

# Basiswissen

Was man im Leben  
braucht ist Basiswissen  
und die Fähigkeit, sich  
wichtige Informationen zu  
erarbeiten. Unsere  
Schulen wissen gar nicht  
was das ist und unsere  
Universitäten entfernen  
sich immer mehr davon.<sup>1</sup>

Ulf

## 1.1. Einführung

Dieses Buch wurde für Maple 15 unter den Betriebssystemen Microsoft Windows, Apple Mac OS X und Linux konzipiert – die meisten Quelltexte sollten aber auch mit anderen Maple-Versionen funktionieren. Zum Verständnis des behandelten Stoffes sind Grundkenntnisse in Analysis und Linearer Algebra empfehlenswert, darüber hinaus könnte Programmiererfahrung hilfreich sein.

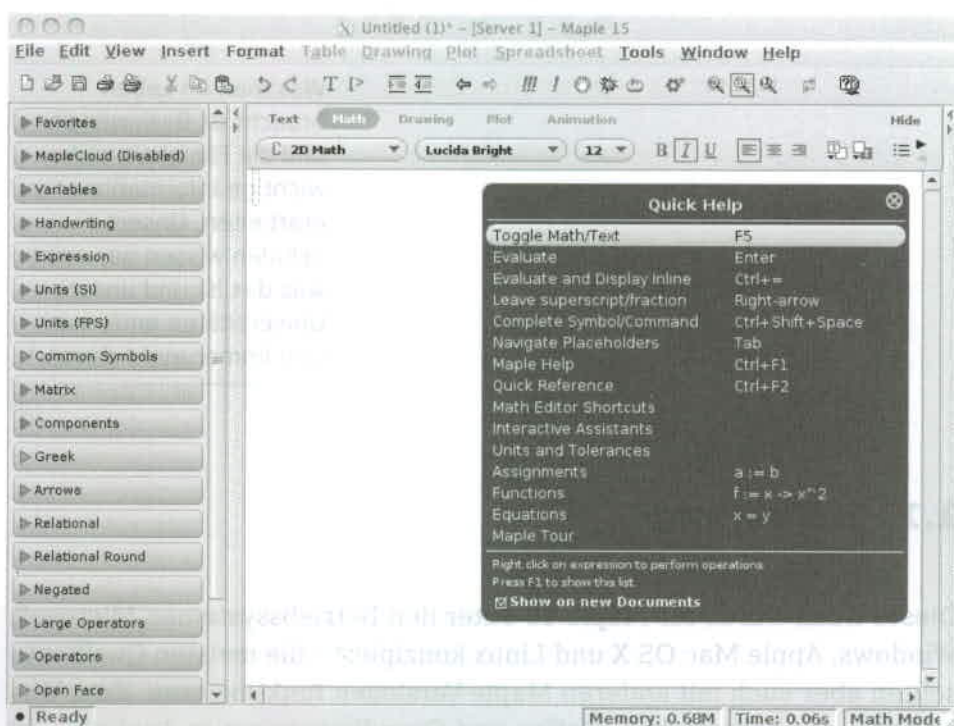
Unter Windows findet man das Programm Maple im Startmenü oder kann es über eine Verknüpfung auf dem Desktop ausführen. Unter Mac OS X befindet sich die Applikation im Ordner */Applications* und kann über das entsprechende Dock-Icon gestartet werden. Unter Linux führt man in einem Terminal-Emulator den Befehl `xmaple` aus oder sucht einen Eintrag im Applikationen-Menü seiner Desktop-Umgebung.

<sup>1</sup> Bei unzähligen Kaffees in diversen Diskussionen haben die Autoren flapsige Sprüche gesammelt, deren Erwähnung sie für unverzichtbar halten.

## 1. Basiswissen

Nach einer kurzen Ladesequenz wird die graphische Benutzeroberfläche von Maple angezeigt. Zu sehen ist ein leeres Dokument (gemeint ist die weiße Arbeitsfläche) im sogenannten *Document Mode*, der modernsten einiger von Maple zur Verfügung gestellten Arbeitsumgebungen. Sie eignet sich zum einfachen Rechnen wie zum Verfassen wissenschaftlicher Arbeiten. Auf die Unterschiede zwischen den verschiedenen Arbeitsmodi wird später eingegangen.

Am oberen Rand des Fensters befinden sich eine Menü- und eine Symbolleiste. Hier werden in der Regel alle Aktionen ausgeführt (bis man mit den zugehörigen Tastenkombinationen vertraut ist). Am linken (und optional auch am rechten) Rand ist ein sogenanntes *Dock* verankert, das durch einen Mausklick ein- und ausgeklappt werden kann. In einem Dock sind *Paletten* zur Eingabe von mathematischen Symbolen, Sonderzeichen, Kontrollelementen, usw. angeordnet.



**Abbildung 1.1.:** Screenshot eines neuen Maple-Dokuments.

Am oberen Rand befindet sich eine Menü- und eine Symbolleiste. Am linken Rand ist ein Dock, auf dem Paletten zur Eingabe von Sonderzeichen angeordnet sind. Am oberen Rand der Arbeitsfläche ist eine Kontextleiste, mit der Verarbeitung und Formatierung der Nutzereingaben beeinflusst werden können. Die Quick Help wird halbdurchsichtig auf der rechten Seite des Dokuments eingeblendet und hilft dem Erstanwender sich zu orientieren.

Am oberen Rand des Dokuments ist die sogenannte Kontextleiste, mit der Verarbeitung und Formatierung der Nutzereingaben beeinflusst werden können.<sup>2</sup> Auf der rechten Seite des Dokuments wird die *Quick Help* eingeblendet, welche die wichtigsten Tastenkombinationen zur Interaktion mit Maple präsentiert.

Schon mit diesem kurzen Überblick und etwas mathematischer Intuition kann man erstaunlich viel Funktionalität aus dem Software-Paket ziehen. Daher wird zunächst die Arbeit im Dokument selbst vorgestellt und auf die fortgeschrittenen Funktionen der graphischen Umgebung schrittweise eingegangen, sobald sie benötigt werden.

## 1.2. Erste Schritte

Man kann Maple als interaktiven Taschenrechner benutzen. Dazu klickt man mit der Maus einmal in das Maple-Dokument, tippt eine Rechenvorschrift ein und bestätigt mit der Enter-Taste. Als Beispiel soll folgende Rechnung dienen:

> 2 + 3

5 (1)

Die erste Zeile ist linksbündig eingerückt und mit einem Pfeil > als Eingabezeile<sup>3</sup> gekennzeichnet. Der Text in dieser Zeile wird vom Anwender selbst verfasst. Die Eingabe wird durch Betätigen der Enter-Taste **ausgewertet**<sup>4</sup>. Die zweite Zeile ist zentriert dargestellt und gibt die daraufhin von Maple generierte Ausgabe wieder. Maple hat den Ausdruck ausgerechnet. Die rechtsbündig dargestellte Nummer zählt die Ausgabezeilen von Maple durch und kann für Verweise genutzt werden.

Das Dokument interpretiert Eingaben in einem sogenannten Kontext. Vereinfacht ausgedrückt kann Maple so Text, Formeln und Grafiken auseinanderhalten. Voreingestellt ist der mathematische Kontext (die *Math*-Schaltfläche in der Kontextleiste ist aktiviert), d.h. Eingaben werden vom Dokument anfangs als Formeln interpretiert.

