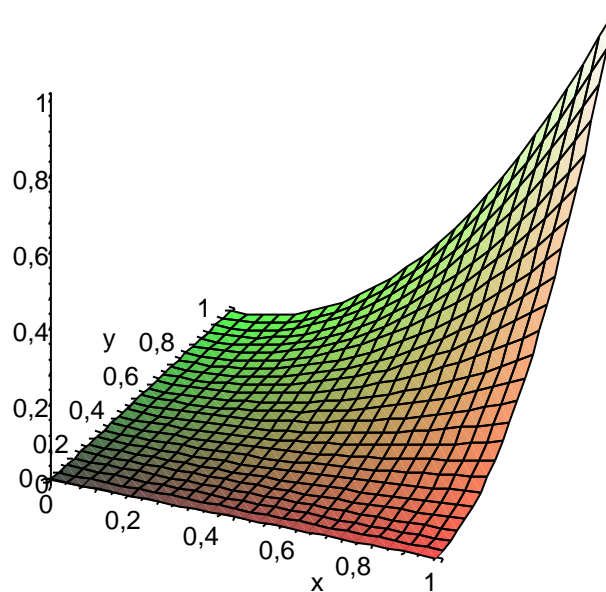
```
> plot([cos(x),sin(x),x=0..3*Pi/2],scaling=CONSTRAINED);
```



2.3 3-dimensionale Darstellungen

Hat eine Funktionen mehrere Parameter, hängt also z.B. von x und y ab, so bietet sich oft eine 3-dimensionale Darstellung an. Dabei werden x und y Parameter in einer Ebene dargestellt und der Funktionswert dann als Höhe in z -Richtung interpretiert. Dazu wird der `plot3d`-Befehl verwendet der als ersten Parameter die Funktion und als die nächsten 2 Parameter die Funktionsparameter erwartet. Anschließend können weitere Optionen angegeben werden.

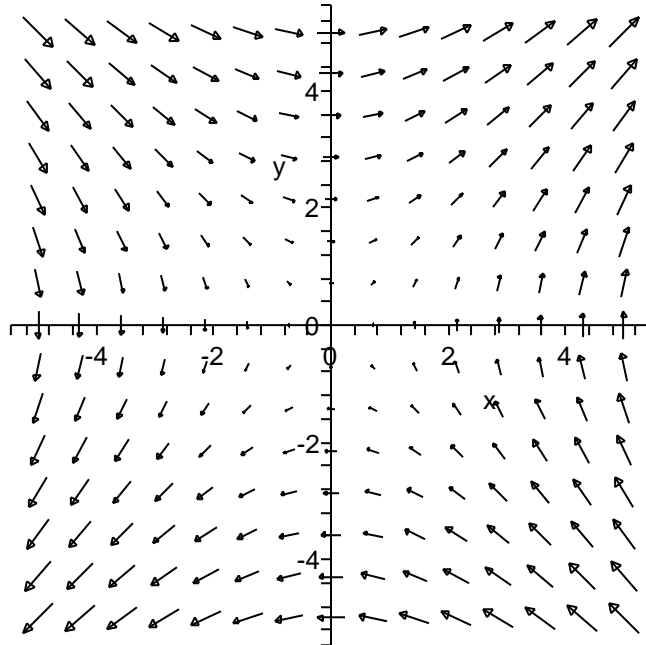
```
> plot3d(x**2*y**2,x=0..1,y=0..1,axes=NORMAL);
```



2.4 Vektorfelder visualisieren

Ist das `plots`-Paket geladen, gibt es außerdem den Befehl `fieldplot` mit dem Vektorfelder schön visualisiert werden können. Außerdem einem (2-dimensionalen) Vektor erwartet der Befehl die Intervalle der beiden Laufvariablen.

```
> fieldplot( [y,x], x=-5..5, y=-5..5, arrows=SLIM,grid=[15,15]);
```



