



Einführung in Mathematica 3.0

1	EINLEITUNG	5
1.1	Was ist Mathematica?	5
1.2	Die Bestandteile von Mathematica	6
1.3	Mathematica 3.0 an der Universität Trier	6
1.3.1	Mathematica 3.0 unter Windows NT	6
1.3.2	Mathematica 3.0 unter UNIX	6
2	DIE MATHEMATICA-BENUTZEROBERFLÄCHE: NOTIZBÜCHER UND ZELLEN	7
2.1	Elementare Operationen mit Notizbuch-Zellen	7
2.1.1	Mathematica-Ausdrücke schreiben und auswerten lassen (Input- und Output-Zellen)	7
2.1.2	Zellen für die weitere Bearbeitung markieren	8
2.1.3	Zellinhalte verschieben, kopieren oder löschen	8
2.2	Notizbücher neu erstellen, sichern, öffnen oder schließen	8
2.3	Notizbücher drucken	9

3	ARBEITEN MIT MATHEMATICA	9
3.1	Der Help-Browser	9
3.2	Numerische Berechnungen	10
3.2.1	Arithmetische Operationen	10
3.2.2	Exakte und angenäherte Ergebnisse	11
3.2.3	Mathematische Funktionen	12
3.2.4	Konstanten	13
3.2.5	Komplexe Zahlen	13
3.3	Wichtige Bestandteile und Regeln der Mathematica-Sprache	14
3.3.1	Rückgriff auf vorangegangene Ausgaben	14
3.3.2	Definition von Variablen	14
3.3.3	Listen von Objekten	15
3.4	Mathematica-Pakete	15
3.5	Algebraische Manipulationen	16
3.6	Mathematische Operationen mit Symbolen	17
3.6.1	Differenzieren	17
3.6.1.1	Partielle Ableitung (Operator D)	17
3.6.1.2	Totale Ableitung (Operator Dt)	17
3.6.1.3	Ableiten unbekannter Funktionen	18
3.6.2	Integrieren	18
3.6.2.1	Unbestimmtes Integral	18
3.6.2.2	Bestimmtes Integral	19
3.6.2.3	Numerische Lösungen	19
3.6.3	Summen und Produkte	19
3.6.3.1	Summe $\sum_{i=i_{\min}}^{i_{\max}} f$	19
3.6.3.2	Doppelsumme $\sum_{i=i_{\min}}^{i_{\max}} \sum_{j=j_{\min}}^{j_{\max}} f$	20
3.6.3.3	Produkt $\prod_{i=i_{\min}}^{i_{\max}} f$	20
3.6.4	Ausdrücke auf Gleichheit testen	21
3.6.5	Relationale und logische Operatoren	21
3.6.6	Gleichungen lösen	22
3.6.6.1	Algebraische Gleichungen in einer Variablen	22
3.6.6.2	Transzendente Gleichungen in einer Variablen	23
3.6.6.3	Gleichungssysteme mit mehreren Veränderlichen	23
3.6.6.4	Vollständige Lösungen	24
3.6.7	Differentialgleichungen	25
3.6.8	Potenzreihen (Taylor-Reihen)	26
3.6.9	Grenzwerte	26
3.7	Eingabe in mathematischer Notation, Arbeiten mit Paletten	26
3.7.1	Eingabe in mathematischer Notation mit Hilfe von Paletten	27
3.7.2	Zwei Varianten der mathematischen Notation: Standardform und traditionelle Form	28
3.7.3	Eingabe in mathematischer Notation mit Hilfe der Tastatur	29

3.8	Grafiken	30
3.8.1	Elementare Diagramme	30
3.8.2	Optionen	31
3.8.3	Größe einer Grafik verändern	31
3.8.4	Grafiken verschieben	31
3.8.5	Grafiken (in andere Anwendungen) übertragen	32
3.8.5.1	Übertragung via Zwischenablage	32
3.8.5.2	Export in eine Datei	32
3.8.6	Diagramme in Variablen speichern, abrufen und modifizieren	32
3.8.7	Diagramme kombinieren	33
3.8.8	Konturdiagramme	34
3.8.9	Dichtediagramme	35
3.8.10	3D-Oberflächendiagramme	36
3.8.11	3D-Aussichtspunkt festlegen	36
3.8.12	Gestaltung der Beschriftungen in Grafiken	37
3.8.13	Parametrische Diagramme	39
3.8.14	Datenlisten zeichnen	40
3.8.15	Grafikanweisungen	40
3.9	Numerische Mathematik	41
3.9.1	Exakte Ergebnisse und numerische Approximationen	41
3.9.2	Numerische Lösung von Gleichungen	42
3.9.2.1	Polynom-Gleichungen	42
3.9.2.2	Transzendente Gleichungen	42
3.9.3	Numerische Optimierung	43
3.10	Funktionen und Programme	44
3.10.1	Funktionen definieren und verwenden	44
3.10.2	Bedingte Anweisungen	45
3.10.3	Schleifen-Konstruktionen	46
3.10.4	Ausgeben von Werten	46
3.11	Listen und Matrizen	47
3.11.1	Erstellen von Wertetabellen	47
3.11.2	Vektoren und Matrizen	48
3.11.2.1	Operationen mit Matrizen	49
3.11.2.2	Lösung linearer Gleichungssysteme in Matrizenschreibweise	52
3.11.3	Weitere Operationen mit Listen	53
3.12	Verwendung externer Dateien	53
3.12.1	Ergebnisse und Definitionen in externe Dateien schreiben	53
3.12.1.1	Ausgabe im Mathematica-Format	54
3.12.1.2	Ausgabe im Textformat	55
3.12.1.3	Weitere Export- bzw. Publikationsoptionen	55
3.12.2	Aus externen Dateien lesen	56
3.12.2.1	Mathematica-Ausdrücke einlesen	56
3.12.2.2	Aus Textdateien lesen	57

4	ZUSÄTZLICHE HINWEISE ZUR NOTIZBUCH-BENUTZEROBERFLÄCHE	57
4.1	Hilfen beim Erstellen und Ausführen von Mathematica-Ausdrücken	57
4.1.1	Funktionsbezeichnungen automatisch vervollständigen lassen	57
4.1.2	Mathematica unterbrechen	58
4.2	Zellenattribute und -stile	58
4.2.1	Zellenattribute	58
4.2.1.1	Editable	58
4.2.1.2	Initialization	58
4.2.2	Zellenstile	60
4.2.3	Style Sheets	60
4.3	Verfahren für Gruppen von Zellen	61
4.3.1	Zellen zu einer Gruppe zusammenfassen	61
4.3.2	Eine Gruppierung aufheben	61
4.3.3	Zellengruppen schließen und öffnen	62
4.3.4	Zellen aufteilen	62
4.3.5	Zellen zusammenlegen	62
5	LITERATUR UND SONSTIGE INFORMATIONSQUELLEN	63
6	ANHANG	64
6.1	X-Mathematica auf der RZSUN00 benutzen	64
6.1.1	Voraussetzungen für das Arbeiten mit Mathematica auf der RZSUN00	64
6.1.1.1	EDV-Kenntnisse	64
6.1.1.2	Benutzerverwaltung	64
6.1.1.3	Anforderungen an Ihren Arbeitsplatzrechner	64
6.1.2	Eine Sitzung mit X-Mathematica auf der RZSUN00	65
6.1.2.1	Vorbereitungen auf dem UNIX-Arbeitsplatzrechner	65
6.1.2.2	Mathematica auf der RZSUN00 starten und beenden	65
6.1.2.3	Verhalten nach dem Beenden von Mathematica	66
6.2	Wichtige Tastaturbelegungen	66
7	STICHWORTVERZEICHNIS	67

Herausgeber: Universitäts-Rechenzentrum Trier
 Universitätsring 15
 D-54286 Trier
 Tel.: (0651) 201-3417, Fax.: (0651) 3921
Leiter: Prof. Dr.-Ing. Manfred Paul
Redaktion: Bernhard Baltes-Götz
Copyright © 1999; URT

1 Einleitung

Dieses Manuskript gibt eine kurze Einführung in die Verwendung von Mathematica 3.0 an der Universität Trier. Es basiert im wesentlichen auf dem mit „A Practical Introduction to Mathematica“ betitelten ersten Teil des Originalhandbuchs (Wolfram 1996), bietet also nur einen ersten und in vielerlei Hinsicht recht oberflächlichen Eindruck von Mathematica 3.0.

Im Manuskript wird primär die MS-Windows-Version 3.0.1 von Mathematica behandelt. Jedoch werden auch die weitgehend analog zu bedienenden 3.0-Versionen für UNIX und Mac-OS berücksichtigt.

1.1 Was ist Mathematica?

Mathematica kann (u.a.) folgende Aufgaben erledigen:

- Numerische Berechnungen
Es stehen mehrere tausend eingebaute Funktionen zur Verfügung, wobei der Benutzer jede beliebige Präzision wählen kann.
- Symbolische Berechnungen
Mathematica beherrscht u.a. die Differential- und Integralrechnung, kann also z.B. Stammfunktionen symbolisch angeben.
- Graphische Darstellungen
Mathematica stellt in zwei- oder dreidimensionalen Grafiken mathematische Funktionen oder Datenlisten dar.

Alle Anweisungen an Mathematica werden in einer einheitlichen Syntax formuliert. Das folgende Fenster zeigt Beispiele zu den drei Aufgabenarten:

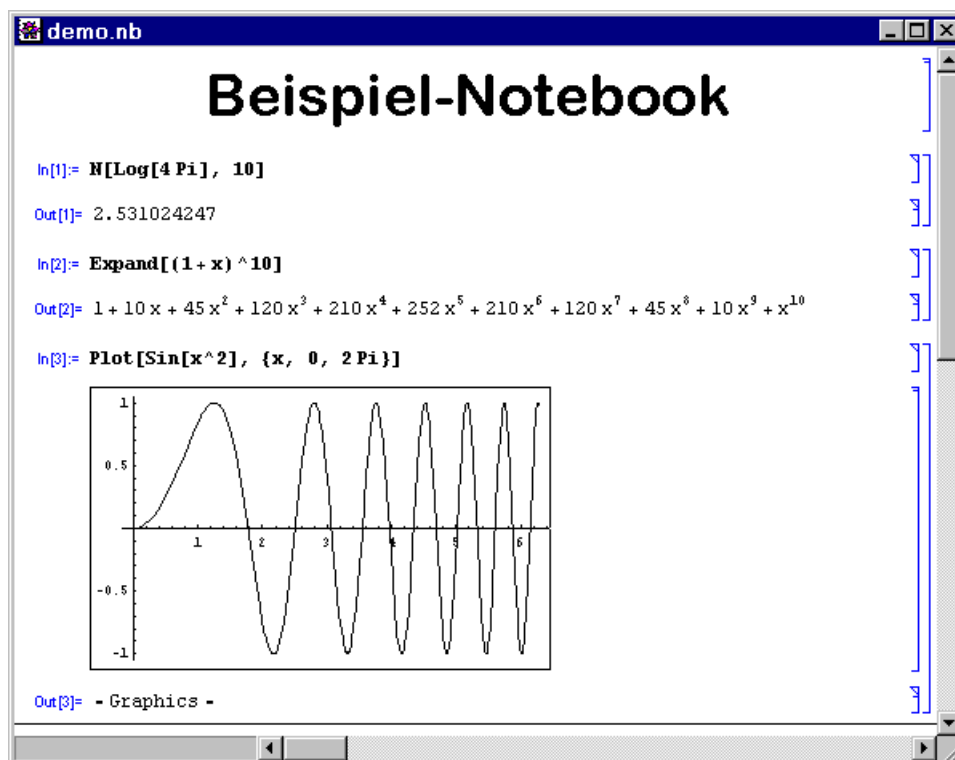


Abbildung 1 Notebook-Fenster unter MS-Windows

Wie dieses Beispiel mit Benutzereingaben *In[n]* und Programmreaktionen *Out[n]* zeigt, kann man mit Mathematica gut interagieren, wenn man seine Sprache beherrscht. In der ersten Eingabe wird z.B. die Berechnung von $\log(4\pi)$ mit 10-stelliger Genauigkeit angefordert.

1.2 Die Bestandteile von Mathematica

Die beiden wichtigsten Bestandteile von Mathematica sind:

- **Benutzeroberfläche** (engl.: **Frontend**)
Die Benutzerschnittstelle ist im Prinzip betriebssystemspezifisch, sieht jedoch unter allen Betriebssystemen mit graphischer Oberfläche (z.B. MS-Windows, UNIX mit X Window, Mac-OS) recht ähnlich aus. Das Frontend nimmt die Anweisungen des Benutzers entgegen, übergibt die auswertenden Mathematica-Ausdrücke (Input) an den Kernel, und bereitet dessen Output für den Benutzer auf.
- **Kern** (engl.: **Kernel**)
Er ist plattformunabhängig, führt alle mathematischen Operationen aus, versteht die Mathematica-Syntax.

Benutzeroberfläche und Kern sind separate Prozesse, die sogar auf verschiedenen Computern ablaufen können. Beide kommunizieren über die Programmschnittstelle **Mathlink** miteinander, die auch dazu benutzt werden kann, von Mathematica aus Mathlink-kompatible andere Programme zu steuern, bzw. Mathematica von solchen Programmen aus (als Unterprogramm) zu verwenden.

Der Kernprozeß wird normalerweise erst bei Anforderung der ersten Berechnung gestartet.

Eine wichtige Methode zur individuellen Erweiterung des Mathematica-Systems stellen die **Pakete** (engl.: packages) mit zusätzlichen oder modifizierten Funktionsdefinitionen dar, von denen viele (die sogenannten Standard Packages, siehe Martin 1996) von Mathematica gleich mitgeliefert werden. So findet sich etwa die Cholesky-Zerlegung im Zusatzpaket **LinearAlgebra**.

1.3 Mathematica 3.0 an der Universität Trier

Öffentlich zugängliche Mathematica-Installationen sind an der Universität Trier für MS-Windows und UNIX vorhanden.

1.3.1 Mathematica 3.0 unter Windows NT

Mathematica 3.0 für Windows NT steht Benutzern der NT-Domäne URT auf Pool- und Büro-PCs zur Verfügung.

Mathematica 3.0 auf Pool-PCs unter Windows NT

In den vom Rechenzentrum betreuten PC-Pools unter Windows NT wird Mathematica 3.0 folgendermaßen gestartet:

**Start > Programme > Wissenschaftliche Programme >
Mathematica vom NT-Server des URT > Mathematica 3.0**

Wer Mathematica häufiger benutzt, wird sich vielleicht zum bequemerem Starten ein Symbol auf den individuell konfigurierbaren Desktop legen.

Mathematica 3.0 auf vernetzten Büro-PCs unter Windows NT

Sie können Mathematica 3.0 auf Ihrem Büro-PC unter Windows NT verwenden und dabei ohne eigenen Kostenbeitrag von der Mehrfachlizenz des Rechenzentrums profitieren. Nähere Informationen erhalten Sie in der Benutzerberatung.

1.3.2 Mathematica 3.0 unter UNIX

Im Anhang wird beschrieben, wie Sie mit der auf dem UNIX-Rechner RZSUN00 installierten X-Version von Mathematica 3.0 arbeiten können. Dies ist vor allem dann interessant, wenn Ihr Arbeitsplatzrechner ebenfalls unter dem Betriebssystem UNIX läuft.