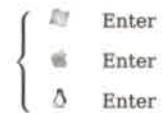
Ein Wechsel zwischen dem *Math*-Kontext und dem *Text*-Kontext ist über die Schaltflächen in der Kontextleiste möglich. Wenn man **den Kontext wechselt** (zum *Text*-Kontext) und einen Kommentar ins Dokument schreibt, wird dieser wörtlich übernommen und beim Bestätigen nicht ausgewertet:

> Ob die Punkt-vor-Strich-Regel gilt?

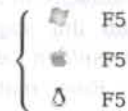
Stattdessen springt die Eingabemarke (wie in einer Textverarbeitung) nur<sup>5</sup> in eine neue Zeile. Um wieder zu rechnen, wechselt man den Kontext erneut (zurück zum *Math*-Kontext).

<sup>2</sup> Diese Beschreibung bezieht sich nur auf die Standardkonfiguration. Maple bietet eine sehr flexible graphische Oberfläche, die der Anwender auf seine Bedürfnisse anpassen kann.

<sup>3</sup> Dieser Pfeil wird in der graphischen Oberfläche standardmäßig nicht angezeigt, nur in der Maple-Konsole. Er dient in diesem Buch lediglich zur besseren Unterscheidbarkeit von Eingabe und Ausgabe.



<sup>4</sup> Hilfreiche Tastenkürzel werden am Seitenrand notiert. Sie erleichtern die Arbeit mit mathematischer Software enorm!



## 1. Basiswissen

<sup>5</sup> Es wird keine Ausgabe erzeugt. Erscheint doch etwas, hat man das Dokument vermutlich aus Versehen im sogenannten *Worksheet Mode* geöffnet. Um in den *Document Mode* zu gelangen, muss man wohl oder übel ein neues Dokument öffnen.

Mit einem Sternsymbol \* kann man multiplizieren:

$$> 12 - 2 \cdot 5$$

$$2 \quad (2)$$

Offenbar gilt die Punkt-vor-Strich-Regel. Außerdem wandelt Maple das Sternsymbol schon während der Eingabe in eine mathematisch ästhetische Schreibweise (den Malpunkt) um.

Mit einem Schrägstrich / als Divisionszeichen kann man den Ausdruck  $21/13$  (eine geläufige Näherung für den goldenen Schnitt) eingeben:

$$> \frac{21}{13}$$

$$\frac{21}{13} \quad (3)$$

Unter Umständen kürzt Maple dabei den Bruch:

$$> \frac{8}{12}$$

$$\frac{2}{3} \quad (4)$$

<sup>6</sup> Der blinkende Strich wird als *Einfügemarke* oder *Marke* bezeichnet.

Beim Eingeben des Schrägstriches hat Maple den Ausdruck automatisch in eine Bruchschreibweise gebracht und die Einfügemarke<sup>6</sup> im Nenner platziert. Dort kann man wie gewohnt weiterschreiben und den Ausdruck schließlich auswerten. Dass sich die Ausgabe nicht von der Eingabe unterscheidet, liegt daran, dass Maple gekürzte Brüche schon als vollständig ausgewertet ansieht.<sup>7</sup> Es wird noch klarer werden, warum es Sinn macht hier nicht den Dezimalwert auszugeben.

### Navigation

<sup>7</sup> Man hätte auf die Idee kommen können, dass aus Versehen der *Text-Kontext* aktiv ist. Eine Verwechslung des Kontextes ist eine häufige Fehlerursache in Maple, zumal man ihn sogar *innerhalb derselben Zeile* ändern kann. Einen mathematischen Ausdruck erkennt man im Zweifel immer daran, dass er *grau gestrichelt umrahmt* wird, wenn man ihn mit der Maus anklickt.

Formeln im *Math*-Kontext können auf verschiedene Weisen formatiert werden. Die voreingestellte Notation ist die sogenannte *2-D Math Notation*, die eine intuitive Schreibweise von Formeln (wie auf einem Blatt Papier) ermöglicht. Sie hebt sich damit von der klassischen eindimensionalen Notation ab (wie sie beispielsweise von älteren Schultaschenrechnern bekannt ist). Später werden noch weitere Notationsmöglichkeiten vorgestellt, aber zunächst wird die *2-D Math Notation* behandelt.

Neben den offensichtlichen Vorteilen bringt die zweidimensionale Notation auch Schwierigkeiten mit sich. Um komplizierte Formeln einzugeben oder zu bearbeiten, muss die Position der Einfügemarke berücksichtigt werden. Es werden zwei Varianten zum Bewegen der Marke präsentiert: Tastaturnavigation und Mausnavigation.

Von Tastaturnavigation spricht man, wenn man die Marke mit Hilfe der Pfeiltasten auf der Tastatur verschiebt. Im einfachsten Fall benötigt man

## 1.2. Erste Schritte

nur den Rechtspfeil um die Marke aus verschachtelten<sup>8</sup> Positionen zu befreien. Indem man in folgendem Beispiel nach dem ersten Bruch die **Einfügemarke freistellt**, kann man nun zwei Näherungen des goldenen Schnitts vergleichen:

$$> \frac{34}{21} - \frac{21}{13}$$

$$\frac{1}{273} \quad (5)$$

Dies entspricht weniger als einem halben Prozent relativer Abweichung! Diese guten Näherungswerte des goldenen Schnitts wurden aus seiner Darstellung als nicht abbrechender (man sagt auch unendlicher) Kettenbruch gewonnen. Durch Abbrechen des Kettenbruchs kann man beliebig gute Näherungen erzeugen:

$$> 1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + 1}}}}}}$$

$$\frac{21}{13} \quad (6)$$

Der nächstlängere Kettenbruch würde 34/21 liefern. Bei solchen Ausdrücken erweist sich die Mausnavigation als komfortabler. Statt mit den Pfeiltasten zu einem Teil des Kettenbruchs zu navigieren, kann man auch mit der Maus an die Stelle klicken, an der die Einfügemarke positioniert werden soll.

### Dezimaldarstellung

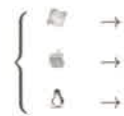
Es gibt mehrere Ansätze, die Dezimaldarstellung von Ergebnissen zu erhalten. Die expliziteste ist die Funktion `evalf` (*evaluate float*), die (falls möglich) den Dezimalwert ihres Arguments zurückgibt:

$$> \text{evalf}\left(\frac{21}{13}\right); \text{evalf}\left(\frac{34}{21}\right);$$

$$1.615384615$$

$$1.619047619 \quad (7)$$

Funktionen<sup>9</sup> sind Vorschriften, die aus einer Reihe von Eingaben eine Ausgabe erzeugen. Sie werden aufgerufen, indem man ihren Namen schreibt und dann<sup>10</sup> in runden Klammern die Argumente (Parameter) aufführt. Sind es mehrere Argumente, werden diese durch Kommata getrennt. Eine schließende Klammer „erkennt“ ihre zugehörige öffnende Klammer, daher ist es in diesem Fall nicht nötig, die Einfügemarke freizustellen.