3 Analysis

3.1 Differenzieren und Integrieren

Maple kann Funktionen auch differenzieren (ableiten) und integrieren. Das Differenzieren erfolgt mit dem `diff`-Befehl, welcher als Parameter die zudifferenzierende Funktion und die Variablen, nach denen differenziert werden soll, erwartet.

```
> diff(x**2,x);
diff(x**3,x,x);
```

$$2x$$

$$6x$$

Beim Integrieren muss zwischen einem bestimmten Integral (mit Grenzen) oder einem unbestimmten Integral (Stammfunktion) unterschieden werden. Beides mal wird der `int`-Befehl verwendet, der als ersten Parameter die zu integrierende Funktion und als zweiten Parameter die Integrationsvariable erwartet. Möchte man ein bestimmtes Integral berechnen muss die Integrationsvariable mit einem Intervall übergeben werden.

```
> int(sin(x),x);
int(cos(x),x=0..Pi/2);
```

$$-\cos(x)$$

$$1$$

3.2 Grenzwerte

Um Grenzwerte mit Maple zu berechnen, gibt es den `limit`-Befehl. Er benötigt als Parameter eine Funktion und eine Stelle, an welcher der Grenzwert gebildet werden soll. Ist die Richtung des Grenzwertes wichtig, kann diese als optionale Parameter übergeben werden, wobei `left`, `right`, `real` und `complex` mögliche Richtungen sind.

```
> limit(1/x,x=0);
limit(1/x,x=0,left);
limit(1/x,x=0,right);
limit(1/x,x=infinity);
```

$$undefined$$

$$-\infty$$

$$\infty$$

$$0$$

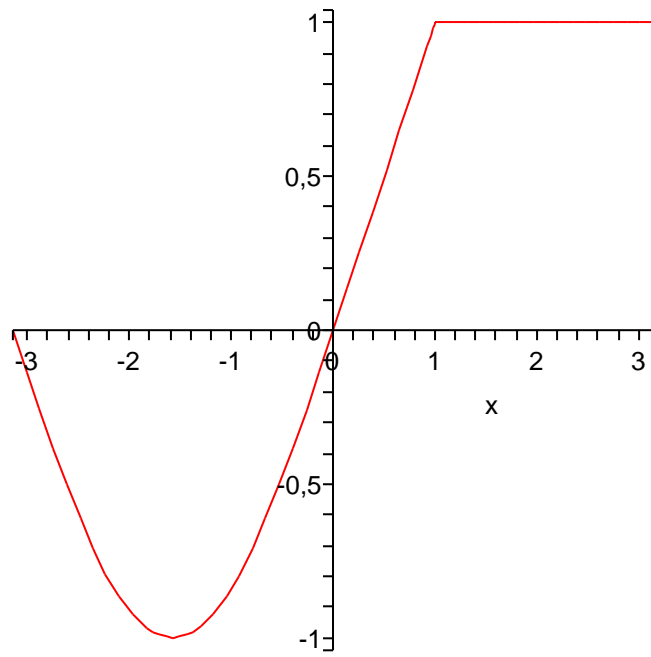
3.3 Stückweise definierte Funktionen

Maple kann Funktionen auch stückweise definieren. Dazu wird der `piecewise`-Befehl verwendet. Ihm könnt immer eine Funktion und eine Bedingung übergeben als Parameter übergeben werden. Der letzte Parameter gibt (ohne Bedingung) an, was für den Rest geschehen soll.

```
> piecewise(x<0,sin(x),x<1 and x>=0,x,1);
```

$$\begin{cases} \sin(x) & x < 0 \\ x & x < 1 \text{ and } 0 \leq x \\ 1 & \text{otherwise} \end{cases}$$

```
> plot(piecewise(x<0,sin(x),x<1 and x>=0,x,1),x=-Pi..Pi);
```



4 Lineare Algebra

Um mit Maple lineare Algebra betreiben zu können, muss meist das Paket `linalg` mit `with(linalg)` geladen werden.

