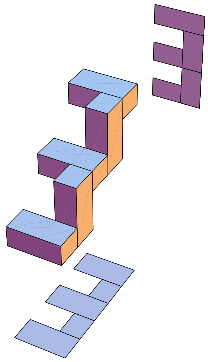

Kapitel 3

Maple als Taschenrechner verwenden

Selbstverständlich stellt Maple keinen sinnvollen Ersatz für Ihren Taschenrechner dar. Wenn Sie schnell einige Zahlen addieren möchten, ist Maple viel zu umständlich. Sie sollen hier also nicht ermuntert werden, Ihren Taschenrechner wegzuwerfen, vielmehr sollen Sie ein Gefühl dafür bekommen, wie Maple mit Zahlen(werten) umgeht. Dabei treten nämlich durchaus einige Unterschiede zur Behandlung von Zahlen durch den Taschenrechner hervor. So versucht Maple in der Regel, Genauigkeitsverluste durch die explizite Berechnung mathematischer Terme zu vermeiden. Wenn Sie Maple dazu zwingen, einen konkreten Zahlenwert auszurechnen, können Sie die gewünschte Stellenzahl (auch 100 oder 1000 Stellen!) einfordern. Spätestens jetzt wird Ihr noch so wissenschaftlicher Taschenrechner im Wettstreit mit Maple unterliegen.

Zahlformate

Maple stellt Zahlen in verschiedenen Formaten dar. Zunächst gibt es exakte Darstellungen für alle ganzen Zahlen beliebiger Größe und für beliebige Brüche (natürlich nur im Rahmen des zur Verfügung stehenden Speicherplatzes). Symbolische Darstellungen wie $\sqrt{3}$ oder $\sin(1)$ sind ebenfalls exakt. Für numerische Näherungen verwendet Maple Gleitkommazahlen, die eine Erweiterung des IEEE/754-Standards sind. Rechnungen mit solchen Zahlen können natürlich nicht mehr exakt sein, da Rundungsfehler unvermeidlich sind. Man kann aber die Genauigkeit der Zahldarstellung recht weit treiben, da Maple sich nicht auf die durch die Hardware zur Verfügung gestellte Gleitkomma-Arithmetik beschränkt. Mit der Systemvariablen `Digits` kann man die Stellenzahl der Mantisse auf beachtliche Größe ausdehnen. Mit `Maple_floats(MAX_DIGITS)` erhält man den maximalen Wert für die Stellenzahl der Mantisse (auf meinem Windows-2000-System z. B. 268435448). Entsprechend liefert `Maple_floats(MAX_EXP)` den größtmöglichen Exponenten einer Gleitkommazahl (Windows-2000: 2147483646). Nähere Informationen zu den Zahlformaten erhält man mit `?numerics` in der Online-Hilfe.

Grundrechenarten

Maple beherrscht natürlich die Grundrechenarten. Beachten Sie aber, dass jede Eingabe mit einem Semikolon (oder einem Doppelpunkt, dann erhält man aber keine Ausgabe) abgeschlossen werden muss und dass Multiplikationen explizit mit * angeschrieben werden müssen. Wie das letzte Beispiel zeigt, besteht auch die Möglichkeit, mehrere Berechnungen in einer Zeile anzuschreiben.

```
2+3;
```

$$5$$

```
2 3;
  syntax error:
  2 3;
    ^
```

```
2*3, 2^3, 5!
```

$$6, 8, 120$$

Wenn Brüche vom Computer als Gleitkommazahlen dargestellt werden (also 0.83333333 statt 5/6), dann kommt es zu einem Genauigkeitsverlust. 0.83333333 stimmt nur auf acht Nachkommastellen mit 5/6 überein. Aus diesem Grund vermeidet es Maple, Ergebnisse numerisch auszurechnen.

```
1/2;
```

$$1/2$$

```
1/2+1/3;
```

$$5/6$$

Wie Sie sehen, gilt das auch für die Berechnung mit verschiedenen Funktionen.

```
sqrt(2), sin(Pi/4);
```

$$\sqrt{2}\,, \frac{\sqrt{2}}{2}$$

Selbstverständlich lassen sich symbolische Ausdrücke wie `sqrt(2)` jederzeit numerisch auswerten. Im zweiten Parameter der Funktion `evalf` (evaluate floating point) können Sie die gewünschte Stellenzahl angeben. Wenn Sie darauf verzichten, rechnet Maple mit der voreingestellten Genauigkeit (siehe unten).

```
evalf( sqrt(2) );
```

$$1.414213562$$

```
evalf( sqrt(2), 50);
```

$$1.4142135623730950488016887242096980785696718753769$$

Beachten Sie aber den Unterschied zwischen symbolischen und numerischen Berechnungen! Bei symbolischen Berechnungen treten keine Rundungsfehler auf, Maple kann mathematische Verfahren zur Vereinfachung der Ausdrücke anwenden.

```
sqrt(2) * sqrt(8) - 4;
```