<sup>8</sup> Mit *Schachtelung* ist gemeint, dass die Einfügemarke in einem Teilbereich des Ausdrucks „gefangen“ ist. Zwei repräsentative Fälle sind z.B. Brüche und Potenzen. Die Marke „freistellen“ bedeutet, diese Bereiche zu verlassen.

<sup>9</sup> Funktionen sind möglicherweise – aber nicht zwangsläufig – mathematische Funktionen. Maple-Funktionen werden noch genauer erläutert werden. Zunächst soll es genügen sie aufzurufen.

<sup>10</sup> Zwischen dem Funktionsnamen und der öffnenden Klammern darf kein Leerzeichen stehen. Maple interpretiert den Ausdruck sonst als Produkt aus dem Funktionsnamen und den Argumenten.

## 1. Basiswissen

Wie im letzten Beispiel kann man mehrere Ausdrücke in eine Zeile schreiben. Als Trennungszeichen wird das Semikolon ; verwendet. Genau genommen ist es kein Trennungszeichen, sondern eines mehrerer möglicher Abschlusszeichen. In früheren Maple-Versionen war der Abschluss von Ausdrücken noch obligatorisch: Man musste jede Eingabe z.B. mit einem Semikolon beenden. Da es heute noch als guter Stil gilt, werden in Folgenden Abschlusszeichen verwendet.

<sup>11</sup>Das sind Vor- und Nachkommastellen, ohne führende oder nachstehende Nullen.

Die Ergebnisse werden untereinander ausgegeben und mit einer gemeinsamen Nummer ausgezeichnet. Die Dezimalzahlen sind *nicht* exakt, sondern werden auf 10 signifikante Stellen<sup>11</sup> gerundet. Eine exakte Angabe (ohne Tricks) wäre auch gar nicht möglich gewesen, denn  $21/13$  hat die nicht abbrechende, periodische Dezimaldarstellung  $1.\overline{p615384}$ , wobei  $p$  den Beginn der Periode kennzeichnet. Ebenso entspricht  $34/21$  dem Wert  $1.\overline{p619047}$ .

Die Anzahl 10 der signifikanten Dezimalstellen ist von Maple voreingestellt. Werte, die die interne Arbeit von Maple beeinflussen, sind in sogenannten Umgebungsvariablen gespeichert. Die Rechengenauigkeit wird durch die Umgebungsvariable `Digits` gesteuert. Man kann sie ändern:

```
> Digits := 20;
```

```
20 (8)
```

```
> evalf(21/13); evalf(34/21);
```

```
1.6153846153846153846 (9)
```

```
1.6190476190476190476
```

<sup>12</sup>Ein Ausdruck kann durch einen Trick eine Ausgabe erzeugen, obwohl er mit einem Doppelpunkt abgeschlossen ist! Dieses Verhalten wird später in der Programmierung wichtig werden. (vgl. Seite 62)

Wie erwartet ist die Zahl der Nachkommastellen auf 19 gestiegen. Die unnötige Ausgabe des Ausdrucks (8) kann mit dem Doppelpunkt : als Abschlusszeichen vermieden werden. Er unterdrückt jegliche automatische<sup>12</sup> Ausgabe:

```
> Digits := 10 :
```

So kann man ein Dokument leicht übersichtlich halten. Die Änderung an der Umgebungsvariable `Digits` wurde manuell rückgängig gemacht, denn Umgebungsvariablen werden nicht nur temporär für die nächste Auswertung, sondern für die gesamte Sitzung (d.h. bis Maple neu gestartet wird), allerdings nur für das aktuelle Dokument geändert.

Wenn einmal etwas nicht so funktioniert wie erwartet, kann die Funktion `restart` weiterhelfen. Sie bringt die Maple-Umgebung in den Startzustand zurück, als würde man Maple neu starten. Dabei gehen insbesondere alle Änderungen an Umgebungsvariablen verloren.

Statt des letzten Ausdrucks hätte man also auch schreiben können:

```
> restart;
```

Es spielt dabei keine Rolle, ob man die Funktion mit einem Semikolon oder Doppelpunkt abschließt, da sie keine Ausgabe erzeugt. Zu beachten ist auch, dass Funktionen ohne Argumente kein leeres Klammernpaar nachgestellt wird. Um sie sprachlich abzugrenzen, nennt man sie *Kommandos*.

Mit der Hilfe von Funktionen kann man den goldenen Schnitt exakt berechnen:

```
> (1 + sqrt(5)) / 2;
```

$$\frac{1}{2} + \frac{1}{2}\sqrt{5} \quad (10)$$

Die Funktion `sqrt` (*square root*) berechnet die Quadratwurzel ihres Arguments. Die zunächst kursiven Buchstaben der Funktion haben sich nach der Eingabe<sup>13</sup> aufgestellt. Mit dieser Formatierung kennzeichnet Maple mathematische Funktionen – sehr praktisch um Tippfehler zu bemerken. Die Wurzel-Funktion wurde in ein Wurzel-Symbol umgewandelt und das Ergebnis auf eine Standardform gebracht.

## Verweise

Den Dezimalwert des letzten Ergebnisses kann man mit einer Kurzschreibweise bestimmen:




```
> evalf(%);
```

$$1.618033988 \quad (11)$$

Hier wurde als Argument nur das Prozentzeichen % übergeben. Dieser *Ditto*-Operator repräsentiert das zuletzt berechnete Ergebnis. So kann man den Dezimalwert des goldenen Schnitts ausgeben lassen, ohne den Ausdruck noch einmal zu kopieren oder abzuschreiben. Die Verwendung des *Ditto*-Operator ist umstritten, da ein solcher Ausdruck keine eigenständige Bedeutung hat und unbeabsichtigte Ergebnisse erzeugen kann. Maßvoll eingesetzt ermöglicht er aber sehr elegante Kurzschreibweisen.

Statt mit dem *Ditto*-Operator auf das letzte Ergebnis zu verweisen, kann man auch alle erzeugten Ausgaben über ihre Nummer (die kleine Zahl rechts in Klammern) ansprechen. Durch das **Einfügen eines Verweises** kann man Maple anweisen, ein Ergebnis für die Eingabe wiederzuverwenden. Solche Verweise sind unproblematischer als der *Ditto*-Operator, da sie nicht von der Auswertungschronik beeinflusst werden.

<sup>13</sup>Wenn man die Zeichen der Reihe nach eintippt, halbiert Maple nur den Term `sqrt(5)` und dehnt den Bruch nicht auf den linken Summanden aus. Verantwortlich ist die Punktvor-Strich-Regel. Um den Bruch richtig einzugeben, kann man zunächst `1/2` schreiben und dann zum Zähler navigieren, um den Ausdruck `+sqrt(5)` hinter der 1 zu ergänzen. Eine andere Möglichkeit ist, erst den Zähler einzugeben, ihn zu markieren und dann den Schrägstrich / zu tippen. Dabei wird der markierte Term nicht ersetzt, sondern zum Zähler umfunktioniert. Man kann Teile von Ausdrücken markieren, indem man den Mauszeiger bei gedrückter Maustaste bewegt, oder indem man bei gehaltener Shift-Taste mit den Pfeiltasten navigiert.