4.1 Vektoren und Matrizen

Vektoren in Maple mit `<>` erzeugt. Dabei werden die Vektorelemente einfach durch Kommata getrennt. Die Matrizen werden ähnlich erzeugt. Dabei wird einfach ein „Vektor“ erzeugt, dessen Elemente aus waagrechten Vektoren bestehen, die analog zu Vektoren erzeugt werden, wobei allerdings ein `|` anstatt der Kommata verwendet wird.

```
> v:=<x,y,z>;
m:=< <m11|m12|m13>, <m21|m22|m23>, <m31|m32|m33> >;
```

$$v := \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$m := \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}$$

4.2 Addition und skalare Multiplikation

Die Addition (Subtraktion) und skalare Multiplikation (Division) von Vektoren mit den selben Befehlen `+`, `-`, `*` und `/` erreicht werden.

```
> a:=<1,2,3>;
b:=<9,8,7>;
a+b;
2*a;
```

$$a := \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

$$b := \begin{bmatrix} 9 \\ 8 \\ 6 \end{bmatrix}$$

$$\begin{bmatrix} 10 \\ 10 \\ 10 \end{bmatrix}$$

$$\begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix}$$

4.3 Skalarprodukt

Für das Skalarprodukt gibt es einen Befehl `dotprod`, falls das Pakete `linalg` geladen wurden. Ihm werden einfach 2 Vektoren als Parameter übergeben.

```
> dotprod(<1,2,3>,<9,8,6>);
```

4.4 Kreuzprodukt

Analog zum Skalarprodukt ist auch das Kreuzprodukt mit `crossprod` zu benutzen. Hier für ist ebenfalls das Paket `linalg` von Nöten.

```
> crossprod(<1,2,3>, <9,8,6>);
```

$[-12, 21, -10]$

4.5 Matrizenmultiplikation

Ist das `linalg`-Paket geladen, kennt Maple den Befehl `multiply` zur Matrixmultiplikation. Ihm werden einfach 2 Matrizen als Parameter übergeben. Zu beachten ist hier natürlich das die Matrizenmultiplikation nicht kommutativ ist!

```
> multiply(<<1|0|0>, <0|1|0>, <0|0|1>, <1,2,3>);  
multiply(<<9|8|7>, <6|5|4>, <3|2|1>, <1|2|3>, <4|5|6>, <7|8|9>);  
multiply(<<1|2|3>, <4|5|6>, <7|8|9>, <9|8|7>, <6|5|4>, <3|2|1>);
```

$[1, 2, 3]$

$$\begin{bmatrix} 90 & 114 & 138 \\ 54 & 69 & 84 \\ 18 & 24 & 30 \end{bmatrix}$$
$$\begin{bmatrix} 30 & 24 & 18 \\ 84 & 69 & 54 \\ 138 & 114 & 90 \end{bmatrix}$$

4.6 Inverse einer Matrix

Maple kann (wenn möglich) auch die Inverse eine Matrix mit `inverse` berechnen

```
> inverse(<<1|1|1>, <0|1|1>, <0|0|1>);
```

$$\begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}$$

5 Statistik

Um mit Maple Statistik betreiben zu können, muss meist das Paket `stats` mit `with(stats)` geladen werden. Für viele Funktionen wird zusätzlich das Unterpaket `describe` benötigt, welches mit `with(stats[describe])` geladen werden kann.

5.1 Listen

In der Statistik ist meist eine große Menge von Daten handzuhaben. Da es auf eine Reihenfolge der Daten nicht angekommen werden sie in Liste gespeichert. Eine Liste wird in Maple als eine durch Kommata und `[]` eingerahmte Auflistung dargestellt.

```
> l := [1, 2, 2, 2, 7, -5];
```

```
l := [1, 2, 2, 2, 7, -5];
```

5.2 Mittelwert

Ist das Paket `describe` geladen, gibt es verschiedene Befehl um den Mittelwert einer Liste zuberechnen. `mean` berechnet das aritmetische Mittel, `geometricmean` das geometrische Mittel, `harmonicmean` das harmonische Mittel und `quadraticmean` das quadratische Mittel.

```
> l := [1, 2, 2, 2, 7, -5]:  
mean(l);  
geometricmean(l);  
harmonicmean(l);  
quadraticmean(l);
```

```

$$\frac{3}{2}$$

$$(-280)^{(1/6)}$$

$$\frac{140}{57}$$

$$\frac{1}{2}\sqrt{58}$$