$$0$$

```
evalf(sqrt(2)) * evalf(sqrt(8)) -4;
```

$$-0.000000002$$

Rechengenauigkeit

Die Genauigkeit für numerische Berechnungen mit Gleitkommazahlen ist durch die Systemvariable `Digits` festgelegt und beträgt normalerweise zehn Stellen. Durch eine Veränderung dieser Variable kann die Genauigkeit eingestellt werden. Das beeinflusst zwar den absoluten Fehler bei numerischen Berechnungen, nicht jedoch die Tatsache an sich, dass Rundungsfehler auftreten.

```
Digits;
```

$$10$$

```
Digits:=50;
```

$$Digits := 50$$

```
evalf(sqrt(2)) * evalf(sqrt(8)) -4;
```

$$-3.0 \times 10^{-49}$$

```
Digits:=10;
```

$$Digits := 10$$

`Digits` hat keinen Einfluss auf die Genauigkeit symbolischer Berechnungen oder auf Berechnungen mit exakten Zahlen. Die Fakultät von 100 wird exakt berechnet, weil diese Funktion eine Funktion von den natürlichen Zahlen in die natürlichen Zahlen ist. Natürlich kann man sich mit `evalf` eine Näherung durch eine Gleitkommazahl berechnen lassen, die dann von `Digits` abhängt.

```
100!;
```

$$93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156518286253697920827223758251185210916864000000000000000000000000$$

```
evalf(100!,5),evalf(100!,20);
```

$$.93326 \times 10^{158}, .93326215443944152682 \times 10^{158}$$

Das Beispiel rechts geht nochmals auf den Unterschied zwischen symbolischer und numerischer Berechnung ein. `log[2](256)` berechnet den Logarithmus von 256 zur Basis 2. Das Ergebnis wird allerdings nicht vereinfacht. Wenn Sie es jetzt mit `evalf` numerisch auswerten, tritt ein Rundungsfehler auf. Wenn Sie Maple dagegen auffordern, das Ergebnis mit `simplify` symbolisch zu vereinfachen, liefert es das korrekte Ergebnis.

```
lg2:=log[2](256);
```

$$lg2 := \frac{\ln(256)}{\ln(2)}$$

```
evalf(lg2);
```

$$7.999999999$$

```
simplify(lg2);
```

$$8$$

Beachten Sie, dass eine nachträgliche Vergrößerung der Stellenzahl durch den Befehl `evalf(.., n)` die Genauigkeit nur scheinbar vergrößert. Obwohl das Ergebnis der Berechnung rechts fünfzig Stellen umfasst, sind ab der zehnten Nachkommastelle alle Ziffern falsch. Der Grund: Die Ausgangszahl in der Variablen $sq2$ wurde nur mit zehnstelliger Genauigkeit berechnet. Alle weiteren Berechnungen können nicht mehr genauer werden!

```
sq2:=evalf(sqrt(2));
```

$$sq2 := 1.414213562$$

```
evalf(1/sq2^2,50);
```

$$0.50000000026381803913919991547\\
70579394602463501555 6$$

Sehr große Zahlen können wahlweise in der Form $n * 10^m$ oder $n\mathtt{e}m$ geschrieben werden. Die zweite Variante hat den Vorteil, dass Sie sich manchmal das Schreiben von Klammern sparen können.

```
149.6*10^9 / (3*10^8);
```

$$498.6666666$$

Allerdings gelten Zahlen der zweiten Variante prinzipiell als Gleitkommazahlen, während mit Zahlen der Form $n * 10^m$ versucht wird, exakt (symbolisch) zu rechnen. In den Beispielen wird die Zeit (in Sekunden) berechnet, den ein Lichtstrahl von der Erde zur Sonne benötigt.

```
149.6e9 / 3e8;
```

$$498.6666667$$

Strings

Seit Release 5 kennt Maple auch echte Zeichenketten (Strings). Zeichenketten werden in doppelte Anführungszeichen eingeschlossen.

```
str1:="Maple hat Strings.";
```

str1 := `"Maple hat Strings."`

Ein Grundvorrat von Funktionen zur Verarbeitung von Strings steht ebenfalls zur Verfügung. Strings sind keine Symbole, insbesondere kann ein String nicht an einen Wert gebunden werden. Die maximale Länge von Strings ist plattformabhängig (bei 32-Bit-Systemen 268435439 Zeichen, das entspricht etwa 90000 Buchseiten!). Das seit Maple 7 vorhandene Package `StringTools` (s. Kapitel 33) stellt eine Vielzahl fortgeschrittener Funktionen zur Stringverarbeitung zur Verfügung.

```
convert(str1,symbol);
```

Maple hat Strings.

```
convert("abc",list);
```

```
["a","b","c"]
```

```
cat(%[]);
```

```
"abc"
```

Komplexe Zahlen, Matrizen, Statistikfunktionen

Maple rechnet mit komplexen Zahlen ebenso selbstverständlich wie mit normalen Zahlen.

```
-20 * I + 200 * I / (20 + 10 * I);
```

$4-12\,I$

```
abs(4-12*I); evalf(%);
```

$4\sqrt{10}$

12.64911064

Grundsätzlich nimmt Maple den Körper der komplexen Zahlen als den Bereich an, in dem Berechnungen durchgeführt werden.

```
sqrt(x^2);simplify(%);
```