2 Die Mathematica-Benutzeroberfläche: Notizbücher und Zellen

Die mit der Standard-Oberfläche von Mathematica erstellten Dokumente werden als **Notizbücher** (engl.: **Notebooks**) bezeichnet. Wesentliche Bestandteile eines Notizbuchs sind:

- erklärenden Text und Überschriften
- Grafiken
- Mathematica-Input und -Output

Ein Beispiel für ein Notizbuch haben Sie schon in Abbildung 1 gesehen. Man kann ein Notizbuch analog zu einem Dokument eines Textverarbeitungsprogramms sichern, neu erstellen, öffnen, schließen, editieren und drucken. Für Dateien mit Mathematica-Notizbüchern wird in der Regel die Namensweiterung „.nb“ verwendet.

Die Notizbuchinhalte sind in **Zellen** organisiert, deren Umfang am rechten Rand des Notebook-Fensters durch Klammern angezeigt wird (siehe Abbildung 1). Die Zellenstruktur eines Notebooks kann **hierarchisch** sein, was durch Mehrebenen-Anordnung der Zellenklammern angezeigt wird.

Ein Notizbuch kann z.B. die Beschreibung einer mathematischen Theorie mit erläuternden Grafiken und ausführbaren Anweisungen enthalten, so daß die Leser(innen) neu erworbenes Wissen sofort ausprobieren können. Damit eignen sich Mathematica-Notizbücher gut für Ausbildungszwecke. Man kann sie auch über das WWW verbreiten (siehe Abschnitt 3.12.1.3).

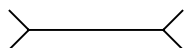
2.1 Elementare Operationen mit Notizbuch-Zellen

Sie können die folgenden Operationen mit den Zellen ausführen:

2.1.1 Mathematica-Ausdrücke schreiben und auswerten lassen (Input- und Output-Zellen)

In einem neuen Notizbuch befindet sich am Anfang eine waagerechte Linie als Zell-Einfügemarke. Sobald Sie Text eintragen, wird für diesen eine Input-Zelle eingerichtet, und die Zell-Einfügemarke verschwindet.

Eine Zelle darf mehrere Zeilen umfassen, die jeweils mehrere Ausdrücke enthalten dürfen. In einem bereits bearbeiteten Notizbuch können Sie die Zell-Einfügemarke per Maus oder mit den vertikalen Pfeiltasten an eine zulässige Stelle setzen. In Abbildung 1 sehen Sie die Zell-Einfügemarke am unteren Fenster- rand. Über zulässigen Stellen sieht der Mauszeiger folgendermaßen aus:



Wenn Sie nach dem Setzen der Zell-Einfügemarke Text eintippen, verschwindet die Zell-Einfügemarke und an ihrer Stelle wird eine neue **Input**-Zelle eingerichtet.

Die Klammer am rechten Rand einer Input-Zelle sieht folgendermaßen aus:



Um den Ausdruck in einer Zelle als Mathematica-Input auswerten zu lassen, muß diese markiert sein (s.u.), oder die Schreibmarke (|) muß sich darin befinden. Die Auswertung wird dann angefordert mit:

<Shift><Return>

Daraufhin erhält die Input-Zelle eine Beschriftung mit fortlaufender Nummer (siehe oben).

Kann der Ausdruck ausgewertet werden, so produziert Mathematica eine (ebenfalls beschriftete und nummerierte) **Output**-Zelle, die unmittelbar hinter der ausgewerteten Input-Zelle erscheint.

Die Klammer zur Output-Zelle unterscheidet sich von der Input-Klammer durch einen zusätzlichen Querstrich:



Per Voreinstellung bilden die zu einer Anforderung gehörenden Zellen eine *Gruppe*, was durch eine gemeinsame Klammer angezeigt wird.

Mathematica numeriert alle ausgewerteten Input-Zellen nach Auswertungsreihenfolge und alle Output-Zellen nach Produktionsreihenfolge durch, so daß die Numerierung nicht mit der räumlichen Anordnung korrespondieren muß.

Über den Menübefehl **Kernel > Show In/Out Names** lassen sich die Zellennummern ab- bzw. anschalten.

Während der Kernel-Aktivität können Sie mit dem Frontend weiterarbeiten, allerdings keine weiteren Kernel-Kommandos ausführen lassen. Außerdem können Sie die gerade auszuwertenden Input-Zellen nicht verändern. Diese Sperre wird durch folgende Doppelklammer angezeigt:



Bei der ersten Auswertungsanforderung in einer Mathematica-Sitzung wird der Kernel geladen, so daß eine längere Wartezeit entsteht.

2.1.2 Zellen für die weitere Bearbeitung markieren

- Eine einzelne Zelle oder Zellengruppe wird per Mausklick auf ihre Klammer markiert. Über markierbaren Klammern hat der Mauszeiger folgende Form:



- Zur Markierung einer Serie klickt man auf das erste Element und zieht die Maus dann bei gedrückter linker Maustaste bis zum letzten Element.
- Mit **Edit > Select All** können Sie *alle* Zellen eines Notizbuchs markieren.

2.1.3 Zellinhalte verschieben, kopieren oder löschen

Markieren Sie alle Zellen bzw. Zellengruppen und benutzen Sie die Befehle des Betriebssystems zum Ausschneiden, Kopieren und Einfügen. Die Zell-Einfügemarke (s.o.) können Sie per Mausklick oder mit den vertikalen Pfeiltasten an die gewünschte (und zulässige) Stelle im Notizbuch setzen.

Um den in einem Notizbuch zuletzt ausgewerteten Ausdruck unter die Einfügemarke zu kopieren, können Sie den Menübefehl **Input > Copy Input From Above** bzw. die folgende Tastenkombination benutzen:

Windows	Macintosh	X
<Strg><L>	<Befehl><L>	<Mod1><L>

Wenn Sie Mathematica-Ausgaben zu einer neuen Eingabe verarbeiten wollen, können Sie den Menübefehl **Input > Copy Output From Above** bzw. die folgende Tastenkombination benutzen:

Windows	Macintosh	X
<Shift><Strg><L>	<Shift><Befehl><L>	<Shift><Mod1><L>

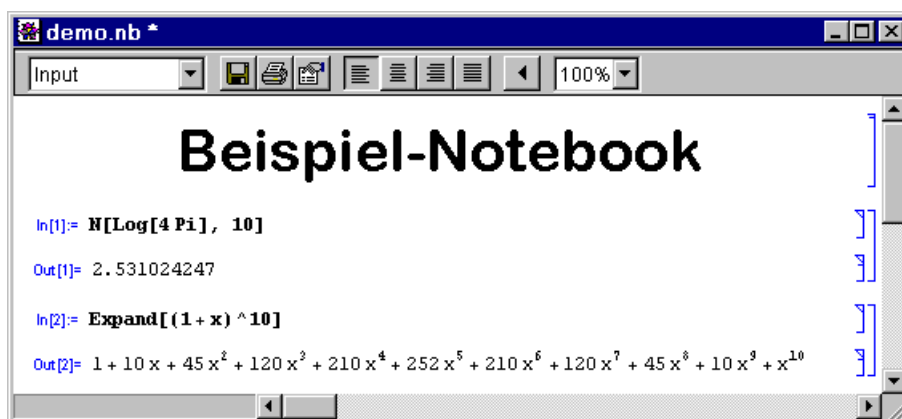
2.2 Notizbücher neu erstellen, sichern, öffnen oder schließen

In Mathematica stehen die üblichen Kommandos zum Verwalten von Dokumenten zur Verfügung:

Funktion	Menübefehl	Tastenkombination		
		Win <Strg>+	Mac <Befehl>+	X <Mod1>+
Neu erstellen	File > New	<N>	<N>	<N>
Sichern	File > Save	<S>	<S>	<S>
Sichern als	File > Save as ...	<Shift>+<S>	<Shift>+<S>	<Shift>+<S>
Öffnen	File > Open	<O>	<O>	<O>
Schließen	File > Close	<F4>	<W>	<W>

Die voreingestellte Namenserverweiterung für Notizbuch-Dateien lautet: ".nb".

Wer das Sichern per Mausklick erledigen möchte, muß mit **Format > Show ToolBar** die Symbolleiste zum Notebook-Fenster einschalten, z.B.:



Das Öffnen der zuletzt bearbeiteten Notizbücher geht am schnellsten über den Menübefehl **File > Notebooks**.

2.3 Notizbücher drucken

Um das gesamte Notizbuch oder aber die momentane Auswahl zu drucken, können Sie den Menübefehl **File > Print ...** oder die folgende Tastenkombination verwenden:

Drucken	Windows	Macintosh	X
Alles	<Strg><P>	<Befehl><P>	<Mod1><P>
Auswahl	<Shift><Strg><P>	<Shift><Befehl><P>	<Shift><Mod1><P>

Über den Menübefehl **File > Printing Settings** sind unter Windows und Mac-OS zahlreiche Optionen zur Gestaltung der Ausgabe verfügbar (z.B. Ränder, Kopf- und Fußzeilen). Bei der UNIX-Version sind die Menübefehle **File > Printing Options...** sowie **File > Headers and Footers...** zuständig.

3 Arbeiten mit Mathematica

3.1 Der Help-Browser

Bei der Arbeit mit Mathematica sollten Sie unbedingt den mit **Help > Help** oder **<F1>** erreichbaren **Help Browser** nutzen, der u.a. über die Option **Buid-in Functions** umfassende Informationen zum Kernel und zum Frontend bereithält und über die Option **Getting Started/Demos** einen guten Einblick in das Mathematica-System bietet:

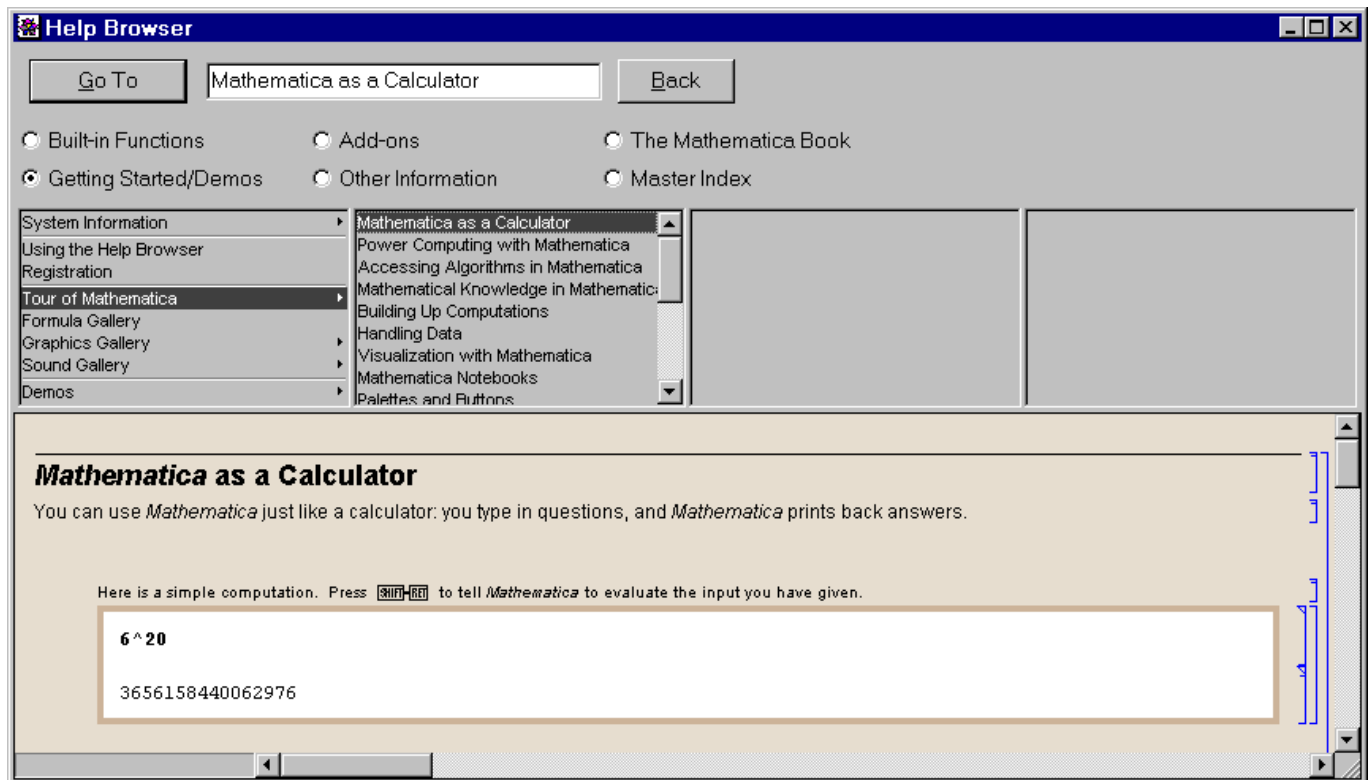


Abbildung 2 Der Help Browser bietet u.a. eine Tour durch die Welt von Mathematica

Außerdem enthält der Help Browser eine komplette elektronische Version der in Buchform erhältlichen Originaldokumentation zu Mathematica:

- Das Mathematica-Buch von Wolfram (1996) finden Sie über die gleichnamige Option im Kopfbereich des Help Browsers.
- Das Handbuch von Martin (1996) zu den Standard-Erweiterungspaketen finden Sie über **Add-ons > Standard Packages**.
- Über **Getting Started/Demos > System Information** finden Sie Beschreibungen der Plattformspezifika unter Windows, Mac-OS, UNIX und NeXT.

3.2 Numerische Berechnungen

In diesem Abschnitt wird beschrieben, wie Sie Mathematica als exzellenten **Taschenrechner** benutzen können.

3.2.1 Arithmetische Operationen

Es stehen die üblichen arithmetischen Operationen zur Verfügung:

Operation	Eingabe
Addition	$x + y$
Minus	$-x$
Subtraktion	$x - y$
Multiplikation	$x y$ oder $x * y$
Division	x / y
Potenzieren	$x ^ y$

Beispiel:

In[1]:= $(3+4)^2 - 2(3+1)$

Out[1]= 41

Die Auswertungsreihenfolge entspricht den üblichen Konventionen und kann nötigenfalls mit **runden** Klammern gesteuert werden.