{		Strg + L
		Cmd + L
		Strg + L

## 1. Basiswissen

Man kann das Wurzel-Symbol, das Maple beim Auswerten der Wurzel-Funktionen automatisch setzt, auch manuell eingeben. Dazu bewegt man die Maus über das linke *Dock* und klappt durch Anklicken des Titels die *Expression*-Palette aus. Dort ist das Symbol  $\sqrt{a}$  zum Wurzelziehen dargestellt. Durch Anklicken fügt man es an der Position der Einfügemarke ins Dokument ein. Die Marke wird dabei automatisch unter dem Wurzelsymbol platziert. Durch den Einsatz von Paletten kann man zahlreiche mathematische Funktionen und Symbole verwenden, ohne die Maple-internen Funktionsnamen zu kennen.

### Paletten




Die Verwendung von Paletten zum Einfügen mehrerer mathematischer Symbole soll am Beispiel des goldenen Winkels (der z.B. in Sonnenblumen auftritt) erfolgen:

$$> \frac{2 \cdot \pi}{\left(\frac{1 + \sqrt{5}}{2}\right)^2} : evalf(\%);$$

2.399963232 (12)

<sup>14</sup>Man hätte natürlich auch unmittelbar `evalf` auf den Ausdruck anwenden können. Bei größeren Formeln kann die angegebene Variante etwas übersichtlicher sein.

<sup>15</sup>Das Schlüsselwort wird mit großem P und kleinem i geschrieben. Schreibt man `pi` klein, interpretiert Maple es als Variablennamen.

Escape oder Strg + Space		}
Escape		
Escape oder Strg + Shift + Space		

Das Wurzelsymbol ist bereits bekannt. Das Quadrat als Spezialfall der Potenz findet man als Symbol  $a^b$  in der gleichen Palette. Die Naturkonstante  $\pi$  ist das erste Symbol in der *Common Symbols*-Palette. Den gesamten Ausdruck konstruiert man, indem man viel navigiert oder geschickt Teilausdrücke markiert. Um direkt den Dezimalwert zu erhalten, kann man wie gezeigt die Ausgabe (mit dem Doppelpunkt) unterdrücken und im Anschluss<sup>14</sup> von `evalf` das letzte (zuvor *unterdrückte*) Ergebnis auswerten lassen.

Formeln zu bauen kann viel Zeit kosten, besonders wenn man Symbole in verschiedenen Paletten suchen muss. Zum Glück wird auf Kurzschreibweisen hingewiesen, wenn man mit der Maus auf eines der Symbole zeigt. Zum Beispiel schreibt man Potenzen mit dem Dachzeichen  $\wedge$  und die Konstante  $\pi$  als `Pi`<sup>15</sup>.

Um aus den gelernten Abkürzungen wieder Symbole zu erzeugen, kann man die **Autovervollständigung** nutzen. In Schlüsselwörtern wie `sqrt` oder `Pi` präsentiert Maple dann eine Auswahl von Vorschlägen, welche Schlüsselwörter oder Symbole gemeint sein könnten. Durch Anwählen mit den Pfeiltasten und Bestätigen mit `Enter` oder der Leertaste (oder auch durch Anklicken mit der Maus) wird der Vorschlag übernommen.



## 1.3. Die Hilfe

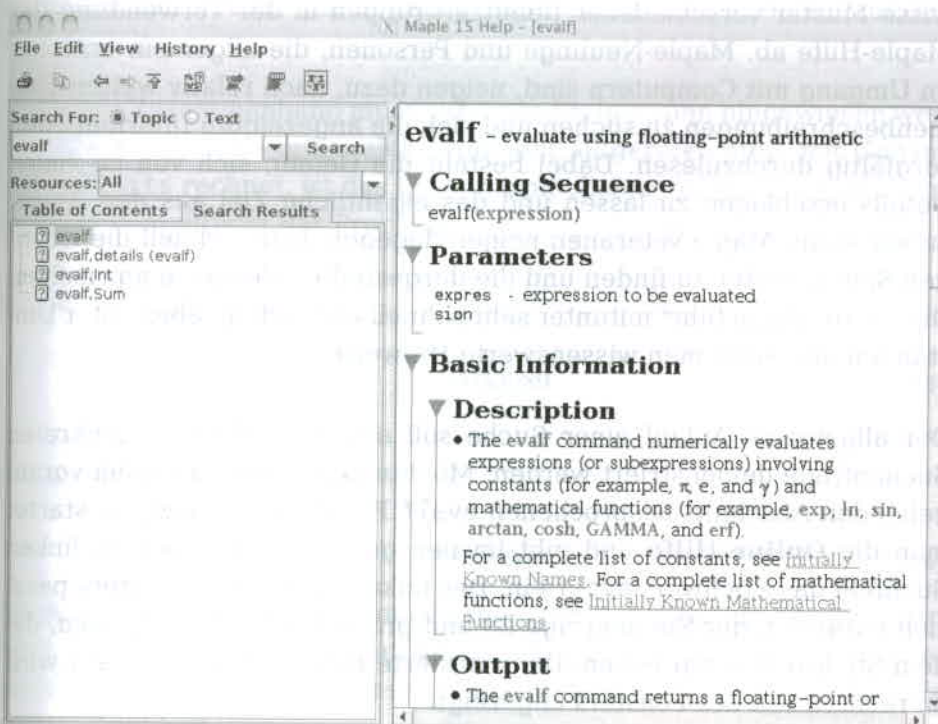
Maple hat sich in den vergangenen 30 Jahren zu einem weit fortgeschrittenen Computer-Algebra-System entwickelt. In der Regel stehen für alle denkbaren Probleme schon zahlreiche Hilfsmittel zur Verfügung, die entweder direkt in Maple integriert sind oder von unabhängigen Entwicklern ergänzt wurden.

Auf Anfänger wirkt diese Fülle an Werkzeugen jedoch oft abschreckend – schließlich setzt sie eine gewisse Bereitschaft voraus, Zeit und Nerven zur Einarbeitung in die neue Umgebung zu investieren. Während der ersten Wochen mit Maple kann eine Menge Frust entstehen, wenn man sich selbst die Frage stellt: „So etwas Triviales muss Maple doch können! Wie mach ich das bloß?“ Die Maple-Hilfe versucht Antworten auf diese und ähnliche Fragen zu geben. Sie ist wohl einer der wichtigsten Bestandteile von Maple und unterstützt gleichermaßen Anfänger und erfahrene Anwender bei der Arbeit mit dem Software-Paket.

Die **Online-Hilfe** erreicht man über den Menüpunkt:

*Help* → *Maple Help*

- ⌘ Strg + F1
- ⌘ Cmd + F1
- ⌘ Strg + F1



**Abbildung 1.2.:** Screenshot der Maple-Hilfe.

Am oberen Rand befindet sich eine Symbolleiste. Im linken Teil befindet sich ein Navigationsbereich mit einem strukturierten Inhaltsverzeichnis und einer Stichwortsuchfunktion. Im rechten Teil (dem Hauptteil) des Fensters wird der ausgewählte Hilfe-Eintrag angezeigt. Dort finden sich Hinweise zur Syntax, eine Beschreibung und einige Beispiele. Unterstrichene Begriffe sind Hyperlinks zu anderen Hilfe-Themen.