$\sqrt{x^2}$

$csgn(x)x$

Will man Maple im Schulbereich verwenden, wenn komplexe Zahlen noch nicht bekannt sind, so kann das zu Irritationen führen. Die neben stehenden Beispiele zeigen solche Fälle, in denen die Annahme der komplexen Zahlen als Bereich für Berechnungen zu Ergebnissen führt, die nicht mit Ergebnissen über den reellen Zahlen übereinstimmen.

```
simplify((-8)^(1/3));
```

$1+I\sqrt{3}$

```
solve(x^2+2*x+5,{x});
```

$\{x=-1+2\,I\},\{x=-1-2\,I\}$

Abhilfe schafft hier das Package `RealDomain`. Nach seinem Aufruf mit `with(RealDomain)` wird die Liste der angepassten Funktionen mit einer Warnmeldung ausgegeben. Danach legt sich Maple für diese Funktionen als Zahlbereich die reellen Zahlen zu Grunde.

```
with(RealDomain):
```

Warning, these protected names have been redefined and unprotected: Im, Re, ^, arccos, ... simplify, sin, sinh, solve, sqrt, surd, tan, tanh

Leider hat das Package (Maple 7) noch einige Mängel, zum Beispiel liefert es im irreduziblen Fall einer Gleichung dritten Grades statt der drei Lösungen die falsche Antwort *undefined*.

```
simplify(sqrt(x^2)),(-8)^(1/3);
```

$$|x|, -2$$

```
solve(x^2+2*x+5,{x});
```

$$undefined$$

Seit Version 6 hat sich die Behandlung von Matrizen wesentlich vereinfacht. Matrizen können jetzt direkt eingegeben werden und zwar sowohl spalten- als auch zeilenweise.

```
mat1:=<< 1| 2| 3>,< 7|11|-3>>;
```

$$mat1 := \begin{bmatrix} 1 & 2 & 3 \\ 7 & 11 & -3 \end{bmatrix}$$

```
mat2:=<<0,-3,5>|<7,17,-1>>;
```

$$mat2 := \begin{bmatrix} 0 & 7 \\ -3 & 17 \\ 5 & -1 \end{bmatrix}$$

Zum Rechnen mit Matrizen wird für die Addition das Zeichen $+$, für die Subtraktion $-$, zur Multiplikation aller Matrizenelemente mit einem Skalar das Zeichen $*$ und für die nichtkommutative Matrizenmultiplikation ein Punkt (neu seit Version 6) verwendet.

```
mat3:=mat1 . mat2
```

$$mat3 := \begin{bmatrix} 9 & 38 \\ -48 & 239 \end{bmatrix}$$

```
3*mat1;
```

$$\begin{bmatrix} 3 & 6 & 9 \\ 21 & 33 & -9 \end{bmatrix}$$

Potenzen von (quadratischen) Matrizen werden mit `^` gebildet, die zu einer invertierbaren Matrix inverse Matrix erhält man mit `^(-1)`.

```
mat4:= mat2 . mat1 +
        <<1,0,0>|<0,1,0>|<0,0,1>>;
```

$$mat4 := \begin{bmatrix} 50 & 77 & -21 \\ 116 & 182 & -60 \\ -2 & -1 & 19 \end{bmatrix}$$

Weitergehende Befehle zur Matrizenrechnung befinden sich im Package `LinearAlgebra`. (Ab Version 6, das Package `linalg` aus früheren Versionen ist weitaus umständlicher, steht aber weiterhin zur Verfügung und ist insbesondere bei einfachen Plots sehr nützlich.)

```
mat4^3;
```

$$\begin{bmatrix} 2660498 & 4144637 & -1429365 \\ 6266048 & 9761072 & -3370056 \\ -54806 & -84631 & 35431 \end{bmatrix}$$

Dieses muss mit `with` aktiviert werden, bevor die Funktionen verwendet werden können. Anschließend kann beispielsweise die Determinante problemlos berechnet werden.

```
mat4^(-1);
```

$$\begin{bmatrix} \frac{1699}{2112} & -\frac{721}{2112} & -\frac{133}{704} \\ -\frac{521}{1056} & \frac{227}{1056} & \frac{47}{352} \\ \frac{31}{528} & -\frac{13}{528} & \frac{7}{176} \end{bmatrix}$$

```
with(LinearAlgebra): Determinant(mat4)
```

$$4224$$

Statistikfunktionen befinden sich ebenfalls in einem eigenen Package. Im Beispiel unten werden zu einer Zahlenreihe der Mittelwert, die Varianz und daraus die Standardabweichung berechnet.

```
with(stats): with(describe):
data:=[ 10.2, 9.9, 10, 9.95, 10, 10.1, 10.4,  9.3, 9.85, 10.05, 10.1 ];
```

$$data := [10.2, 9.9, 10, 9.95, 10, 10.1, 10.4, 9.3, 9.85, 10.05, 10.1]$$

```
mean(data);
```

$$9.986363636$$

```
variance(data);
```

$$.06776859503$$

```
sqrt(%);
```

$$.2603240193$$