Bei der **Schreibweise** ist zu beachten:

- Bei der Multiplikation darf die übliche mathematische Schreibweise ohne Operatorzeichen verwendet werden. Im allgemeinen **muß** dabei ein Leerzeichen zwischen den Faktoren stehen. Z.B. wird "hi" von Mathematica *nicht* als Produkt aufgefaßt, sondern als neues Symbol. "3i" wird jedoch richtig verstanden. Es darf auch ein Stern zwischen die Faktoren gesetzt werden (z.B. h*i).
- Als Dezimaltrennzeichen ist der **Punkt** zu verwenden.
- Wissenschaftliche Exponentialschreibweise ist erlaubt, z.B.:

```
In[1]:= 2.3 10^-70
```

```
Out[1]= 2.3 × 10-70
```

3.2.2 Exakte und angenäherte Ergebnisse

Während Taschenrechner mit begrenzter Genauigkeit arbeiten, liefert Mathematica nach Möglichkeit exakte Ergebnisse, z.B.:

```
In[1]:= 2^100
```

```
Out[1]= 1267650600228229401496703205376
```

Wenn Sie ein *Näherungsergebnis* wollen, müssen Sie die Funktion **N** (für: "**N**umeric") auf den Ausdruck anwenden:

```
In[1]:= N[2^100]
```

```
Out[1]= 1.26765 × 1030
```

Für die Funktion **N** gibt es noch eine alternative Postfix-Schreibweise:

```
In[1]:= 2^100//N
```

Wenn Sie eine ganze Zahlen eingeben, hält Mathematica diese für exakt und rechnet entsprechend exakt damit, z.B.:

```
In[1]:= 452/62
```

```
Out[1]=  $\frac{226}{31}$ 
```

Wenn Sie eine Zahl mit Dezimalpunkt eingeben, hält Mathematica diese für gerundet. Enthält in einem Ausdruck irgendeine Zahl einen Dezimalpunkt, wird das Ergebnis des Ausdrucks numerisch ermittelt, z.B.:

```
In[1]:= 452.0/62
```

```
Out[1]= 7.29032
```

Sie können beim Anfordern einer Näherungslösung optional die **Genauigkeit** bestimmen:

```
In[1]:= N[452/62, 10]
```

```
Out[1]= 7.290322581
```

3.2.3 Mathematische Funktionen

In Mathematica-Ausdrücken können mathematische Funktionen verwendet werden, z.B.:

```
In[1]:= Log[1]
```

```
Out[1]= 0
```

Bei der Suche nach einer bestimmten Funktion werden Sie vom **Help Browser** unterstützt, der mit dem Menübefehl **Help > Help** oder mit der Taste **<F1>** gestartet werden kann, z.B.:

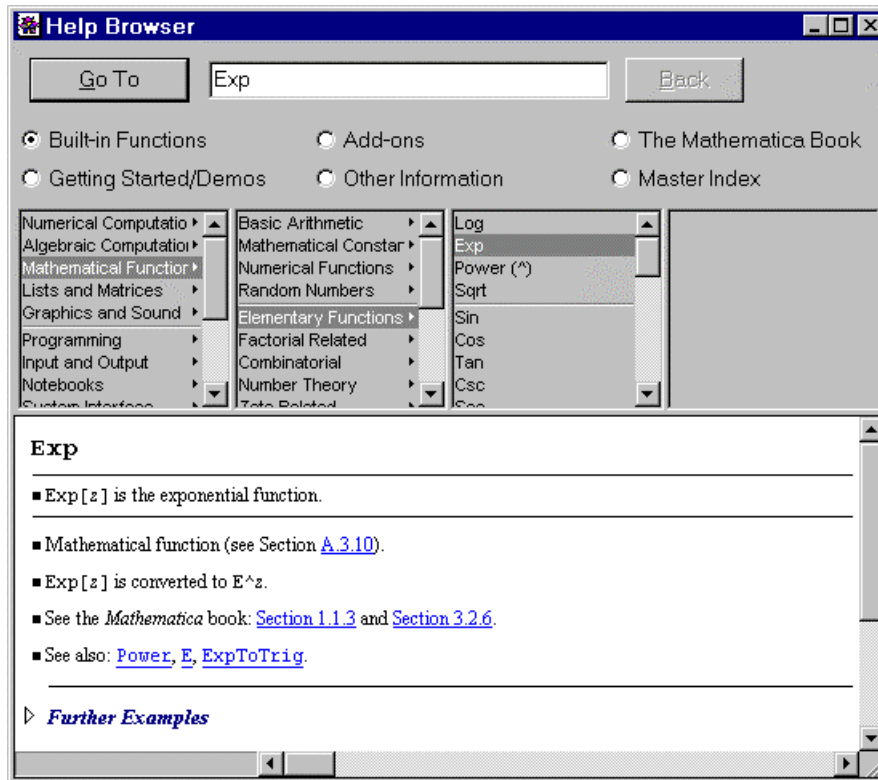


Abbildung 3 Der Help Browser informiert umfassend über mathematische Funktionen

Bei der **Schreibweise von Funktionsaufrufen** ist zu beachten:

- Die Argumente werden in **eckige** Klammern gesetzt.
- Abkürzungen sind verboten, allerdings werden wir eine Möglichkeit kennenlernen, einen Ausdruck automatisch vervollständigen zu lassen (siehe Abschnitt 4.1).
- Groß/Klein-Schreibung** ist signifikant. Die Namen der eingebauten Mathematica-Funktionen beginnen z.B. mit Großbuchstaben. Wenn Sie diese durch Kleinbuchstaben ersetzen, erhalten Sie eine Fehlermeldung, z.B.:

```
In[1]:= log[1]
```

```
General::spell1: Possible spelling error: new symbol  
name "log" is similar to existing symbol "Log".
```

```
Out[1]= log[1]
```

Wie bei arithmetischen Operationen (vgl. Abschnitt 3.2.2) versucht Mathematica auch bei der Auswertung von Funktionsausdrücken, exakte Ergebnisse zu finden, z.B.:

```
In[1]:= 30!
```

```
Out[1]= 265252859812191058636308480000000
```

Sie können eine Näherungslösung erzwingen, indem Sie die Funktion **N** (für: "**N**umeric") auf den Ausdruck anwenden:

```
In[1]:= N[30!]
```

```
Out[1]= 2.65253×1032
```

3.2.4 Konstanten

Sie können in Mathematica-Ausdrücken einige **Konstanten** über *Namen* ansprechen, z.B.:

```
In[1]:= Pi^2//N
```

```
Out[1]= 9.8696
```

In *In[1]* wird die **N(umeric)**-Funktion in Postfix-Schreibweise angewendet.
Einige oft benötigte Konstanten mit ihren Mathematica-Namen:

Mathematica-Name	Wert
Pi	3,1459...
E	2,7182...
I	$\sqrt{-1}$
Infinity	∞

Über weitere Konstanten informiert der **Help Browser** unter **Built-in Functions > Mathematical Functions > Mathematical Constants**.

3.2.5 Komplexe Zahlen

Komplexe Zahlen werden in Mathematica mit Hilfe der Konstanten **I** (groß!) geschrieben, z.B.:

```
In[1]:= Sqrt[-4]
```

```
Out[1]= 2 I
```

```
In[2]:= Exp[2+9I]//N
```

```
Out[2]= -6.73239 + 3.04517 I
```

Mathematica kennt die üblichen Operationen mit komplexen Zahlen:

```
In[3]:= Re[1+2I]
```

```
Out[3]= 1
```

```
In[4]:= Im[1+2I]
```

```
Out[4]= 2
```

```
In[5]:= Conjugate[1+2I]
```

```
Out[5]= 1 - 2 I
```

```
In[6]:= Abs[1+I]
```

```
Out[6]= Sqrt[2]
```

```
In[7]:= Arg[1+I]
```

```
Out[7]=  $\frac{\pi}{4}$ 
```

3.3 Wichtige Bestandteile und Regeln der Mathematica-Sprache

Natürlich ist Mathematica unvergleichlich mächtiger als ein Taschenrechner. Sie können seine Leistungen über die Mathematica-Sprache abrufen, über die Sie in diesem Abschnitt einige elementare Informationen erhalten.

3.3.1 Rückgriff auf vorangegangene Ausgaben

Beim Arbeiten mit Mathematica ist es oft bequem, in der aktuellen Input-Zelle eine frühere Ausgabe anzusprechen:

Symbol	Bedeutung	Beispiel
%	letzte Ausgabe	N[%]
%%	vorletzte Ausgabe	Show[%%, ViewPoint -> {0, -2, 0}]
%n	Inhalt der Ausgabe Out[n]	%2 + %3

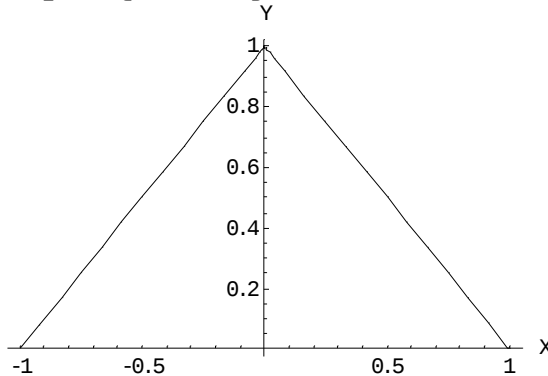
3.3.2 Definition von Variablen

Beim Aufbau komplizierter Ausdrücke ist es sinnvoll, Zwischenergebnisse in benannten Variablen zu speichern, z.B.:

```
In[1]:= rd = 1-Sign[x]x
```

```
Out[1]= 1 - x Sign[x]
```

```
In[2]:= Plot[rd,{x,-1,1}, AxesLabel -> {X, Y}]
```



```
Out[2]= -Graphics-
```

Achtung: Variablendefinitionen bleiben bis zum Widerruf wirksam. Aus der Nichtbeachtung dieser Regel resultieren die meisten Fehler beim Arbeiten mit Mathematica. Nicht mehr benötigte Definitionen sollten daher zur Vermeidung unerwünschter Effekte folgendermaßen aufgehoben werden:

```
In[3]:= rd = .
```

Regeln für **Variablenamen**:

- Das erste Zeichen muß ein Buchstabe sein.
- **Groß/Klein-Schreibung** ist signifikant, „b“ und „B“ bezeichnen also zwei verschiedene Variablen.
- Der erste Buchstabe sollte *klein* geschrieben werden, damit Benutzer-Variablen gut von eingebauten Mathematica-Objekten unterschieden werden können.

3.3.3 Listen von Objekten

Sie können mit Hilfe der geschweiften Klammern mehrere Objekte zu einer Liste zusammenfassen und anschließend gemeinsam behandeln, z.B.:

```
In[1]:= liste = {3,4,5}
```

```
Out[1]= {3, 4, 5}
```

```
In[2]:= liste ^ 2
```

```
Out[2]= {9, 16, 25}
```

```
In[3]:= Log[list] // N
```

```
Out[3]= {1.09861, 1.38629, 1.60944}
```

Sie können ...

- eine Liste einer Variablen als Wert zuweisen (siehe *In[1]* oben)
- Listen in arithmetischen Operationen verwenden (siehe *In[2]* oben)
- Listen als Argumente in mathematischen Funktionen verwenden (siehe *In[3]* oben)

Listen können in Mathematica die Funktionen der Felder (Arrays) anderer Computersprachen übernehmen. Wenn Sie Elemente einer Liste ansprechen wollen, sind die Indizes in doppelte eckige Klammern einzuschließen, z.B.

```
In[4]:= liste[[2]]
```

```
Out[4]= 4
```

3.4 Mathematica-Pakete

Mathematica-Pakete sind Dateien (mit der Extension ".m"), die zahlreiche Mathematica-Definitionen enthalten und dadurch spezielle Anwendungsgebiete erschließen. Zum Mathematica-Lieferumfang gehören zahlreiche Standardpakete. Mit dem Help Browser (**Add-ons > Standard Packages**) kann man sich einen Überblick über die Standardpakete und ihre Funktionen verschaffen.

Die Paketfunktionen werden genauso benutzt wie die eingebauten Funktionen. Manche regelmäßig verfügbare Mathematica-Funktionen werden sogar über Pakete realisiert, die beim Start automatisch geladen werden.

Natürlich muß ein Paket zuerst geladen werden, bevor man seine Funktionen benutzen kann, z.B.:

```
In[1]:= data={1,2,3}
```

```
Out[1]= {1, 2, 3}
```

```
In[2]:= <<Statistics'DescriptiveStatistics'
```

```
In[3]:= Mean[data]
```

```
Out[3]= 2
```

Achtung: Wenn Sie vor dem Laden eines Paketes eine darin befindliche Funktion angesprochen haben, müssen Sie das damit unbeabsichtigt eingeführte Symbol zunächst wieder aus dem System entfernen, weil sonst die konkurrierende Definition des Paketes *nicht* benutzt wird, z.B.:

```
In[2]:= Mean[data]
```

```
Out[2]= Mean[{1, 2, 3}]
```

```
In[3]:= <<Statistics'DescriptiveStatistics'
Mean::shdw : Symbol Mean appears in multiple contexts
{Statistics'DescriptiveStatistics', Global'}; definitions
in context Statistics'DescriptiveStatistics' may
shadow or be shadowed by other definitions.
```

So beseitigt man ein Symbol:

```
In[2]:= Remove[Mean]
```

3.5 Algebraische Manipulationen

Bisher hat Mathematica für uns numerische Lösungen (exakt oder angenähert) ermittelt. Nun lernen wir seine Fähigkeiten kennen, algebraische bzw. arithmetische Ausdrücke mit **Symbolen** umzuformen, z.B.:

```
In[1]:= Sqrt[z]^2 (3x - y - 2x)(y + x) z^-1
```

```
Out[1]= (x - y) (x + y)
```

```
In[2]:= Expand[%]
```

```
Out[2]= x^2 - y^2
```

```
In[3]:= Factor[%]
```

```
Out[3]= (x - y) (x + y)
```

```
In[4]:= Apart[%^-1]
```

```
Out[4]= - 1 / (2 x (-x + y)) + 1 / (2 x (x + y))
```

```
In[5]:= Simplify[%]
```

```
Out[5]= 1 / (x^2 - y^2)
```

Wie *Out[1]* zeigt, bemüht sich Mathematica automatisch, Ausdrücke zu vereinfachen. Darüber hinaus gibt es zahlreiche Möglichkeiten, die Umformung eines algebraischen Ausdrucks zu steuern, von denen hier nur die wichtigsten genannt werden sollen:

Expand[ausdruck]	Produkte und Potenzen werden ausmultipliziert, bei rationalen Ausdrücken aber nur im Zähler.
ExpandAll[ausdruck]	Expand wird bei rationalen Ausdrücken auch auf den Nenner angewandt.
Factor[ausdruck]	Factor versucht, den Ausdruck in Faktoren zu zerlegen.
Apart[ausdruck]	Zerlegt einen rationalen Ausdruck in eine Summe von Termen mit möglichst einfachen Nennern.
Simplify[ausdruck]	Sucht eine möglichst kleine Form des Ausdrucks.

Die bisher genannten Transformationen sind *immer* korrekt, d.h. gültig für alle möglichen Werte der symbolischen Variablen. Das gilt z.B. auch für die folgende, von Mathematica automatisch ausgeführte, Vereinfachung der ganzzahligen Potenz eines Produktes:

```
In[1]:= (x y)^2
```

```
Out[1]= x^2 y^2
```

Mathematica vereinfacht Ausdrücke der Form $(x y)^c$ jedoch nicht automatisch, wenn c keine ganze Zahl ist, z.B.:

```
In[2]:= (x y)^c
```

```
Out[2]= (x y)^c
```

Die Potenzregel $(x \cdot y)^c = x^c \cdot y^c$ gilt nämlich bei nicht ganzzahligem c nur für $x, y > 0$.

PowerExpand führt demgegenüber auch Transformationen durch, die nur unter bestimmten Voraussetzungen stimmen, im Beispiel für $x, y > 0$:

```
In[3]:= PowerExpand[%]
```

```
Out[3]= x^c y^c
```

Im **Help Browser** finden Sie eine komplette Liste der möglichen Transformationen von algebraischen Ausdrücken unter **Built-in Functions > Algebraic Computation**.

3.6 Mathematische Operationen mit Symbolen

In diesem Abschnitt wird gezeigt, wie mathematische Standardmethoden (z.B. Differentiation, Integration) auf Ausdrücke mit Symbolen angewandt werden.

3.6.1 Differenzieren

Mathematica kann alle elementaren mathematischen Funktionen ableiten. Die folgende Darstellung beschränkt sich auf die wichtigsten Möglichkeiten. Eine vollständige Beschreibung finden Sie in der Online-Hilfe zu den Mathematica-Operatoren **D** und **Dt**.

3.6.1.1 Partielle Ableitung (Operator D)

Mit dem Mathematica-Operator **D** können Sie eine Funktion f partiell nach einer Variablen x ableiten lassen, wobei für alle anderen Symbole im Funktionsausdruck angenommen wird, daß sie nicht von x abhängen.