## 1. Basiswissen

In einem neuen Fenster öffnet sich dann der Hilfe-Browser. Nicht nur die Namensgebung erinnert ein wenig an das Internet, sondern auch Optik und Benutzung des Hilfe-Systems zeigen große Ähnlichkeit mit dem Besuchen von Webseiten.

Die Suche nach Informationen erfolgt auf zweierlei Weise. In einem ersten Schritt sollte der Anwender sich grob auf ein Thema festlegen, über das er Näheres erfahren möchte. Zu diesem Zweck kann er entweder mit der Suchfunktion (links-oben) die Hilfe nach Stichworten durchsuchen oder er versucht im hierarchisch strukturierten Inhaltsverzeichnis (links-unten) fündig zu werden. In beiden Fällen öffnet sich im Hauptteil des Fensters ein schlicht gestalteter Eintrag mit einer sachlichen Beschreibung des Schlagwortes (wie in einem Lexikon) und Anwendungsbeispielen. Im zweiten Schritt versucht der Anwender, seine Suche durch die zusätzlichen Hinweise und Schlagwörter zu konkretisieren. Hierzu befinden sich Hyperlinks (Verweise auf andere Hilfe-Einträge, die grün und unterstrichen markiert sind) im Text und am Ende der Seite, die auf Informationen zu ähnlichen Themen verweisen.

Wenn Maple eine Zeit lang produktiv eingesetzt wird, zeichnen sich gewisse Muster verschiedener Benutzergruppen in der Verwendung der Maple-Hilfe ab. Maple-Neulinge und Personen, die allgemein unsicher im Umgang mit Computern sind, neigen dazu, nach relativ weiten Themenbeschreibungen zu suchen und sich die angezeigten Informationen sorgfältig durchzulesen. Dabei besteht die Gefahr, sich von zu vielen Details erschlagen zu lassen und das eigentliche Ziel aus den Augen zu verlieren. Maple-Veteranen neigen dagegen dazu, schnell die richtigen Schlagwörter zu finden und die dargestellten Beispiele anzusehen. Dieses Vorgehen führt mitunter sehr schnell zum Erfolg, aber unter Umständen übersieht man wertvolle Hinweise.

Der allgemeine Ablauf einer Suche soll am Beispiel einer konkreten Suchanfrage demonstriert werden. Möchte man mehr zu der im vorangehenden Abschnitt besprochenen `evalf` Funktion erfahren, so startet man die **Online-Hilfe** und gibt im neu geöffneten Fenster im linken Suchfeld das Stichwort `evalf` ein. Der linke Bereich des Fensters passt sich daraufhin der Suchanfrage an und präsentiert einige Themen, die dem Stichwort entsprechen. Der markierte Hilfe-Eintrag zu `evalf` wird im rechten Teil des Fensters angezeigt.

Durch die Überschrift und die Syntaxbeschreibung erfährt man sofort, dass es sich um eine Funktion handelt, die einen Ausdruck numerisch auswertet. Aus der Beschreibung geht hervor, dass sowohl Konstanten als auch mathematische Funktionen ausgewertet werden können, und es sind weiterführende Verweise zu beiden aufgeführt. Schließlich wird die Verwendung von `evalf` an einigen kurzen und selbsterklärenden Beispielen veranschaulicht.

Es gibt noch eine weitere Möglichkeit, die Maple-Hilfe aufzurufen. Wenn während des Arbeitens Fragen zu einem konkreten Stichwort aufkommen, hilft die integrierte help-Funktion weiter. Im `evalf`-Beispiel hätte man (da der Funktionsname schon bekannt ist) auch `help(evalf)` oder kurz `?evalf` als Befehl im Maple-Dokument aufrufen können, um direkt die gewünschte Seite der Online-Hilfe anzuzeigen. Für Profis existiert das Kürzel `??evalf`, das lediglich eine Liste von Beispielen anzeigt.

Oft gibt es zu Hilfe-Themen wie Funktionen nicht nur eine Hauptseite, sondern noch eine weitere Seite mit Detail-Informationen oder gar eine ganze Reihe von dedizierten Unterseiten zu ausgewählten Schwerpunktthemen. Zum `evalf`-Eintrag findet man beispielsweise eine Details-Seite und zwei Unterseiten über das Zusammenspiel mit der `Sum`- und `Int`-Funktion. (Auf die letztgenannten beiden wird vorerst nicht eingegangen, da sie inhaltlich zu weit führen.)

Die Details-Seite von `evalf` führt einen zweiten Parameter `n` ein, der die Anzahl der Ausgabeziffern steuert.<sup>16</sup> Im Haupt-Eintrag wird er nicht explizit<sup>17</sup> erwähnt, da die Entwickler ihn nicht als relevant für die Kernaufgabe von `evalf` ansehen. In gewissen Situationen kann er allerdings von Vorteil sein, da man nicht manuell die Umgebungsvariable `Digits` anpassen muss. Auch hierzu findet man weitere Informationen auf der Details-Seite.

In diesem Zusammenhang muss auf einen Stolperstein hingewiesen werden. Da `evalf` ohne optionales Argument aufgerufen stets mit Genauigkeit `Digits` rechnet, ist das Ergebnis möglicherweise nicht exakt. Ein Beispiel verdeutlicht das Problem:

$$\begin{aligned} > \text{evalf}\left(\frac{75025}{46368}\right); \\ & \qquad \qquad \qquad 1.618033989 \qquad \qquad \qquad (13) \end{aligned}$$

$$\begin{aligned} > \text{evalf}\left(\frac{121393}{75025}\right); \\ & \qquad \qquad \qquad 1.618033989 \qquad \qquad \qquad (14) \end{aligned}$$

Maple zeigt für beide Näherungen des goldenen Schnitts den gleichen Wert an, weil sie ihn beide auf wenigstens 10 Dezimalstellen genau approximieren. Gleichheit von `evalf`-Werten ist so gesehen schwächer als echte Gleichheit.

Schließlich liefert die Details-Seite eine kurze Anleitung, wie man die `evalf`-Befehlsschnittstelle um eigene Unterroutinen erweitern kann, um beispielsweise selbstdefinierte Konstanten und Funktionen zu behandeln. Dieser Punkt geht weit über den aktuellen Kenntnisstand hinaus und wird in einem späteren Teil des Buches wieder aufgegriffen. Man sieht hier aber schon, dass die Maple-Hilfe auch bei fortgeschrittenen Themen gut gepflegt und sehr ausführlich ist.

<sup>16</sup>Das zweite Argument taucht in eckigen Klammern direkt hinter dem Funktionsnamen auf, statt wie erwartet mit Komma getrennt hinter dem ersten Argument. `evalf` verwendet hier eine eher ungebräuchliche Syntax für optionale Argumente.

<sup>17</sup>Immerhin wird in den Beispielen darauf hingewiesen.

## 1. Basiswissen

Maple bietet eine sogenannte *Quick-Reference*, die einen Überblick über die meist genutzten Befehle und Funktionen liefert. Eine gedruckte Version ist im Lieferumfang von Maple enthalten, aber man findet sie auch auf der Webseite von Maplesoft und direkt in Maple zum schnellen Nachsehen oder Drucken:

*Help* → *Quick Reference*

Gerade Anfänger profitieren von so einer kurzen Befehlsübersicht.