```

6 Typische Probleme/Fehler

6.1 Groß/Kleinschreibung

Maple unterscheidet zwischen Groß/Kleinschreibung. Bei vordefinierten Funktionen gibt es oft Unterschiede in der Funktion. Der Befehl `int` berechnet zum Beispiel ein Integral, während `Int` nur das Integral anzeigt, aber nicht berechnet. Oft ist aber auch nur eine Version definiert und man wundert sich warum Maple `Simplify` nicht mehr kennt.

```
> Int(sin(x)*exp(x), x);
int(sin(x)*exp(x), x);
```

$$\int \sin(x)e^x dx$$

$$-\frac{1}{2} \cos(x)e^x + \frac{1}{2} \sin(x)e^x$$

Vor allem beim selbst definierten Variablen (vgl. [Abschnitt 1.3](#)) ist Vorsicht geboten, da man hier evtl. völlig andere Sachen gespeichert hat.

```
> f:= x -> x**2;
F:= x -> sqrt(x);
f(2);
F(2);
```

$$f := x \rightarrow x^2$$

$$F := x \rightarrow \sqrt{x}$$

$$4$$

$$\sqrt{2}$$

6.2 Verwendung des %-Operators

Bei der Verwendung des %-Operators (bzw. der %- und %%-Operatoren) ist zu beachten, dass Maple immer auf die zuletzt berechneten, nicht die vorherigen Ergebnisse im Dokument zurückgreift. Daher ist es allgemein hilfreich, ein Maple Dokument immer mit dem `restart`-Befehl zu beginnen und dann immer das komplette Arbeitsblatt mit dem `!!!`-Symbol neu rechnen zu lassen, weil dann immer derselbe Ablauf gewährleistet ist.

6.3 Seltsame Ergebnisse mit I

Lässt man Maple zum Beispiel die Quadratische Gleichung $x^2 + 4x + 100 = 0$ lösen, gibt Maple nur 2 „seltsame“ Ergebnisse mit `I` zurück.

```
> solve(x**2+4*x+100=0, x);
```

$$-2 + 4I\sqrt{6}, -2 - 4I\sqrt{6}$$

Das hängt damit zusammen, dass diese Gleichung keine Lösungen in \mathbb{R} hat. Maple rechnet aber immer \mathbb{C} (komplexe Zahlen) und liefert daher auch ein Ergebnis, welches allerdings komplex ist. Es gilt $I^2 = -1$.

Stichwortverzeichnis

Symbole

\int	10
()	3
*	3, 12
**	3
+	3, 12
-	3, 12
/	3, 12
:	3
:=	4
:	3
<>	12
[]	14
#	3
%	4, 15
%%	4, 15
%%%	4, 15
^	3
2-dimensionale Darstellungen	6
3-dimensionale Darstellungen	8

A

ableiten	10
abs	5
arccos	5
arcsin	5
arctan	5
arithmetisches Mittel	14
assume	4

C

complex	10
cos	5
crossprod	13

D

describe	14
diff	10
differenzieren	10
dotprod	12

E

evalf	3
Execution Group	3
exp	5

F

factor	5
fieldplot	9

Fließkomma-Berechnung	3
Funktion	5

G

geometricmean	14
geometrisches Mittel	14
Groß/Kleinschreibung	15

H

harmonicmean	14
harmonisches Mittel	14

I

I	15
infinity	6
Int	15
int	10, 15
integer	4
integrieren	10
Integral	10
inverse	13
Inverse einer Matrix	13

K

Kreuzprodukt	13
--------------------	----

L

left	10
limit	10
linalg	12, 13
line	6
Lineare Algebra	12
Liste	14
ln	5
log10	5

M

Matrix	12
Matrizenmultiplikation	13
mean	14
Mittelwert	14
multiply	13

P

Pakete	
linalg	12
plots	9
stats	14
describe	14

parametrische Darstellung	7
piecewise	10
plot	6, 7
plot3d	8
plots	9
point	6

Q

quadraticmean	14
quadratisches Mittel	14
Quadratwurzel	3

R

real	10
restart	15
right	10

S

Simplify	15
simplify	5
sin	5
skalare Multiplikation	12
Skalarprodukt	12

sqrt	3, 5
Stammfunktion	10
Statistik	14
stats	14

T

tan	5
-----------	---

U

unbestimmtes Integral	10
-----------------------------	----

V

Variable	4
Vektor	12
vereinfachen	5

W

with	3
Wurzel	3

Z

zeichnen	6
Zuweisung	4