$D[f, x]$	$\frac{\partial f}{\partial x}$
-----------	---------------------------------

Im folgenden Beispiel wird n als Konstante behandelt:

```
In[1]:= D[x^n, x]
```

```
Out[1]= n x^{-1+n}
```

3.6.1.2 Totale Ableitung (Operator Dt)

Bei der sogenannten *totalen* Ableitung einer Funktion f nach einer Variablen x wird im Gegensatz zu der eben beschriebenen partiellen Ableitung angenommen, daß die restlichen Symbole im Ausdruck von x abhängen.

$Dt[f, x]$	$\frac{d f}{d x}$
------------	-------------------

Beispiel:

In[1]:= Dt[x^n, x]

Out[1]= $n x^{-1+n} + x^n \text{Dt}[n, x] \text{Log}[x]$

Hier wird angenommen, daß n von x abhängen kann. "Dt[n, x]" steht für $\frac{dn}{dx}$. Wer will, kann sich von der

Richtigkeit des Ergebnisses überzeugen:

$$\left[x^{n(x)} \right]' = \left[e^{n(x) \ln(x)} \right]' = x^{n(x)} \left[n'(x) \ln(x) + \frac{n(x)}{x} \right] = n(x) x^{n(x)-1} + x^{n(x)} n'(x) \ln(x)$$

3.6.1.3 Ableiten unbekannter Funktionen

Mathematica kann auch Ausdrücke mit Symbolen für unbekannte Funktionen ableiten, z.B.:

In[1]:= D[f[x]^2, x]

Out[1]= $2 f[x] f'[x]$

Im nächsten Beispiel liefert Mathematica eine wohlbekannte Ableitungsregel:

In[2]:= D[f[x]/g[x], x]

Out[2]= $\frac{f'[x]}{g[x]} - \frac{f[x] g'[x]}{g[x]^2}$

3.6.2 Integrieren

3.6.2.1 Unbestimmtes Integral

Mathematica kann Stammfunktionen bestimmen.

Integrate[f, x]	$\int f(x) dx$
-----------------	----------------

Um Mathematica testen zu können, wollen wir zunächst eine Stammfunktion F zur Funktion

$$f(x) = \frac{x}{\sqrt{1-x^2}} \text{ mit der Substitutionsregel selbst ausrechnen (vgl. z.B. Heuser 1986, Teil 1, S. 442):}$$

Wir substituieren x durch $\varphi(t) := \sqrt{1-t}$, wobei gilt: $\varphi'(t) = -\frac{1}{2} \frac{1}{\sqrt{1-t}}$ und $\varphi^{-1}(x) = 1-x^2$. Die Stammfunktion $\Phi(t)$ von $f(\varphi(t))\varphi'(t)$ ist leicht zu ermitteln:

$$\int f(\varphi(t))\varphi'(t) dt = -\frac{1}{2} \int \frac{\sqrt{1-t}}{\sqrt{1-t}} \frac{1}{\sqrt{1-t}} dt = -\frac{1}{2} \int \frac{1}{\sqrt{t}} dt = -\frac{1}{2} \frac{t^{\frac{1}{2}}}{\frac{1}{2}} = -\sqrt{t}$$

Dann ist $F(x) := \Phi(\varphi^{-1}(x))$ eine Stammfunktion zu f :

$$F(x) = -\sqrt{1-x^2}$$

Nun die Lösung von Mathematica:

In[1]:= Integrate[x/Sqrt[1-x^2], x]

Out[1]= $-\sqrt{1-x^2}$

3.6.2.2 Bestimmtes Integral

Auch bei bestimmten Integralen mit symbolischen Grenzen versucht Mathematica, einen geschlossenen Ausdruck als Lösung zu finden.

$\text{Integrate}[f, \{x, \min, \max\}]$	$\int_{\min}^{\max} f(x) dx$
--	------------------------------

Auch hier wollen wir zur Probe ein Beispiel per Hand ausrechnen (mit partieller Integration, siehe z.B. Heuser 1986, Teil 1, S. 463):

$$\int_0^a e^x x dx = e^x x \Big|_0^a - \int_0^a e^x dx = e^a a - e^x \Big|_0^a = e^a a - (e^a - 1) = e^a (a - 1) + 1$$

Die Lösung von Mathematica:

In[1]:= Integrate[Exp[x]x, {x, 0, a}]

Out[1]= 1 + (-1 + a) E^a

3.6.2.3 Numerische Lösungen

Für manche bestimmte Integrale existiert kein geschlossener Ausdruck, z.B. für das Integral der Normalverteilungsdichte:

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^a e^{-\frac{x^2}{2}} dx$$

In[1]:= Sqrt[2Pi]^(-1) Integrate[Exp[-0.5 * x^2], {x, -Infinity, a}]

Out[1]= $\frac{1}{\sqrt{2\pi}} \left(\text{If}[a < 0, 1.25331 - 1. \sqrt{a^2} \text{Hypergeometric1F1}\left[\frac{1}{2}, \frac{3}{2}, -0.5 a^2\right], \int_{-\infty}^a e^{-0.5 x^2} dx] \right)$

Mathematica unterscheidet in seiner Lösung die Fälle ($a < 0$) und ($a > 0$) und gibt für den interessanteren zweiten Fall das Integral unbearbeitet wieder.

In einer solchen Situation erhält man durch Einsetzen konkreter Integrationsgrenzen jedoch eine numerische Lösung, z.B.:

In[2]:= Sqrt[2Pi]^(-1) Integrate[Exp[-0.5 * x^2], {x, -Infinity, a}]

Out[2]= 0.975002

3.6.3 Summen und Produkte

3.6.3.1 Summe $\sum_{i=\min}^{i\max} f$

In folgendem Beispiel werden die ersten 11 Glieder der geometrischen Reihe zur Basis a aufaddiert, wobei die Input-Zelle mit Hilfe der Eingabe-Palette in mathematischer Standardnotation erstellt wurde:

In[1]:= Sum[a^i, {i, 0, 10}]

Out[1]= 1 + a + a² + a³ + a⁴ + a⁵ + a⁶ + a⁷ + a⁸ + a⁹ + a¹⁰

Wenn ein expliziter Wert als Basis angegeben, produziert Mathematica ein numerisches Ergebnis, das nach den oben vereinbarten Konventionen exakt:

```
In[2]:= Sum[(1/2)^i, {i, 0, 10}]
```

```
Out[2]= 2047
        1024
```

oder angenähert ausgegeben wird:

```
In[3]:= Sum[0.5^i, {i, 0, 10}]
```

```
Out[3]= 1.99902
```

Die Syntax für die Anforderung einer Summe ist denkbar einfach:

```
Sum[f, {i, imin, imax, di}]
```

Mit dem optionalen letzten Parameter kann die Schrittweite für den Laufindex auf einen alternativen Wert gesetzt werden, z.B.:

```
In[4]:= Sum[a^i, {i, 0, 10, 2}]
```

```
Out[4]= 1 + a^2 + a^4 + a^6 + a^8 + a^10
```

Mathematica 3.0 kann den Wert konvergenter unendlicher Reihen nun direkt angeben:

```
In[5]:= Sum[(1/2)^i, {i, 0, Infinity}]
```

```
Out[5]= 2
```

Die Vorgängerversion 2.2 hatte sich hier noch zurückgehalten:

```
Out[5]=
```

```
Sum[(-)^(1/2)^i, {i, 0, Infinity}]
```

3.6.3.2 Doppelsumme $\sum_{i=i_{\min}}^{i_{\max}} \sum_{j=j_{\min}}^{j_{\max}} f$

In folgendem Beispiel werden vier geometrische Reihen addiert:

```
In[1]:= Sum[(1/i)^j, {i, 2, 5}, {j, 0, Infinity}]
```

```
Out[1]= 73
        12
```

Die Mathematica-Syntax für Doppelsummen:

```
Sum[f, {i, imin, imax}, {j, jmin, jmax}]
```

3.6.3.3 Produkt $\prod_{i=i_{\min}}^{i_{\max}} f$

Wegen

$$\prod_{i=1}^{\infty} e^{(1/2)^i} = e^{\sum_{i=1}^{\infty} (1/2)^i} = e^1$$

kann man die Eulersche Zahl auch auf die folgende, umständliche, Art ausrechnen:

```
In[1]:= Product[Exp[0.5^i], {i, 1, Infinity}]
```

```
Out[1]= 2.71828
```

Die Mathematica-Syntax für Produkte:

`Product[f, {i, imin, imax, di}]`

3.6.4 Ausdrücke auf Gleichheit testen

Mit dem einfachen Gleichheitszeichen wird in Mathematica einer Variablen ein Ausdruck zugeordnet, z.B.:

`In[1]:= y=2`

`Out[1]= 2`

`In[2]:= x=2y`

`Out[2]= 4`

Ein doppeltes Gleichheitszeichen steht für den Vergleichsoperator "==", mit dem getestet werden kann, ob die Ausdrücke auf den beiden Seiten des Operatorzeichens übereinstimmen, z.B.:

`In[3]:= x==4`

`Out[3]= True`

Mathematica kann auch symbolische Ausdrücke miteinander vergleichen:

`In[1]:= (x+y)(x-y)==x^2-y^2`

`Out[1]= True`

`In[2]:= (x+y)(y-y)==x^2-y^2`

`Out[2]= False`

3.6.5 Relationale und logische Operatoren

Die im letzten Abschnitt behandelten Gleichungen sind spezielle Aussagen, also sprachliche Gebilde, die wahr, falsch oder unbestimmt (z.B. „ $a == b$ “ mit undefinierten Variablen a und b) sein können. Mit Hilfe der bekannten Operatorzeichen können wir noch andere Aussagen bilden, für deren Wahrheitsgehalt wir uns gelegentlich interessieren. Ihnen ist gemeinsam, daß ein Vergleichsoperator zwischen zwei Ausdrücken auftritt, so daß wir die entstehenden Aussagen als **Vergleiche** bezeichnen können.

Mathematica kennt die folgenden Vergleiche:

$x == y$	gleich
$x != y$	ungleich
$x > y$	größer als
$x >= y$	größer oder gleich
$x < y$	kleiner als
$x <= y$	kleiner oder gleich

Beispiele:

`In[1]:= 6 > 8`

`Out[1]= False`

`In[2]:= a < b`

`Out[2]= a < b`

Der letzte Ausdruck ist unbestimmt, weil die Variablen a und b momentan keine feste numerische Bedeutung haben.

Ein Vergleich ist ein besonders einfach aufgebauter **logischer Ausdruck**. Mit Hilfe der von logischen Operatoren können wir aus vorhandenen logischen Ausdrücke neue, komplexere herstellen. Die drei bekanntesten logischen Operatoren, Negation, Konjunktion und Adjunktion, werden in Mathematica folgendermaßen notiert:

$!p$	NOT
$p \&\& q \&\& \dots$	UND
$p \parallel q \parallel \dots$	ODER

Beispiele:

```
In[3]:= 2<4 && 7>6
```

```
Out[3]= True
```

Weitere logische Operatoren finden Sie im Help Browser unter:

Build-in Functions > Programming > Logical Operations

3.6.6 Gleichungen lösen

In diesem Abschnitt beschäftigen wir uns mit der Suche nach den *exakten* Lösungen von Gleichungen. Numerische Lösungsmethoden werden später behandelt.

3.6.6.1 Algebraische Gleichungen in einer Variablen

Mit der Mathematica-Funktion **Solve** kann man Polynom-Gleichungen mit einer höchsten Potenz kleiner oder gleich vier stets lösen, z.B.:

```
In[1]:= Solve[x^2==1,x]
```

```
Out[1]= {{x -> -1}, {x -> 1}}
```

Als Ergebnis erhalten wir eine Liste von **Ersetzungsregeln** für x , welche die Gleichung erfüllen. Diese Form der Ergebnisausgabe wird gewählt, weil man in Mathematica solche Ersetzungsregeln auf beliebige Ausdrücke mit der Ersetzungsvariablen anwenden kann, um konkrete Ergebnisse unter Verwendung der Ersetzungswerte zu erhalten. Wenn man z.B. die Ersetzungsregeln aus obiger Lösung auf den einfachen Ausdruck x anwendet, resultieren die Lösungswerte selbst:

```
In[2]:= x/.%
```

```
Out[2]= {-1, 1}
```

Man kann die Ersetzungen aber auch auf beliebige andere Ausdrücke mit der Variablen x anwenden:

```
In[3]:= Exp[x] /. {{x -> -1}, {x -> 1}}
```

```
Out[3]= {1/E, E}
```

Die allgemeine Syntax zur Verwendung von Ersetzungsregeln:

<i>ausdruck /. Ersetzungsregeln</i>	Verwende die Liste der Ersetzungsregeln, um Werte für den Ausdruck zu erhalten.
-------------------------------------	---

Es können auch Gleichungen gelöst werden, die symbolische Konstanten enthalten. Im folgenden Beispiel erhalten wir als Lösungsmenge die wohlbekannte „p-q-Regel“:

In[4]:= Solve[x^2 + a x + b == 0, x]

Out[4]= $\left\{ \left\{ x \rightarrow \frac{1}{2} \left(-a - \sqrt{a^2 - 4b} \right) \right\}, \left\{ x \rightarrow \frac{1}{2} \left(-a + \sqrt{a^2 - 4b} \right) \right\} \right\}$

Erinnerung: Sollten Sie eine andere Lösungsmenge erhalten haben, waren in Ihrer Mathematica-Sitzung mit großer Wahrscheinlichkeit die Variablen a und b bereits mit Werten belegt.

Mathematica kann auch einige Polynom-Gleichungen mit Potenzen größer als vier lösen, z.B.:

In[5]:= Solve[x^5==5, x]

Out[5]= $\left\{ \left\{ x \rightarrow -(-5)^{1/5} \right\}, \left\{ x \rightarrow 5^{1/5} \right\}, \left\{ x \rightarrow (-1)^{2/5} 5^{1/5} \right\}, \left\{ x \rightarrow -(-1)^{3/5} 5^{1/5} \right\}, \left\{ x \rightarrow (-1)^{4/5} 5^{1/5} \right\} \right\}$

Kann Mathematica *keine* expliziten Lösungen bestimmen, wird das Ergebnis in symbolischer Form mit Hilfe der **Root**-Funktion ausgedrückt, z.B.:

In[7]:= Solve[x^6-x^5==6, x]

Out[7]= $\left\{ \left\{ x \rightarrow \text{Root}[-6 - \#1^5 + \#1^6, 1] \right\}, \left\{ x \rightarrow \text{Root}[-6 - \#1^5 + \#1^6, 2] \right\}, \left\{ x \rightarrow \text{Root}[-6 - \#1^5 + \#1^6, 3] \right\}, \left\{ x \rightarrow \text{Root}[-6 - \#1^5 + \#1^6, 4] \right\}, \left\{ x \rightarrow \text{Root}[-6 - \#1^5 + \#1^6, 5] \right\}, \left\{ x \rightarrow \text{Root}[-6 - \#1^5 + \#1^6, 6] \right\} \right\}$

Bei *Polynom*-Gleichungen können Sie dann auf jeden Fall mit **N** alle numerischen Lösungen anfordern:

In[8]:= N[%]

Out[8]= $\left\{ \left\{ x \rightarrow -1.22 \right\}, \left\{ x \rightarrow 1.59015 \right\}, \left\{ x \rightarrow -0.537119 - 1.13651 I \right\}, \left\{ x \rightarrow -0.537119 + 1.13651 I \right\}, \left\{ x \rightarrow 0.852045 - 1.10965 I \right\}, \left\{ x \rightarrow 0.852045 + 1.10965 I \right\} \right\}$

3.6.6.2 Transzendente Gleichungen in einer Variablen

Mathematica kann auch manche transzendente Gleichungen lösen, wobei in der Regel eine Warnung darauf hinweist, daß vermutlich nicht alle Lösungen gefunden wurden, z.B.:

In[1]:= Solve[Sin[x]==a, x]

Solve::ifun: Warning: Inverse functions are being used by Solve, so some solutions may not be found.