### 1.4. Syntax

Maple nur als Taschenrechner zu verwenden, würde einem Computer-Algebra-System nicht gerecht. Im Folgenden werden die grundlegenden Techniken vorgestellt, die das System dynamisch und für Mathematiker, Naturwissenschaftler und Ingenieure interessant machen, aber auch die Probleme, die dabei auftreten können.

#### Variablen

Anstatt wiederholt Verweise auf alte Ergebnisse zu setzen, bietet Maple auch die Möglichkeit, Zahlen in Variablen zu speichern. Das Konzept wurde bereits mit der vordefinierten *Digits*-Variable eingeführt, aber man kann auch eigene Variablen definieren:

```
> a := 3;
```

3 (15)

Auf diese Weise wurde der Wert 3 unter dem Variablennamen *a* hinterlegt bzw. abgespeichert und kann jederzeit abgerufen werden:

```
> a;
```

3 (16)

Selbstverständlich kann man Variablen auch in Rechnungen verwenden:

```
> a + 2;
```

5 (17)

Die gleiche Variable kann mehrfach in einen Ausdruck eingehen:

```
> a2 - a;
```

6 (18)

Schließlich kann man eine Variable mit einem neuen Wert überschreiben, wodurch ihr alter Wert irreversibel verloren geht:

```
> a := 4;
```

4 (19)

Ähnlich wie bei Verwendung des *Ditto*-Operators, ist ein Ausdruck isoliert betrachtet nun nicht mehr eindeutig. Wenn man im Dokument die letzte Zeile über dieser Definition neu berechnet (indem man sie mit der Maus anklickt und auswertet), ergibt sich ein anderes Ergebnis:

```
> a2 - a;
```

12 (18)

Die Zeilennummer (18) der Rechnung ist dabei gleich geblieben, da sich die Position im Dokument nicht geändert hat. Verweise auf diese Zeile werden ab jetzt also auch ein anderes Ergebnis liefern. Es ist wichtig, das Dokument nur als Chronik vergangener Rechnungen anzusehen, wenn man es strikt von oben nach unten ausgewertet hat. Ansonsten kann jedes Ergebnis aus Sicht des Anwenders „veraltet“ sein<sup>18</sup>.

Wenn eine Variable in ihre eigene Definition eingeht, wird sie auf der rechten Seite der Zuweisung ausgewertet, um ihren neuen Wert zu bestimmen:

```
> a := a + 1;
```

5 (20)

Natürlich muss  $a$  zu diesem Zeitpunkt schon initialisiert gewesen sein. Durch wiederholte Auswertung ändert sich ihr Wert fortlaufend:

```
> a := a + 1;
```

6 (20)

Als Variablennamen können auch beliebige Wörter verwendet werden:

```
> ergebnis :=  $\frac{a}{2}$ ;
```

3 (21)

Nach dieser einmaligen Auswertung ist der Wert von `ergebnis` unabhängig vom Wert von  $a$ , wie an folgendem Beispiel zu sehen ist:

```
> a := 10 : ergebnis;
```

3 (22)

<sup>18</sup>Man kann eine chronologische Auswertung erzwingen. In der Symbolleiste befindet sich ein Symbol mit drei Ausrufezeichen, das eine Ausführung des gesamten Dokuments anstößt. Anschließend hängt jedes Ergebnis garantiert nur von den Eingaben ab, die weiter oben im Dokument stehen.

## 1. Basiswissen

Eine dauerhafte Abhängigkeit wird hingegen durch unbestimmte Variablen modelliert:

```
> ergebnis := b2;
```

$$b^2 \quad (23)$$

Diese Variable „merkt“, wenn  $b$  ein Wert zugewiesen wurde:

```
> b := 5 : ergebnis;
```

$$25 \quad (24)$$

```
> b := 6 : ergebnis;
```

$$36 \quad (25)$$

<sup>19</sup>Für *unassign*('b') kann die Kurzschreibweise  $b := 'b'$  verwendet werden. Analog weist die Funktion *assign* eine Variable zu. Der  $:=$  Operator ist eine Kurzschreibweise dafür.

Die Definition von Variablen kann wieder aufgehoben werden<sup>19</sup>:

```
> unassign('b');
```

Anschließend ist auch *ergebnis* wieder unbestimmt:

```
> ergebnis;
```

$$b^2 \quad (26)$$

Unbestimmt zugewiesene Variablen verhalten sich also in etwa wie mathematische Funktionen. Maple versteht unter Funktionen allerdings noch etwas Anderes! Der Unterschied wird gleich diskutiert.

### Funktionen

*Maple-Funktionen* stellen wie mathematische Funktionen Vorschriften dar, aus Argumenten ein Ergebnis zu erzeugen. Sie bieten aber noch einige technische Kniffe, die sie sehr vielseitig einsetzbar machen. Man kann *Funktionen* als *Zuordnungen* oder *Prozeduren* definieren.

<sup>20</sup>Der Zuordnungspfeil  $\rightarrow$  kann als Zeichenkombination  $\rightarrow$  eingegeben werden und wird automatisch umgewandelt.

Eine *Zuordnung* beschreibt, welche Operationen auf eine Eingabe angewandt werden sollen. Man verwendet dazu die *Pfeil*<sup>20</sup>-Schreibweise:

```
> x  $\rightarrow$  x2;
```

$$x \rightarrow x^2 \quad (27)$$

In dieser isolierten Form besitzt die *Zuordnung* noch keinen Nutzen. Man muss sie in einen anderen Term einbetten. Am einfachsten ist es, die *Zuordnung* in einer *Variablen* zu hinterlegen:

```
> f := x  $\rightarrow$  x2 ;
```

Eine *Variable*, die eine *Zuordnung* enthält, nennt man *Funktion*. Angewendet wird sie mit der bekannten *Klammer-Schreibweise*:

```
> f(7);
```

49 (28)

Das Argument der *Zuweisung* (in diesem Beispiel die Variable  $x$ ) wird *gebunden*. Das bedeutet, zukünftige Auswertungen werden nicht von der Belegung dieser Variablen beeinflusst – sie hat nur als Platzhalter in der Definition gedient.

```
> x := 2 : f(7);
```

49 (29)

Insbesondere kann man auch bereits belegte Variablen zum Definieren einer Funktion verwenden, ohne dass sich die Bedeutung wie bei unbestimmt zugewiesenen Variablen ändern.

Man kann auch Funktionen mit mehreren Argumenten definieren, indem man die Argumentliste in Klammern setzt. Der Aufruf erfolgt intuitiv:

```
> g := (x, y) → x2 + y2 ;
```

```
> g(3, 4);
```

25 (30)

*Zuordnungen* können Maple-interne Kommandos enthalten, wodurch überraschende Definitionen möglich werden. Beispielsweise kann man seinen eigenen Hilfe-Aufruf schreiben:

```
> h := thema → help(thema) ;
```

```
> h(evalf);
```

Durch diesen Aufruf wird tatsächlich ein Hilfe-Fenster geöffnet.

Ebenso sind selbstmodifizierende Funktionen realisierbar:

```
> umdef := arg → assign('umdef', 'arg → arg2') ;
```

```
> umdef(10);
```

```
> umdef(10);
```

100 (31)

## 1. Basiswissen

Beim ersten Aufruf definiert diese Funktion sich selbst zur Quadratfunktion um (und generiert dabei keine Ausgabe). Beim zweiten Aufruf findet dann eine gewohnte Auswertung statt. Solche „Tricks“ sind natürlich im Allgemeinen nicht zielführend, aber sie erleichtern das Verständnis, wie Maple intern arbeitet.