Out[1]= $\left\{ \left\{ x \rightarrow \text{ArcSin}[a] \right\} \right\}$

In der Regel können transzendente Gleichungen jedoch nicht in symbolischer Form gelöst werden, z.B.:

In[2]:= Solve[Cos[x]==x, x]

Solve::tdep: The equations appear to involve transcendental functions of the variables in an essentially non-algebraic way.

Out[2]= Solve[Cos[x]==x, x]

In diesem Fall können Sie eine numerische Näherungslösung ermitteln, indem Sie die Funktion **FindRoot** verwenden und einen geeigneten Startwert vorgeben:

In[3]:= FindRoot [Cos[x]==x, {x, 1}]

Out[3]= $\{ x \rightarrow 0.739085 \}$

Mit den numerischen Lösungen für transzendente Gleichungen werden wir uns später noch näher beschäftigen.

3.6.6.3 Gleichungssysteme mit mehreren Veränderlichen

Mit der Mathematica-Funktion **Solve** lassen sich alle linearen Gleichungssysteme und zahlreiche Systeme mit Polynom-Gleichungen explizit lösen.

i) Zwei Gleichungen in einer Veränderlichen x

```
In[1]:= Solve[{x^2==1, x^3==1}, x]
```

```
Out[1]= {{x -> 1}}
```

ii) Zwei Gleichungen in zwei Veränderlichen x und y

Hier wird man in der Regel nach x und y auflösen wollen, weil eine Auflösung nach x allein der Suche nach denjenigen x -Werten entspräche, welche die Gleichungen für beliebige (ggf. fest vorgegebene) y -Werte erfüllen.

```
In[2]:= Solve[{y==x/2+3, y==x-3}, {x, y}]
```

```
Out[2]= {{x -> 12, y -> 9}}
```

iii) Mehrere Gleichungen in mehreren Veränderlichen

Wenn z.B. zwei Gleichungen in den drei Veränderlichen x , y und a nach x und y aufgelöst werden sollen, erhalten wir Ergebnisterme mit a :

```
In[3]:= Solve[{x==y-3a, y==2x-a}, {x, y}]
```

```
Out[3]= {{x -> 4 a, y -> 7 a}}
```

In dieser Situation kann man aber auch eine Veränderliche, z.B. a , eliminieren. Dann bleibt noch eine Gleichung mit den beiden Veränderlichen x und y übrig:

$$\begin{array}{rclcl} x & - & y & + & 3a & = & 0 \\ 2x & - & y & - & a & = & 0 \\ \hline 7x & - & 4y & & & = & 0 \end{array}$$

Mit folgender Syntax erhalten Sie sofort die Auflösung nach x bei Elimination von a :

```
In[4]:= Solve[{x==y-3a, y==2x-a}, x, a]
```

```
Out[4]= {{x -> \frac{4 y}{7}}}
```

iv) Die allgemeine Syntax der Funktion Solve

`Solve[{ $ls_1 == rs_1$ ,  $ls_2 == rs_2$ , ...}, { $x, y$ , ...}, { $a, b$ , ...}]` Löse das Gleichungssystem nach $x, y, \dots$ durch Eliminieren von $a, b, \dots$

Die geschweiften Klammern dürfen entfallen, wenn die zugehörige Liste einelementig ist.

3.6.6.4 Vollständige Lösungen

Wir haben in Abschnitt 3.6.6 bislang die **Solve**-Funktion benutzt. Alle dabei erzielten Ergebnisse können wir auch mit der verwandten Funktion **Reduce** ermitteln, wobei aber ein anderes Ausgabeformat verwendet wird. An Stelle von Ersetzungslisten liefert **Reduce** logische Ausdrücke, z.B.:

```
In[1]:= Solve[x^2==1, x]
```

```
Out[1]= {{x -> -1}, {x -> 1}}
```

```
In[2]:= Reduce[x^2==1, x]
```

```
Out[2]= x == -1 || x == 1
```

Dabei stehen die beiden senkrechten Striche für den Operator **Oder** (vgl. Abschnitt 3.6.5). Es gibt jedoch noch einen weiteren wichtigen Unterschied zwischen **Solve** und **Reduce**. Während **Solve** nur allgemein-

gültige Lösungen liefert, sucht **Reduce** zusätzlich auch die Lösungen, die nur für bestimmte Werte anderer Variablen gelten:

```
In[3]:= Solve[a x == 0, x]
```

```
Out[3]= {{x -> 0}}
```

Reduce berücksichtigt auch die Möglichkeit, daß a gleich Null ist:

```
In[4]:= Reduce[a x == 0, x]
```

```
Out[4]= a == 0 || x == 0
```

Im ersten Teil der Adjunktion sind die Lösungspaare aus der Menge $\{(x, a): x \in \mathbb{R}, a = 0\}$ gemeint. Die Syntax der Funktion **Reduce** stimmt mit der **Solve**-Syntax überein (s.o.).

3.6.7 Differentialgleichungen

Die Funktion **DSolve** löst lineare und nichtlineare gewöhnliche Differentialgleichungen sowie Differentialgleichungssysteme und liefert wie die Funktion **Solve** Ersetzungsregeln als Ergebnisse. In folgendem Beispiel wird die Differentialgleichung $y'(x) = a y(x)$ gelöst:

```
In[1]:= DSolve[y'[x]==a y[x], y[x], x]
```

```
Out[1]= {{y[x] -> E^{a x} C[1]}}
```

Die unbestimmte Konstante $C[1]$ kann über eine Anfangsbedingung festgelegt werden, z.B.:

```
In[2]:= DSolve[{y'[x]==a y[x], y[0]==1}, y[x], x]
```

```
Out[2]= {{y[x] -> E^{a x}}}
```

Wir erhalten das Ergebnis in Form einer Ersetzungsregel für $y[x]$. Leider läßt sich diese Ersetzungsregel nur eingeschränkt weiter verwenden, wie der folgende mißlungene Versuch einer „Probe“ für die gefundene Lösung (mit Hilfe des Ableitungs-Operators **D**) demonstriert:

```
In[3]:= D[y[x], x]/.%
```

```
Out[3]= {y'[x]}
```

Um besser verwendbare Ersetzungsregeln zu erhalten, müssen wir im **DSolve**-Aufruf statt der Lösung für $y(x)$ eine Lösung für y im Form einer sogenannten **reinen Funktion** anfordern:

```
In[4]:= DSolve[{y'[x]==a y[x], y[0]==1}, y, x]
```

```
Out[4]= {{y -> (E^{a #1} &)}}
```

Das „&“ hinter der Definition der Ergebnisfunktion weist darauf hin, daß deren Argumente keine Namen erhalten haben, sondern mit #1, #2, usw. durchnummeriert worden sind. Auch die Funktion selbst bleibt ohne Namen, es wird lediglich die auf ein Argument anzuwendende Vorschrift angegeben.

Nun gelingt die Weiterverarbeitung der Ersetzungsregel mit der Ableitungs-Operation **D**:

```
In[5]:= D[y[x], x]/.%
```

```
Out[5]= {a E^{a x}}
```

Die allgemeine Syntax der Funktion **DSolve**:

$\text{DSolve}[\{gl_1, gl_2, \dots\}, y[x], x]$	Löse die Differentialgleichungen nach $y[x]$ mit x als unabhängiger Variable
---	--

DSolve[{gl ₁ , gl ₂ , ...}, y, x]	Ermittle eine Lösung für y in Form einer reellen Funktion.
---	--

Die geschweiften Klammern dürfen entfallen, wenn die zugehörige Liste einelementig ist.

3.6.8 Potenzreihen (Taylor-Reihen)

Viele Funktionen lassen sich durch Potenzreihen endlicher Ordnung approximieren. Eine wesentliche Rolle spielt dabei der Satz von Taylor:

Sei f auf dem Intervall I $(n+1)$ mal stetig differenzierbar und x_0 ein innerer Punkt von I . Dann gilt für alle $x \in I$:

$$f(x) = \sum_{v=0}^n \frac{f^{(v)}(x_0)}{v!} (x - x_0)^v + R_n(x, x_0)$$

Zum Restglied R_n siehe z.B. Endl & Luh (1983, Band 2, S.202ff).

Mit der Mathematica-Operation **Series** kann man für viele Funktionen eine Potenzreihenentwicklung um einen Punkt x_0 bis zur gewünschten Ordnung n ermitteln, z.B.:

In[1]:= Series[Exp[x], {x, 0, 4}]

Out[1]= $1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24} + O[x]^5$

An der Lösung für eine unbekannte Funktion f sieht man, daß Mathematica den Satz von Taylor verwendet:

In[2]:= Series[f[x], {x, 0, 4}]

Out[2]= $f[0] + f'[0] x + \frac{1}{2} f''[0] x^2 + \frac{1}{6} f^{(3)}[0] x^3 + \frac{1}{24} f^{(4)}[0] x^4 + O[x]^5$

Für manche Funktionen liefert Mathematica allerdings Potenzreihenentwicklungen, die *keine* Taylor-Reihen sind.

Die Syntax der Funktion **Series**:

Series[ausdruck, {x, x ₀ , n}]	Ermittle die Reihenentwicklung für den Ausdruck um den Entwicklungspunkt $x = x_0$ bis zur n -ten Ordnung.
---	--

3.6.9 Grenzwerte

In Mathematica können Sie z.B. den Grenzwert der Funktion $\frac{\sin(x)}{x}$ für $x \rightarrow 0$ folgendermaßen bestimmen:

In[1]:= Limit[Sin[x]/x, x -> 0]

Out[1]= 1

3.7 Eingabe in mathematischer Notation, Arbeiten mit Paletten

Wir haben bisher in den Input-Zellen die klassische Mathematica-Syntax (InputForm) benutzt, für die neben dem Gewohnheitsargument nach wie vor einige weitere Vorteile sprechen, z.B.:

- Allerdings hat die traditionelle Syntax auch Nachteile:

- Erfreulicherweise unterstützt Mathematica seit der Version 3.0 die mathematische Notation nicht nur in den Output – Zellen (s.o.), sondern auch in den Input-Zellen. Wir werden uns in diesem Abschnitt mit den Eingabehilfen beschäftigen, die zum Erstellen der zwar intuitiven, aber leider auch technisch aufwendigen mathematischen Notation verfügbar sind.

3.7.1 Eingabe in mathematischer Notation mit Hilfe von Paletten

[illegible]

```
In[1]:= Integrate[Exp[x]x,{x,0,a}]
```

- Klicken Sie in der Palette auf das Integral mit Platzhaltern für die Integrationsgrenzen , welches daraufhin in der aktuellen Input-Zelle erscheint.
- Ersetzen Sie den (durch einen kleinen Punkt markierten) Platzhalter für die untere Integrationsgrenze durch eine Null.
- Markieren Sie den Platzhalter für die obere Integrationsgrenze per Mausklick, und ersetzen Sie ihn durch das Symbol „a“.

- Markieren Sie den Platzhalter für den Integranden per Mausklick, und klicken Sie in der Palette auf das Symbol für einen Potenz-Ausdruck , das daraufhin im Integranden erscheint. Ersetzen Sie die aktuell markierte Basis durch das Mathematica-Symbol „E“ für die Eulersche Zahl. Markieren Sie den Platzhalter für den Exponenten, und ersetzen Sie ihn durch das Variablen-Symbol „x“.
- Erweitern Sie den Integranden noch um den Faktor x. Die richtige Position der Einfügemarke läßt sich per Maus oder über die horizontalen Pfeiltasten einstellen.
- Markieren Sie per Mausklick den Platzhalter für die Integrationsvariable, und fügen Sie dort die Variable x ein.

Ihre Input-Zelle sollte nun folgendermaßen aussehen und kann ausgeführt werden:

$$\int_0^a E^x x \, dx$$

Als Ergebnis erhalten wir wie erwartet: $1 + (-1 + a) E^a$.

Hinweis: Die Arbeit mit den teilweise recht kleinen Platzhaltersymbolen kann durch eine Vergrößerung der Notebook-Anzeige über **Format > Magnification** erleichtert werden.

Über den Menübefehl **File > Palettes** können noch weitere Paletten geöffnet werden, die u.a. über 700 mathematische Sonderzeichen verfügbar machen. Außerdem können Sie über den Menübefehl **Input > Create Table/Matrix/Palette** beliebige eigene Paletten zusammenstellen.

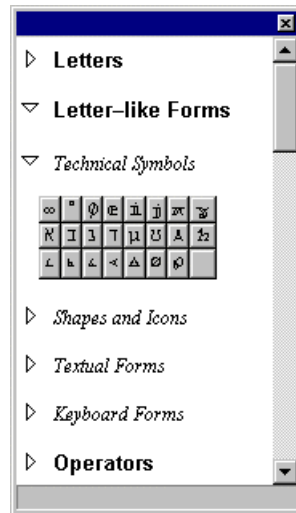
3.7.2 Zwei Varianten der mathematischen Notation: Standardform und traditionelle Form

Bei der Wahl des Symbols für die Eulersche Zahl bestand beim Erstellen der obigen Formel keine gestalterische Freiheit. Ein klein geschriebenes „e“ wird von Mathematica als Bezeichner für eine unbekannte Variable interpretiert, was zu einem völlig anderen Integrationsproblem führt. Wir haben bisher die **Standardform** der mathematischen Notation kennengelernt, die durch absolut eindeutige Bezeichnungen gekennzeichnet ist. Daneben unterstützt Mathematica noch die sogenannte **traditionelle Form**, die zugunsten einer weitestgehenden Angleichung an übliche mathematische Publikationen einige Mehrdeutigkeiten in Kauf nimmt. Eine Input-Zelle in mathematische Standardnotation (oder auch in der eindimensionalen Input-Form) kann über den Menübefehl **Cell > ConvertTo > TraditionalForm** in die traditionelle Form gebracht werden. Für unser Beispiel erhalten wir folgendes Ergebnis mit einem klein geschriebenen Symbol für die Eulersche Zahl:

$$\int_0^a e^x x \, dx$$

Bei diesem Symbol handelt es sich allerdings immer noch um ein Sonderzeichen, so daß die Formel noch keine Mehrdeutigkeiten enthält. Sie kann daher ohne weiteres ausgewertet oder in die Standardform zurück transformiert werden (mit **Cell > ConvertTo > StandardForm**).

Sie finden das klein geschriebene Symbol für die Eulersche Zahl übrigens in der Palette **Complete-Characters** in der Abteilung **Letter-Like Forms, Technical Symbols**:



Generell sollte die traditionelle Form nur dazu verwendet werden, Output zu produzieren, der in andere Dokumente übernommen werden soll.

Insgesamt kennt Mathematica 3.0 vier Formate, die Input- oder Output-Zellen mit dem Menübefehl **Cell > Convert To** zugewiesen werden können:

InputForm	Klassische Mathematica-Syntax, kann mit jeder Standardtastatur eindeutig und bequem eingegeben werden
OutputForm	Dieses Format nutzt einige zweidimensionale Darstellungselemente, beschränkt sich aber auf die Zeichen einer Standardtastatur.
StandardForm	Diese für Input und Output geeignete Form verwendet zweidimensionale Positionierung sowie Sonderzeichen und zeichnet sich durch perfekt-eindeutige Bezeichnungen aus.
TraditionalForm	Diese primär für Output gedachte Form imitiert die übliche mathematische Typographie, wobei gelegentlich die Bezeichnungseindeutigkeit verloren geht.

Über **DefaultInputFormatType** bzw. **DefaultOutputFormatType** können die Mathematica-Voreinstellungen für neue Zellen festgelegt werden.

3.7.3 Eingabe in mathematischer Notation mit Hilfe der Tastatur

Bei der Erstellung von Input-Zellen in mathematischer Notation sind Sie nicht auf Paletten und Maus angewiesen, sondern können alle Eingaben auch per Tastatur bewerkstelligen. Ähnlich wie bei der Input-Form (s.o.) ist der Lohn für den Lernaufwand eine höhere Eingabegeschwindigkeit.

Gehen Sie folgendermaßen vor, um unser Integral-Beispiel zu erstellen (vgl. Wolfram 1996, S. 173ff):

- Drücken Sie die **<Esc>**-Taste, Tippen Sie „int“ und drücken Sie erneut die **<Esc>**-Taste. Es erscheint ein Integralzeichen.
- Drücken Sie die Tastenkombination **<Strg>+<+>**, und geben Sie die Null als untere Integrationsgrenze ein.
- Drücken Sie die Tastenkombination **<Strg>+<%>**, und geben Sie „a“ als obere Integrationsgrenze ein.
- Drücken Sie die Tastenkombination **<Strg>+<Leertaste>**, um auf die Eingabeebene für den Integranden zu gelangen.
- Drücken Sie die **<Esc>**-Taste, Tippen Sie „ee“, und drücken Sie erneut die **<Esc>**-Taste. Es erscheint das Sonderzeichen für die Eulersche Zahl.

- Drücken Sie die Tastenkombination **<Strg>+<^>**. Daraufhin erscheint ein Platzhalter für den Exponenten zur Eulerschen Zahl. Tragen Sie die Variable x als Exponenten ein, bewegen Sie die Schreibmarke nach rechts, und notieren Sie die Variable x auch noch als Faktor im Integranden.
- Drücken Sie die **<Esc>**-Taste, Tippen Sie „dd“, und drücken Sie erneut die **<Esc>**-Taste. Es erscheint das Sonderzeichen für den Differentialoperator. Ergänzen Sie noch die Integrationsvariable x .

Nun sollten Sie folgende Input-Zelle erzeugt haben:

$$\int_0^a e^x x \, dx$$

Die paletten- und die tastatur-gestützte Eingabe können natürlich auch kombiniert werden.

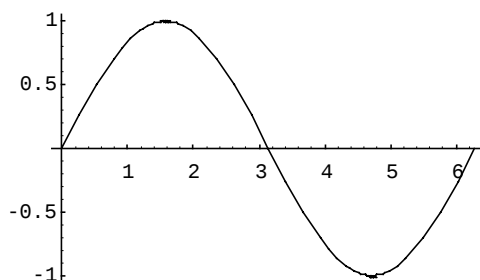
3.8 Grafiken

Bei der Ausführung von Grafikbefehlen (z.B. Plot) stellt der Kernel zunächst alle graphischen Objekte (z.B. Linien, Punkte, Polygone) als Mathematica-Ausdrücke dar und übersetzt die Abbildung dann in die Seitenbeschreibungssprache **Postscript**. Schließlich wird diese Darstellung vom Frontend auf dem gewünschten Gerät wiedergegeben.

3.8.1 Elementare Diagramme

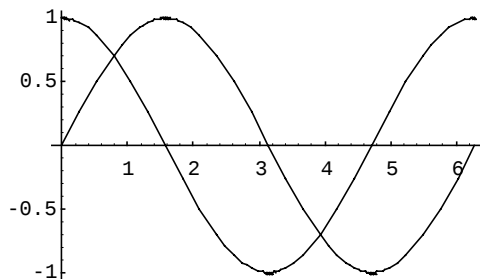
Sie können den **Plot**-Befehl verwenden, um Funktionen zu zeichnen, einzeln oder mehrere zusammen:

In[1]:= Plot[Sin[x], {x,0,2 Pi}]



Out[1]= -Graphics-

In[2]:= Plot[{Sin[x], Cos[x]}, {x,0,2 Pi}]



Out[2]= -Graphics-

Die Syntax:

Plot[f,{x, xmin, xmax}] f wird als Funktion von x auf $[xmin, xmax]$ gezeichnet

Plot[{f₁, f₂, ...},{x, xmin, xmax}] Es werden mehrere Funktionen gezeichnet

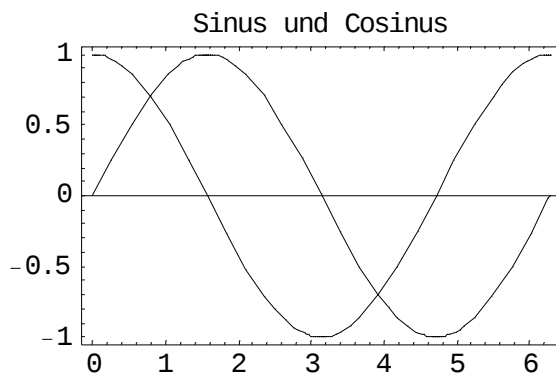
3.8.2 Optionen

Viele Eigenschaften einer Grafik können über **Optionen** beeinflusst werden (z.B. Skalierung, Stützstellen). Nach der Mathematica-Syntax werden Optionen generell als letzte Argumente eines Funktionsaufrufs in einer Folge von Regeln angegeben. Für die Plot-Funktion lautet also die Syntax:

$$\text{Plot}[\{f_1, f_2, \dots\}, \{x, x_{\min}, x_{\max}\}, \text{opt1} \rightarrow \text{wert1}, \text{opt2} \rightarrow \text{wert2}, \dots]$$

In folgendem Beispiel wird ein Rahmen eingeschaltet und ein Titel ergänzt:

```
In[1]:= Plot[{Sin[x], Cos[x]}, {x, 0, 2 Pi}, Frame -> True,
PlotLabel -> "Sinus und Cosinus"]
```

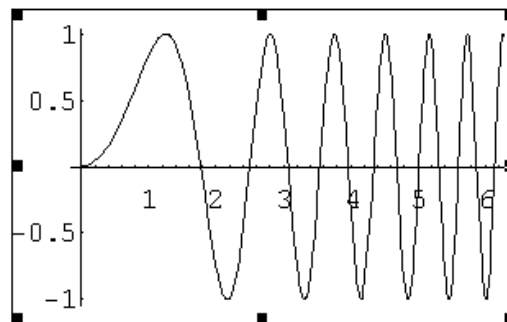


Out[1] = - Graphics -

Eine vollständige Liste aller Optionen liefert der **Help Browser** unter **Graphics and Sound > Basic Options bzw. 3D Options** etc.

3.8.3 Größe einer Grafik verändern

Nach dem Anklicken einer Grafik wird ein Rahmen mit Anfassern eingeblendet. Wie üblich kann das Seitenverhältnis bei der Größenänderungen erhalten oder modifiziert werden:



Während der Größenänderung werden Breite und Höhe in der Statuszeile des Mathematica-Fensters angezeigt.

3.8.4 Grafiken verschieben

Eine markierte Grafik kann innerhalb ihrer Zelle, die nötigenfalls nach unten beliebig vergrößert wird, per Maus-Drag-and-Drop verschoben werden. Während der Bewegung werden die aktuellen Koordinaten der linken, oberen Ecke des Grafikrahmens in der Statuszeile des Mathematica-Fensters angezeigt.

3.8.5 Grafiken (in andere Anwendungen) übertragen

3.8.5.1 Übertragung via Zwischenablage

Unter Windows können Sie folgendermaßen vorgehen, um eine Mathematica-Grafik in eine andere Anwendung zu übertragen:

- Markieren Sie die Grafik.
- Übertragen Sie die Grafik mit **Edit > Copy** bzw. mit **<Strg><C>** in die Zwischenablage.
- Wechseln Sie in die Zielanwendung, und wählen Sie dort den Menübefehl **Bearbeiten > Inhalte einfügen**.
- Nun können Sie zwischen mehreren Datenformaten wählen, die Mathematica in der Zwischenablage erstellt hat. Nach meiner Erfahrung klappt das Einfügen des **Grafik**-Formates am besten.

Sie können eine Abbildung auch innerhalb des selben Notizbuchs oder in ein anderes Notizbuch kopieren.

Unter UNIX oder Mac-OS ist ein analoges Vorgehen zu wählen (siehe Tastatur-Entsprechungen im Anhang).

3.8.5.2 Export in eine Datei

Wenn Sie die Grafik in einem bestimmten Format in eine Datei schreiben möchten, müssen Sie folgendermaßen vorgehen:

- Markieren Sie die Abbildung.
- Wählen Sie den Menübefehl **Edit > Save Selection As...**
- Wählen Sie ein Grafik-Dateiformat

Unter Windows stehen zur Auswahl:

- Encapsulated PostScript File (*.EPS)
- Adobe Illustrator File (*.AI)
- Microsoft Windows Bitmap (*.BMP)
- Enhanced Metafile (*.EMF)
- Microsoft Windows Metafile (*.WMF)

Noch reichhaltiger ist das Angebot auf dem Macintosh:

- EPS
- Adobe Illustrator
- Bitmap (MGF)
- PICT
- Bitmap PICT
- PICT with Embedded Postscript
- PICS
- Quicktime

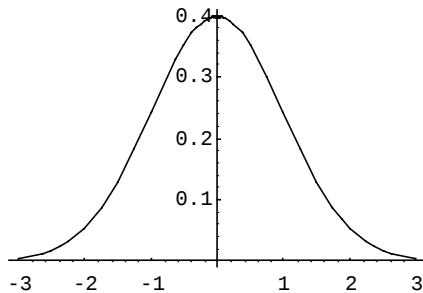
Unter UNIX ist die Auswahl etwas kleiner:

- EPS
- Adobe Illustrator

3.8.6 Diagramme in Variablen speichern, abrufen und modifizieren

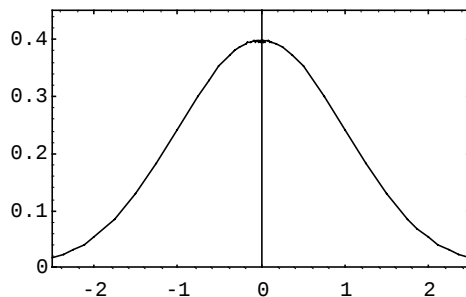
Jeder Plot wird von Mathematica gespeichert und kann später mit dem Befehl **Show** erneut gezeichnet werden. Dabei können nachträglich Optionen verwendet werden, z.B.:

```
In[1]:= Plot[Exp[-x^2/2]/Sqrt[2 Pi],{x, -3,3}]
```



```
Out[1]= -Graphics-
```

```
In[2]:= Show[%,Frame->True, PlotRange->{{-2.5,2.5},{0,0.45}}]
```



```
Out[2]= -Graphics-
```

Wie *In[2]* demonstriert, kann mit **PlotRange** der anzuzeigende x- und y-Achsenbereich nachträglich beliebig abgeändert werden. Dabei greift Mathematica allerdings stets auf vorhandene Stützstellen zurück, d.h. bei starker Vergrößerung werden die Geradensegmente sichtbar, aus denen eine Kurve zusammengesetzt ist. Um die Anzahl der Stützstellen mit der Option **PlotPoints** zu erhöhen, muß man die Plot-Funktion erneut verwenden. Im **Show**-Befehl stehen also nicht alle Grafik-Optionen zur Verfügung.

Um das spätere Abrufen zu erleichtern, können Diagramme auch in Variablen gespeichert werden. Dies ist auch mit per **Show** reproduzierten Diagrammen möglich, z.B.:

```
In[3]:= alt = Plot[Exp[-x^2/2]/Sqrt[2 Pi],{x, -3,3}]
```

```
...
```

```
In[4]:= neu = Show[alt, Frame->True, FrameLabel->"Dichte"]
```

```
...
```

3.8.7 Diagramme kombinieren

Sie können mehrere Plots in einem gemeinsamen, von Mathematica neu angepaßten Koordinatensystem zeichnen lassen, z.B.:

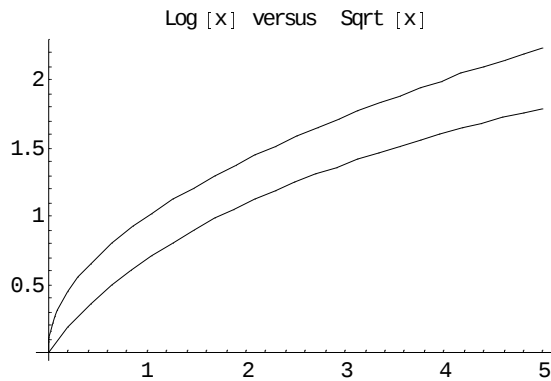
```
In[1]:= p1 = Plot[Sqrt[x],{x,0,5}]
```

```
...
```

```
In[2]:= p2 = Plot[Log[x+1],{x,0,5}]
```

```
...
```

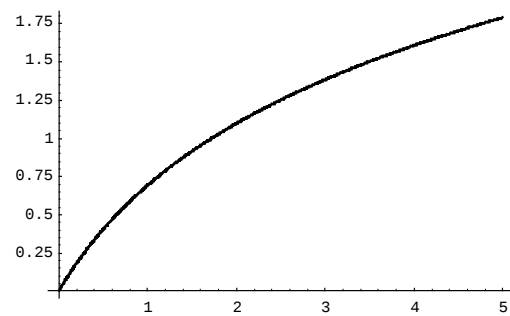
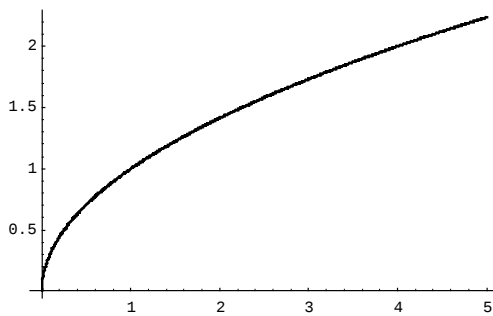
```
In[3]:= Show[p1,p2,PlotLabel->"Log[x] versus Sqrt[x]"]
```



```
Out[3]= -Graphics-
```

Mathematica unterstützt aber auch die gleichzeitige Präsentation mehrerer Diagramme in jeweils eigenen Koordinatensystemen, z.B.:

```
In[4]:= Show[GraphicsArray[{p1, p2}, GraphicsSpacing -> {0.3,0}],  
            Ticks -> Automatic]
```



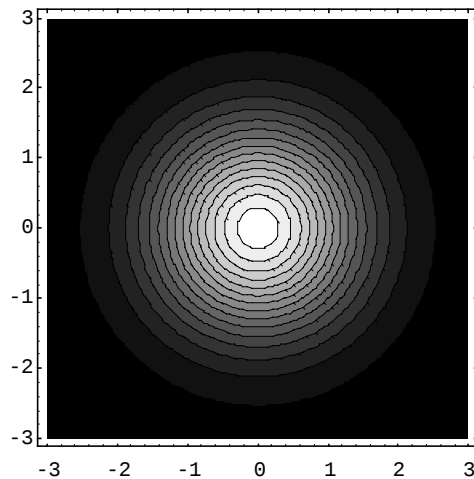
```
Out[4]= -Graphics-
```

Die im **Show**-Befehl angegebene Option gilt für beide Diagramme. Sie können die einzelnen Diagramme auch untereinander bzw. in Matrixform anordnen (siehe Hilfe zur Funktion **GraphicsArray**).