Mit der Ergänzung von Variablen und Funktionen zu den bekannten Rechenregeln ergeben sich ein paar beliebte Fehlerquellen, auf die hier frühzeitig hingewiesen werden soll.

Maple achtet teils penibel auf das Setzen oder Weglassen von Leerzeichen um Variablen. Betrachte die Multiplikation zweier Variablen:

>  $a \cdot b$  :

Dieser Ausdruck bleibt zulässig, wenn man den Malpunkt weglässt, da Maple ein Leerzeichen zwischen zwei Variablen als Multiplikation auffasst:

>  $a b$  :

Man darf die Variablen allerdings nicht ohne Leerzeichen verbinden, da sie sonst als eine dritte Variable ab gedeutet werden. Problematischer wird der Fall, wenn Klammern in Ausdrücken vorkommen. Als Beispiel dient folgender korrekter Term:

>  $c \cdot (c + 1)$  :

Auch hier darf man den Malpunkt durch ein Leerzeichen ersetzen:

>  $c (c + 1)$  :

Im Gegensatz zur mathematischen Konvention muss das Leerzeichen aber zwingend gesetzt werden, und wird nicht etwa durch die Klammer impliziert:

>  $c(c + 1)$  :

Diesen Ausdruck liest Maple als den Aufruf einer Funktion  $c$  mit dem Argument  $c+1$ , was nicht der Intention entspricht. Auf der sicheren Seite ist man, wenn man grundsätzlich den Malpunkt zum Multiplizieren benutzt.

Das gleiche Problem tritt umgekehrt auf, wenn man fälschlicherweise ein Leerzeichen vor die öffnende Klammer eines Funktionsaufrufs setzt:

>  $\cos (Pi)$

Hier kommt Maple nicht wie erwartet zum Ergebnis  $-1$ , da es keinen Funktionsaufruf sieht. Durch das Leerzeichen wird der Ausdruck als Produkt von  $\cos$  und der Konstante  $Pi$  gedeutet und ist somit unbestimmt.



### Verzögerte Auswertung

In den letzten Beispielen wurde gelegentlich das einfache Anführungszeichen ' um Teilausdrücke gesetzt, ohne näher darauf einzugehen. Es handelt sich dabei um einen *Auswertungsverzögerer*. Manche Funktionen erwarten als Argument Terme eines bestimmten Typs, die normalerweise aber schon beim Übergeben ausgewertet werden. Ein schönes Beispiel tritt auf, wenn man in der Maple-Hilfe Informationen über Konstanten sucht:

```
> help(Digits);
```

Dieser Aufruf liefert nicht die erwartete Hilfe-Seite, da die Konstante Digits zu einer Zahl (z.B. 10) ausgewertet wird, bevor sie als Argument an die help-Funktion weitergereicht wird – help sucht also nach Informationen zum Thema 10. Es ist somit wünschenswert, Maple die Auswertungsreihenfolge vorzugeben.

Die Aufgabe des *Auswertungsverzögerers* ist es, bei einer Auswertung sein unverändertes Argument zurückzuliefern:

```
> 'Digits';
```

```
Digits (32)
```

Durch Schachtelung kann man Terme auch vor mehreren Auswertungen<sup>21</sup> schützen:

```
> "Digits";
```

```
'Digits' (33)
```

Rückblickend werden dadurch manche Beispiele klarer. Die Kurzschreibweise der unassign-Funktion lässt sich so verstehen, dass einer Variablen ihre eigene symbolische Form zugewiesen wird:

```
> x := 'x':
```

Ohne die Anführungszeichen würde man der Variablen ihren aktuellen Wert zuweisen, also praktisch gar nichts tun.

Im Beispiel der selbstmodifizierenden Funktion wurden gleich mehrere Auswertungen verzögert:

```
> umdef := arg → assign('umdef', 'arg → arg2):
```

Hier wird zum einen der Name der umdef-Variable und zum anderen die Roh-Beschreibung einer Zuordnung an die assign-Funktion übergeben. Man beachte, dass beim ersten Aufruf von umdef der Wert des Arguments arg keine Rolle spielt, da es nicht in die neue Zuweisung eingeht. Die Anführungszeichen verhindern hier ein vorzeitiges Einsetzen in das zweite Argument der assign-Funktion, was die Zuordnung ungültig machen würde.

<sup>22</sup>Das sind genau solche Aufrufe, die dem Lehrer zeigen, dass Schüler gerne zu den Worten beitragen. Das muss du einfach können, da hilft dir kein Taschenrechner.

<sup>21</sup>In diesem Beispiel stehen links und rechts von Digits jeweils zwei einfache Anführungszeichen, keine doppelten Anführungszeichen (Gänsefüßchen).

## 1.5. Symbolisches Rechnen

<sup>22</sup>Das sind genau solche Aufgaben, die Lehrer ihren Schülern gerne mit den Worten beibringen: „Das musst du einfach können, da hilft dir kein Taschenrechner.“

Auch wenn Computer-Algebra-Systeme sich hervorragend als fortgeschrittene Taschenrechner eignen, liegt ihre wirkliche Stärke darin, mit unbestimmten Variablen umgehen zu können. Maple ist in der Lage, Terme umzuformen und Gleichungen zu lösen<sup>22</sup>.

### Vereinfachen und Auswerten

In der Einleitung haben wir gesehen, dass Maple einige Ausdrücke automatisch vereinfacht, z.B. kürzt es Brüche. In anderen Fällen muss man Maple aktiv dazu „überreden“. Betrachte folgende Rechnung:

```
> sqrt(a^2);
```

$$\sqrt{a^2} \quad (34)$$

Die Wurzel des Quadrates wird unverändert ausgegeben, obwohl sich der Term scheinbar weiter vereinfachen lässt. Der Grund ist, dass das Auflösen einer Wurzel keine rein arithmetische Umformung darstellt. Um hier weiterzukommen, müssen wir entweder vereinfachende Annahmen machen, oder Maple erlauben, Vereinfachungsregeln auf Funktionsebene anzuwenden.

Eine Annahme kann mit der `assume`-Funktion getroffen werden:

```
> assume(a > 0) : sqrt(a^2);
```

$$a \sim \quad (35)$$

Maple weiß nun im voraus, dass  $a$  für eine positive Variable steht. Daher wird ihr Wert durch das Quadrieren und Wurzelziehen nicht verändert, und die Funktionen können entfernt werden. Maple gibt aber nicht die Variable  $a$ , sondern eine mit dem Tilde-Symbol  $\sim$  versehene neue Variable  $a \sim$  zurück. Damit soll ausgedrückt werden, dass es sich nicht um die ursprüngliche Variable  $a$  handelt, denn es wurden spezialisierende Annahmen (nämlich `assume(a>0)`) gemacht.