3.8.8 Konturdiagramme

Für Funktionen vom $\mathbb{R}^2$ in den $\mathbb{R}^1$ werden Bereiche mit Funktionswerten innerhalb bestimmter Intervalle durch Grenzlinien und Graustufen (je größer der Funktionswert, desto heller) markiert, z.B.:

```
In[1]:= ContourPlot[Exp[-(x^2+y^2)/2]/Sqrt[2 Pi],{x,-3,3}, {y,-3,3},
PlotPoints->100,Contours->15]
```



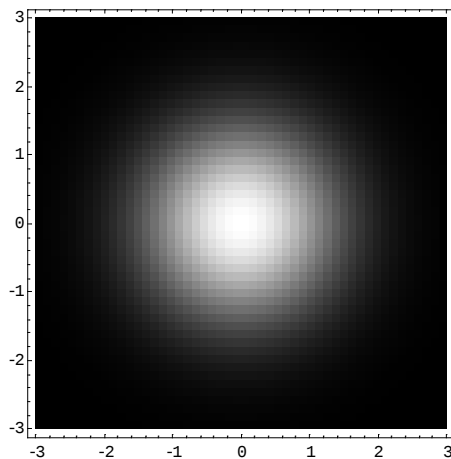
```
Out[1]= -Graphics-
```

Mit **Contours->15** werden auf der z-Achse 15 Intervalle gebildet, die in der Abbildung jeweils durch eine Graustufe dargestellt werden. Mit **PlotPoints->100** wird das Gitternetz mit den Stützstellen verfeinert, für die jeweils das zugehörige z-Intervall bzw. die Graustufe ermittelt wird.

3.8.9 Dichtediagramme

Dichtediagramme sind ebenfalls für Funktionen vom $\mathbb{R}^2$ in den $\mathbb{R}^1$ geeignet. Die Argumentenebene wird in Quadrate eingeteilt, die mit wachsendem mittleren Funktionswert heller dargestellt werden, z.B.:

```
In[1]:= DensityPlot[Exp[-(x^2+y^2)/2]/Sqrt[2 Pi],{x,-3,3}, {y,-3,3},
PlotPoints->50, Mesh->False]
```



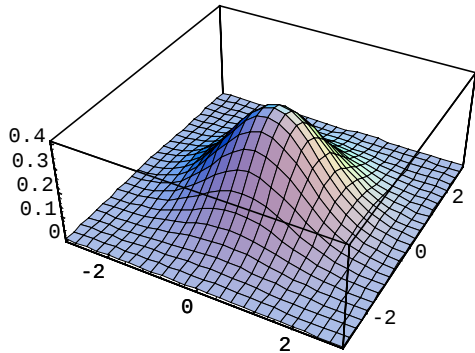
```
Out[1]= -Graphics-
```

Mit **PlotPoints->50** wird das Gitternetz verfeinert, und mit **Mesh->False** werden die Gitterlinien abgeschaltet.

3.8.10 3D-Oberflächendiagramme

Nun kommen wir zu der meist attraktivsten Darstellung für Funktionen vom $\mathbb{R}^2$ in den $\mathbb{R}^3$, der räumlichen Darstellung der Funktionsoberfläche, z.B.:

```
In[1]:= Plot3D[Exp[-(x^2+y^2)/2]/Sqrt[2 Pi],{x,-3,3}, {y,-3,3},
PlotPoints -> 25]
```



```
Out[1]= -Graphics-
```

Zu den Grafik-Optionen, die auch bei zweidimensionalen Grafiken verwendet werden können (z.B. PlotPoints, PlotRange, AxesLabel), kommen nun spezielle 3D-Optionen hinzu. Besonders wichtig ist der 3D-Aussichtspunkt, der sehr bequem festgelegt werden kann:

3.8.11 3D-Aussichtspunkt festlegen

Sie können den Aussichtspunkt festlegen, von dem aus die gezeichneten Objekte betrachtet werden sollen. Die Koordinaten des Aussichtspunktes beziehen sich auf ein Koordinatensystem, das vom dreidimensionalen Rahmen um das zu zeichnende Objekt abhängt:

- Die längste Seite des Rahmens hat die Länge 1.
- Der Mittelpunkt des Rahmens hat die Koordinaten $\{0,0,0\}$.

Die Voreinstellung "direkt vorn" hat die kartesischen Koordinaten $\{x, y, z\} = \{1.3, -2.4, 2\}$. Zur Festlegung des 3D-Aussichtspunktes dient die Option **ViewPoint**, die in den Mathematica-Versionen für Windows und Macintosh nach dem Menübefehl **Input > 3D ViewPoint Selector...** mit folgender Dialogbox produziert werden kann:

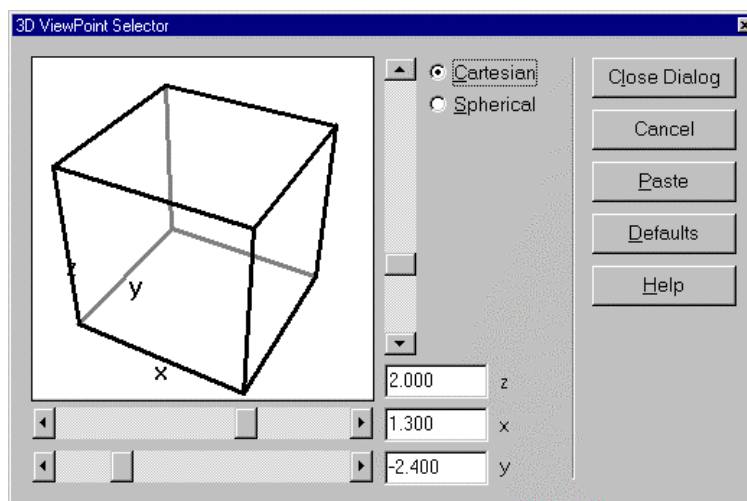


Abbildung 4 Dialogbox zur Festlegung des 3D-Aussichtspunktes

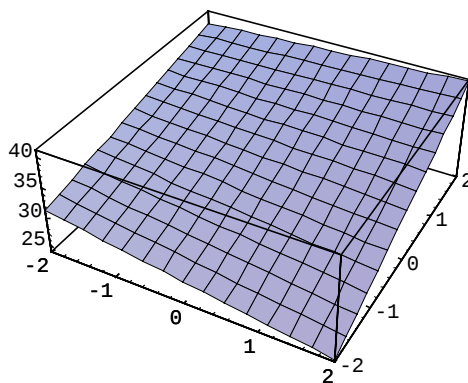
Der Aussichtspunkt kann mit einer der folgenden Methoden gewählt werden:

- Würfel per Maus anpacken und drehen
- Rollbalken benutzen
- Numerische Eingabe nach vorheriger Wahl des gewünschten Koordinatensystems (kartesisch oder sphärisch).

Dem gewählten Aussichtspunkt entspricht eine ViewPoint-Option. Nach einem Mausklick auf den **Paste**-Schalter überträgt Mathematica diese Option in das Notizbuch ab dem aktuellen Einfügebepunkt. Damit bietet sich folgende Vorgehensweise an:

Erstellen Sie das gewünschte 3D-Oberflächendiagramm, z.B.:

```
In[1]:= Plot3D[33.92-0.64x + 3.03 y + 0.54 x y , {x,-2,2}, {y,-2,2}]
```

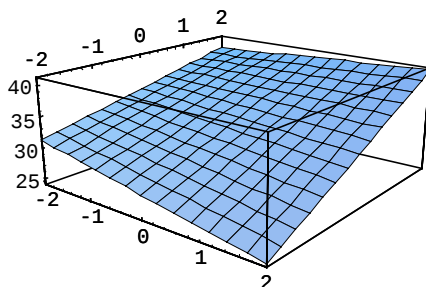


```
Out[1]= -Graphics-
```

Bereiten Sie ein **Show**-Kommando zur Aufnahme der ViewPoint-Option vor, z.B.: **Show[%, |]**.

(Der senkrechte Strich ist nicht einzugeben, sondern stellt die Einfügemarke dar.) Hier erscheint die ViewPoint-Option, wenn Sie im 3DViewPoint Selector auf den **Paste**-Schalter klicken, z.B.:

```
In[2]:= Show[%, ViewPoint->{1.6, -2, 0.8}]
```



```
Out[2]= -Graphics-
```

3.8.12 Gestaltung der Beschriftungen in Grafiken

Über die globalen Variablen **\$TextStyle** können die Voreinstellungen zu diversen Schriftmerkmale für die Texte in Grafiken geändert werden, z.B.:

```
In[1]:= $TextStyle={FontFamily->"Helvetica-Bold",FontSize->16}
```

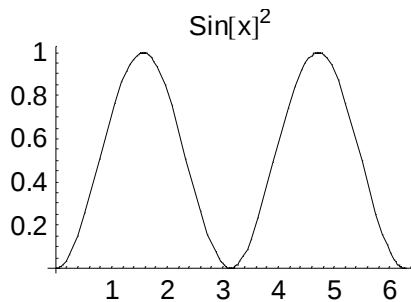
```
Out[1]= {FontFamily -> Helvetica-Bold, FontSize -> 16}
```

Als Schriftfamilie können wohl die PostScript-Fonts angegeben werden. Voreinstellung ist Courier. Weitere Textstile:

Stil	Typische Werte
FontSlant	Plain, Italic, Oblique
FontWeight	Plain, Bold

Mit der globalen Variablen **\$FormatType** kann die Schreibweise für mathematische Ausdrücke zwischen StandardForm (Voreinstellung) und TraditionalForm umgeschaltet werden, z.B.:

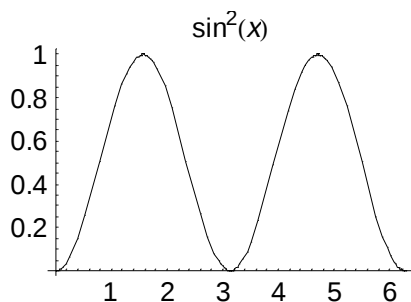
```
In[2]:= Plot[Sin[x]^2, {x,0,2 Pi},PlotLabel -> Sin[x]^2]
```



```
Out[2]= -Graphics-
```

```
In[3]:= $FormatType=TraditionalForm
```

```
In[4]:= Plot[Sin[x]^2, {x,0,2 Pi},PlotLabel -> Sin[x]^2]
```



```
Out[4]= -Graphics-
```

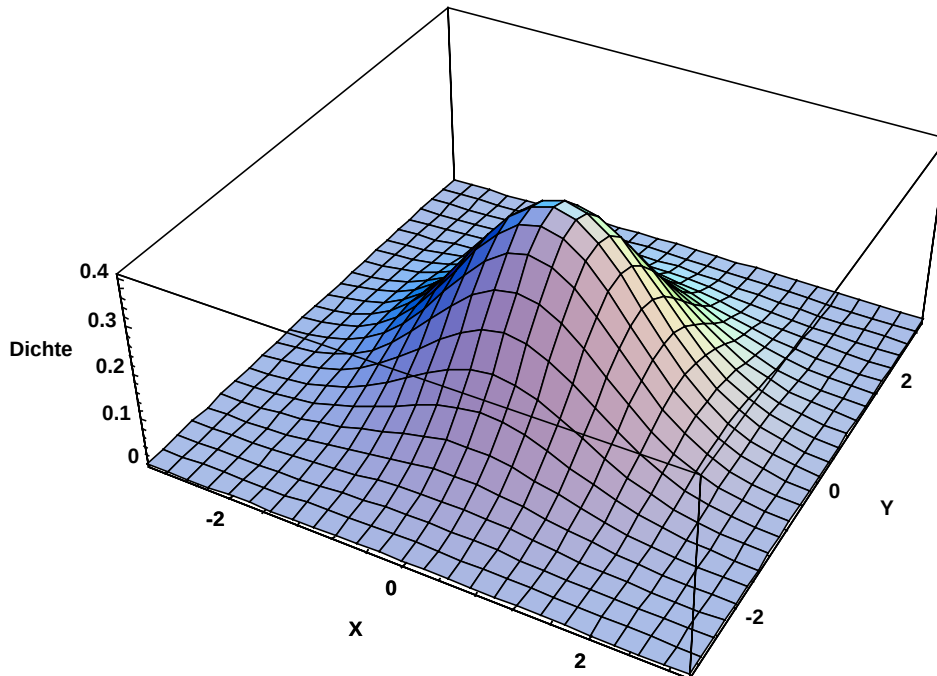
TextStyle und FormatType können auch als Optionen von Grafikbefehlen verwendet werden, um die Schriftmerkmale in einer einzelnen Grafik zu beeinflussen, z.B.:

```
In[5]:= Plot[Sin[x]^2, {x,0,2 Pi},PlotLabel -> Sin[x]^2,
           FormatType->TraditionalForm]
```

Mit Hilfe der Funktion **StyleForm** können Sie Schriftattribute für einzelne Texte festlegen, z.B.

```
In[6]:= Plot3D[Exp[-(x^2+y^2)/2]/Sqrt[2Pi],{x,-3,3}, {y,-3,3},
  PlotPoints->25, AxesLabel->{"X", "Y","Dichte"},
  PlotLabel->StyleForm["Dichte der 2-dim. Standard-NV",
  FontFamily->"Times-Bold", FontSize->16]]
```

Dichte der 2-dim. Standard-NV

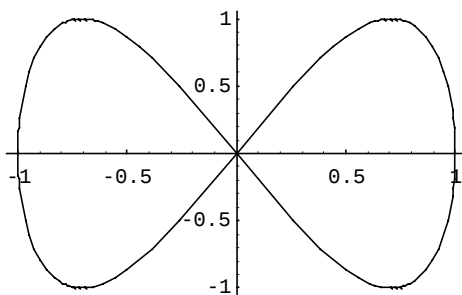


Out[6]= -Graphics-

3.8.13 Parametrische Diagramme

Im zweidimensionalen Fall werden die x - und die y -Koordinate eines Punktes als Funktion eines dritten Parameters dargestellt, z.B.:

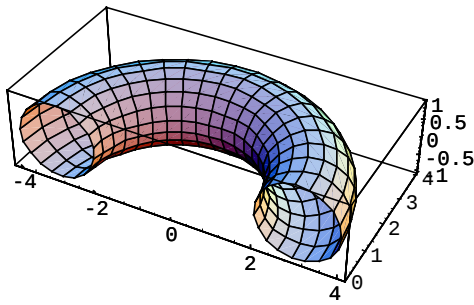
```
In[1]:= ParametricPlot[{Sin[t], Sin[2t]}, {t,0,2Pi}]
```



Out[1]= -Graphics-

Besonders eindrucksvoll sind parametrische 3D-Plots, z.B.:

```
In[2]:= ParametricPlot3D[{Cos[t](3 + Cos[u]), Sin[t](3 + Cos[u]),
    Sin[u]}, {t,0,Pi}, {u,0,2Pi}]
```