Annahmen, die mit `assume` getroffen werden, hält Maple über die gesamte Sitzung fest. Die Variable  $a$  im Beispiel ist dauerhaft an die Annahme gebunden. Um sie wieder als normale (nicht nur positive) Variable zu verwenden, muss man die Definition aufheben (`a := 'a' ;`) oder Maple neu starten (`restart ;`). Weiß man im voraus, dass die Annahme nur für eine einzelne Auswertung getroffen werden muss, kann man stattdessen die `assuming`-Eigenschaft verwenden.

Eine *Eigenschaft* besteht aus einem Schlüsselwort und möglicherweise zugehörigen Termen, die an einen Ausdruck angehängt werden, um diesen näher zu spezifizieren.

## 1.5. Symbolisches Rechnen

Der Aufruf erfolgt beispielsweise so:

```
> sqrt(a^2) assuming a < 0;
```

$$-a \quad (36)$$

```
> sqrt(a^2);
```

$$\sqrt{a^2} \quad (37)$$

Die *assuming*-Eigenschaft hat den gleichen Zweck wie die *assume*-Funktion, aber sie wirkt nur lokal. Das Ergebnis der ersten Zeile enthält keine Tilde, da die Variable *a* nicht dauerhaft beeinflusst wurde. Der Ausdruck in der zweiten Zeile wird ohne Annahmen an *a* ausgewertet (und deshalb nicht weiter vereinfacht).

Eigenschaften müssen nicht am Ende eines Terms stehen. Verschachtelt man sie, muss man aber genau auf ihren Gültigkeitsbereich achten:

```
> sqrt(a^2 assuming a < 0);
```

$$\sqrt{a^2} \quad (38)$$

Dieser Ausdruck kann nicht vereinfacht werden, da die Annahme  $a < 0$  nur in die Berechnung des Quadrates eingeht und anschließend vergessen wird. (Sie hatte also keine Auswirkungen.) Die anschließende Wurzelfunktion weiß nichts mehr davon und das ursprüngliche Problem tritt wieder auf.

Wenn man keine Annahmen über die Variablen treffen will oder kann, muss man Kenntnisse über die beteiligten Funktionen einbringen:

```
> simplify(sqrt(a^2));
```

$$\operatorname{csgn}(a) a \quad (39)$$

Die Funktion *simplify* wendet eine Reihe von Vereinfachungsregeln auf den Ausdruck an. Das Ergebnis ist in diesem Fall ein Produkt aus *a* und seiner komplexen Signum-Funktion *csgn(a)*. Maple geht offenbar implizit davon aus, dass die Variable eine beliebige komplexe Zahl sein kann.

Man kann *annehmen*, dass es sich bei *a* um eine reelle Zahl<sup>23</sup> handelt:

```
> simplify(sqrt(a^2)) assuming a :: real;
```

$$|a| \quad (40)$$

Dieses Resultat hat man vermutlich schon eine Weile erwartet. Maple ist beim Umformen von Ausdrücken sehr gründlich. Dieses Verhalten kann gelegentlich aber allerdings übervorsichtig oder umständlich wirken, wenn man nicht weiß, worauf man achten muss.

<sup>23</sup>Der Vierpunkt-Operator `::` erfüllt eine Vielzahl von kontextabhängigen Aufgaben in Maple. Hier wird er genutzt, um Variablen einen Typ zuzuweisen.

## 1. Basiswissen

Analog zur `simplify`-Funktion zum Vereinfachen von Termen ist `eval` die Standard-Funktion um Auswertungen zu steuern. Man betrachte ein Polynom in der Variablen  $x$ :

$$> x^3 + x^2 + x;$$

Um dieses Polynom an einer Stelle auszuwerten, könnte man die Variable temporär mit einem Wert belegen:

$$> x := 2 : x^3 + x^2 + x; x := 'x' : \quad 14 \quad (41)$$

Alternativ könnte man temporär eine Funktion definieren und ihr die Stelle als Argument übergeben:

$$> f := x \rightarrow x^3 + x^2 + x : f(2); f := 'f' : \quad 14 \quad (42)$$

Mit beiden muss man offenbar am Ende Variablen freigeben, wenn man die Maple-Session nicht dauerhaft „zumüllen“ möchte.

Eine bessere Lösung ist es, den Ausdruck auszuwerten, ohne temporäre Variablen zu belegen:

$$> eval(x^3 + x^2 + x, x = 2); \quad 14 \quad (43)$$

<sup>24</sup>Oder ein Gleichungssystem, mehr dazu im nächsten Kapitel.

Die Funktion erwartet als erstes Argument einen Term, der ausgewertet werden soll, und als zweites Argument eine Gleichung<sup>24</sup>, die der Variablen einen Wert zuweist.

Beachte den Unterschied zwischen der `eval`-Funktion und der `assuming`-Eigenschaft. Auch wenn sie zunächst ähnlich eingesetzt werden, führen nicht immer beide zum Ziel:

$$> x^3 + x^2 + x \text{ assuming } x = 2; \quad x^3 + x^2 + x \quad (44)$$

Das Polynom wurde nicht ausgewertet, da `assuming` nur Annahmen zur symbolischen Vereinfachung vorgibt. Das Polynom lässt sich aber auch unter der Annahme, dass die Variable einen bestimmten Wert annimmt, nicht einfacher schreiben! *Termumformungen* (insbesondere *Vereinfachungen*) und *Auswertungen* sind Maple-intern verschiedene Dinge.

Die `eval`-Funktion kann auch bei rein symbolischem Rechnen nützlich sein. Man betrachte eine Abfolge von sich gegenseitig beeinflussenden Variablendefinitionen:

$$> m := n + 1 : n := 2 : o := p - 1 :$$

## 1.5. Symbolisches Rechnen

Wenn man durch einen einfachen Aufruf eine symbolische Auswertung anstößt, arbeitet Maple den Ausdruck vollständig rekursiv<sup>25</sup> ab:

$$m; \quad (45)$$

dabei wurden sowohl Variablensubstitutionen als auch Termumformungen ausgeführt. Möchte man die Substitutionen einzeln nachvollziehen, kann man der eval-Funktion den gewünschten Fortschritt der Auswertung als Argument übergeben:

$$\text{eval}(m, 1); \quad (46)$$

$$\text{eval}(m, 2); \quad (47)$$

$$\text{eval}(m, 3); \quad (48)$$

Dass jeder Ausdruck in eine (einfache) Normalform gebracht wird, hängt mit der Arbeitsweise von Maple zusammen. Im folgenden Abschnitt wird beschrieben, wie man die Gestalt von Termen beeinflusst.

### Termumformung

Die Algorithmen, die hinter den Vereinfachungen und Auswertungen in Maple stehen, bringen alle Ausdrücke in eine sogenannte *Normalform*. Diese Form ist so gewählt, dass ein Mensch sie in der Regel als „einfach“ oder „ordentlich“ empfindet. Gerade bei Ausdrücken, für die es verschiedene vorteilhafte Darstellungen gibt, ist dieses Vorgehen nicht immer ausreichend, sodass man als Anwender selbst eine Form vorgeben möchte.

Für die Formatierung von Polynomen kann man die Funktionen `factor` und `expand` verwenden. Die erste bringt das Polynom in eine multiplikative Normalform, die zweite in eine additive Normalform:

$$p := x \cdot (x^2 + x); \quad (49)$$

<sup>25</sup>Das Konzept der Rekursion wird später noch ausführlich aufgegriffen, beispielsweise im Kapitel über Programmierung.