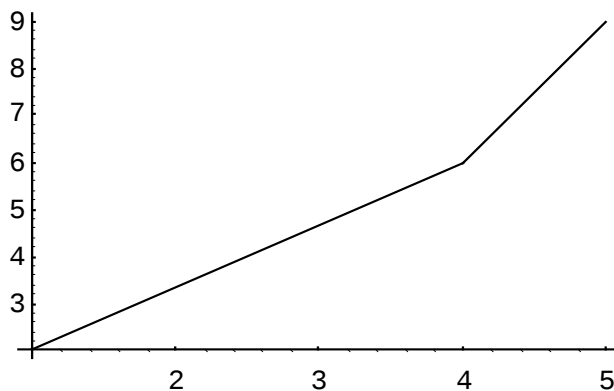


```
Out[2]= -Graphics-
```

3.8.14 Datenlisten zeichnen

Mathematica kann nicht nur Diagramme von Funktionen erstellen, sondern auch Datenlisten 2- oder 3-dimensional grafisch darstellen. Wir beschränken uns auf einfache Liniendiagramme, die mit dem Befehl **ListPlot** gezeichnet werden:

```
In[1]:= ListPlot[{{1,2}, {4,6}, {5,9}}, PlotJoined->True]
```

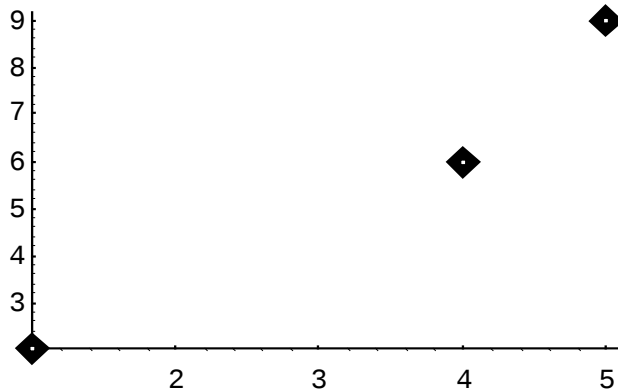


```
Out[1]= -Graphics-
```

3.8.15 Grafikanweisungen

Wenn Sie im letzten Beispiel die vorgegebenen Punkte kräftiger markieren wollen, müssen Sie eine **Grafikanweisung** (graphics directive) verwenden, z.B. **AbsolutePointSize**, die mit Hilfe der **PlotStyle**-Option eingefügt werden kann:

```
In[1]:= ListPlot[{{1,2}, {4,6}, {5,9}},
  PlotStyle -> AbsolutePointSize[6]]
```



```
Out[1]= -Graphics-
```

Über das nicht ganz triviale Zusammenspiel von Grafikanweisungen und Grafikoptionen bei der Gestaltung einer Abbildung informiert das Mathematica-Buch zur Version 3 auf den Seiten 466ff.

3.9 Numerische Mathematik

Die numerische Mathematik ist in diesem Manuskript leider etwas zu kurz geraten, z.B. fehlt das Thema Interpolation komplett.

3.9.1 Exakte Ergebnisse und numerische Approximationen

Wie wir bereits wissen, bevorzugt Mathematica Ausdrücke in einer exakten, symbolischen Form. Wenn eine numerische Approximation gewünscht ist, muß diese in der Regel explizit angefordert werden, z.B.:

```
In[1]:= 10!^(1/2)
```

```
Out[1]= 720 Sqrt[7]
```

```
In[2]:= N[%, 10]
```

```
Out[2]= 1904.940944
```

Analog versuchen Funktionen wie **Integrate** stets, exakte Ergebnisse abzuliefern. Ist dies nicht möglich, beschränken sie sich auf eine Vereinfachung des Mathematica-Ausdrucks oder tun gar nichts, und dem Anwender bleibt nur der Operator **N**, um wenigstens numerische Lösungen zu erzwingen.

Die Funktion **NIntegrate** beschränkt sich demgegenüber von vornherein auf numerische Ergebnisse, z.B. (vgl. Abschnitt 3.6.2.3):

```
In[3]:= Integrate[Exp[x^2/-2], {x, -Infinity, 2}]
```

```
Out[3]=  $\sqrt{\frac{\pi}{2}} (1 + \text{Erf}[\sqrt{2}])$ 
```

```
In[4]:= N[%]
```

```
Out[4]= 2.4496
```

```
In[5]:= NIntegrate[Exp[x^2/-2], {x, -Infinity, 2}]
```

```
Out[5]= 2.4496
```

Analog gibt es zu den Funktionen **Sum** und **Product** die numerischen Varianten **NSum** und **NProduct**.

3.9.2 Numerische Lösung von Gleichungen

3.9.2.1 Polynom-Gleichungen

Der Funktion **Solve** zum Lösen von Polynom-Gleichungen entspricht die numerische Variante **NSolve**, die für jede Polynom-Gleichung alle approximativen Lösungen liefert, z.B. (vgl. Abschnitt 3.6.6.1):

```
In[1]:= Solve[x^6-x^5==6,x]
```

```
Out[1]= {{x -> Root[-6 - #1^5 + #1^6&, 1]}, {x -> Root[-6 - #1^5 + #1^6&, 2]}, {x -> Root[-6 - #1^5 + #1^6&, 3]},
         {x -> Root[-6 - #1^5 + #1^6&, 4]}, {x -> Root[-6 - #1^5 + #1^6&, 5]}, {x -> Root[-6 - #1^5 + #1^6&, 6]}}
```

```
In[2]:= N[%]
```

```
Out[2]= {{x -> -1.22}, {x -> 1.59015}, {x -> -0.537119 - 1.13651 I}, {x -> -0.537119 + 1.13651 I},
         {x -> 0.852045 - 1.10965 I}, {x -> 0.852045 + 1.10965 I}}
```

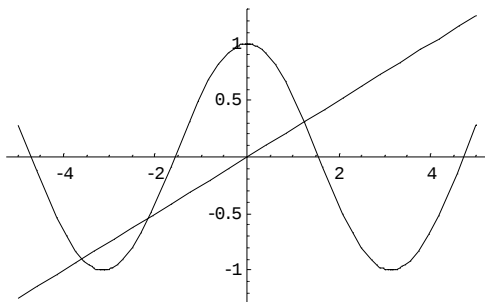
```
In[3]:= NSolve[x^6-x^5==6,x]
```

```
Out[3]= {{x -> -1.22}, {x -> -0.537119 - 1.13651 I}, {x -> -0.537119 + 1.13651 I}, {x -> 0.852045 - 1.10965 I},
         {x -> 0.852045 + 1.10965 I}, {x -> 1.59015}}
```

3.9.2.2 Transzendente Gleichungen

Da es bei transzendenten Funktionen keine systematische Methode zum Auffinden aller numerischen Lösungen gibt, führt der Operator **N** hier nicht viel weiter (vgl. Abschnitt 3.6.6.2). Wir wollen in einem Beispiel die Schnittpunkte der beiden Funktionen $\cos[x]$ und $x/4$ suchen lassen:

```
In[1]:= Plot[{Cos[x], x/4}, {x, -5, 5}]
```



```
Out[1]= -Graphics-
```

```
In[2]:= Solve[Cos[x]==x/4,x]
```

```
Solve::tdep: The equations appear to involve transcendental functions of the
variables in an essentially non-algebraic way.
```

```
Out[2]= Solve[Cos[x] == x/4, x]
```

```
In[3]:= N[%]
```

```
Solve::tdep: The equations appear to involve transcendental functions of the
variables in an essentially non-algebraic way.
```

```
Out[3]= Solve[Cos[x] == 0.25 x, x]
```

In dieser Situation steht die Funktion **FindRoot** zur Verfügung, die in Abhängigkeit von einem Startwert nach numerischen Lösungen sucht:

```
In[4]:= FindRoot[Cos[x]==x/4, {x,1}]
```

```
Out[4]= {x → 1.25235}
```

Während sich **Solve** bemüht, *alle* Lösungen einer Gleichung zu finden, erhält man mit **FindRoot** nur eine Lösung, die vom gewählten Startwert des Verfahrens abhängt, z.B.:

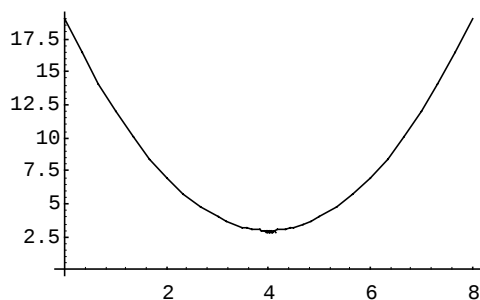
```
In[5]:= FindRoot[Cos[x]==x/4, {x,-2}]
```

```
Out[5]= {x → -2.13333}
```

3.9.3 Numerische Optimierung

Zur numerischen Optimierung ohne Nebenbedingungen steht die Funktion **FindMinimum** zur Verfügung, die an einem einfachen Beispiel demonstriert werden soll:

```
In[1]:= Plot[19 - 8*x + x^2, {x,0,8}]
```



```
Out[1]= -Graphics-
```

```
In[2]:= FindMinimum[19 - 8*x + x^2, {x,2}]
```

```
Out[2]= {3., {x → 4.}}
```

Für die Optimierung **mit Nebenbedingungen** bietet Mathematica die Funktion **ConstrainedMin**, die Unter Verwendung des berühmten Kuh-Schaf-Beispiels demonstriert werden soll (vgl. Hettich 1989):

In einem landwirtschaftlichen Betrieb mit Kühen und Schafen sind folgende Ressourcen vorhanden.

Kuhstall	für max. 50 Kühe
Schafstall	für max. 200 Schafe
Weide	72 Morgen
Arbeitszeit	10000 Stunden

Pro gehaltenem Tier werden folgende Ressourcen verbraucht:

	Kuh	Schaf
Platz im Kuhstall	1	0
Platz im Schafstall	0	1
Weide	1	0,2
Arbeitszeit	150	25

Pro Tier werden folgende Gewinne gemacht:

Kuh	250
Schaf	45

Wenn x für die Anzahl gehaltener Kühe und y für die Anzahl gehaltener Schafe steht, so ist die folgende Zielfunktion bei gegebenen Ressourcen (= Nebenbedingungen) zu maximieren:

$$250x + 45y$$

Bei derartigen Problemen (lineare Zielfunktion, lineare Nebenbedingungs-Ungleichungen) wendet Mathematica die Methoden der **Linearen Programmierung** an und findet dabei in jedem Fall das globale Minimum:

```
In[3]:= ConstrainedMin[-(250x + 45y), {x <= 50, y <= 200,
      x + 0.2 y <= 72, 150 x + 25 y <= 10000}, {x, y}]
```

```
Out[3]= {-17200., {x -> 40., y -> 160.}}
```

3.10 Funktionen und Programme

Einige Hinweise vorab:

- In späteren Abschnitten folgen noch Themen von hoher Relevanz für das Programmieren mit Mathematica (z.B. Lesen und Schreiben von externen Dateien).
- Wesentliche Möglichkeiten (z.B. das objektorientierte Programmieren mit Mathematica), können in diesem Manuskript nicht behandelt werden.

3.10.1 Funktionen definieren und verwenden

In folgendem Beispiel wird die Funktion f definiert, welche zu jedem Argument das Quadrat liefert:

```
In[1]:= f[x_] := x^2
```

Mathematica erzeugt *keine* Ausgabezeile zur Bestätigung der Definition.

Bei der Syntax ist zu beachten:

- Das Zuweisungszeichen ("=") wird durch das, auch in der Mathematik übliche, Definitionszeichen (":=") ersetzt.
- Auf der **linken** Seite der Definitionsgleichung ist dem Argument ein **Unterstrich** anzufügen.
- Die benutzerdefinierte Funktionsbezeichnung sollte mit einem Kleinbuchstaben beginnen, um Verwechslungen mit eingebauten Mathematica-Funktionen zu vermeiden.

Die neu definierte Funktion kann als Argumente Zahlen und auch Ausdrücke mit Symbolen verarbeiten:

```
In[2]:= f[2]
```

```
Out[2]= 4
```

```
In[3]:= f[3x+x^2]
```

```
Out[3]= (3 x + x^2)^2
```

Sie können sich jederzeit über die aktuelle Definition eines Symbols informieren:

```
In[4]:= ?f
```

```
Global`f
f[x_] := x^2
```

Mit der folgenden Anweisung wird die Funktionsdefinition für f wieder gelöscht:

```
In[5]:= Clear[f]
```

Wie bei der Definition gibt Mathematica auch beim Löschen *keine* Bestätigung.

Das **Clear**-Kommando löscht auch die zugewiesenen Werte von Variablen. Ebenso wie überflüssig gewordene Variablen-Definitionen sollten auch obsoletere Funktions-Definitionen sofort gelöscht werden, damit sie nicht später für Verwirrung sorgen.

Clear arbeitet weniger radikal als der verwandte Befehl **Remove**, den wir oben bereits kennengelernt haben:

- Clear löscht den Wert bzw. die Definition eines Symbols. Jedoch bleibt das Symbol samt Attributen erhalten.

- Remove beseitigt ein Symbol vollständig.

Die Funktion im folgenden Beispiel baut aus n Faktoren ein Polynom in x auf:

```
In[6]:= expro[n_]:=Expand[Product[x+i, {i,1,n}]]
```

```
In[7]:= expro[2]
```

```
Out[7]= 2 + 3 x + x^2
```

Soll nun auch noch der Koeffizient zu einem bestimmten Grad des per **expro** erzeugten Polynoms bestimmt werden, können wir entweder eine Verschachtelung von Funktionsaufrufen konstruieren, oder von der Möglichkeit Gebrauch machen, bei der Definition einer Funktion **mehrere** Mathematica-Kommandos anzugeben, die durch Strichpunkte getrennt und insgesamt mit runden Klammern umgeben werden müssen, z.B.:

```
In[8]:= cep[n_,k_] := (t=expro[n]; Coefficient[t, x^k])
```

Die Variable t wird benötigt, um das Ergebnis des ersten Ausdrucks in den zweiten einsetzen zu können. Das Symbol "%" (Rückgriff auf die letzte Ausgabe) erfüllt diesen Zweck nicht, weil eine benutzerdefinierte Funktion nur *eine* Ausgabe nach Auswertung aller Ausdrücke liefert.

Die Mathematica-Funktion **Coefficient** liefert zu einem Polynom den Koeffizienten des gewünschten Grades. Für $n = 2$ und $k = 1$ errechnet unsere Funktion **cep** z.B.:

```
In[9]:= cep[2,1]
```

```
Out[9]= 3
```

Wie das Beispiel **cep** verdeutlicht, können benutzerdefinierte Funktionen als Funktions-**Unterprogramme** aufgefaßt werden. Folglich ist wünschenswert, die darin verwendeten Variablen **lokal** zu machen, damit sich keine Nebeneffekte ergeben. Dazu bietet Mathematica die Möglichkeit, ein **module** zu erstellen:

```
In[10]:= mcep[n_,k_] := Module[{u},u=expro[n];Coefficient[u, x^k]]
```

mcep und **cep** verhalten sich identisch:

```
In[11]:= mcep[2,1]
```

```
Out[11]= 3
```

Die zur Definition von **mcep** verwendete Variable u ist aber im Unterschied zur Variablen t , die bei der **cep**-Definition benötigt wurde, nicht global, d.h. außerhalb der Funktion nicht bekannt:

```
In[12]:= t
```

```
Out[12]= 2 + 3 x + x^2
```

```
In[13]:= u
```

```
Out[13]= u
```

3.10.2 Bedingte Anweisungen

Mit dem If-Kommando können Fallunterscheidungen vorgenommen werden, z.B.:

```
In[1]:= x=-2;  
If[x>=0, y=x, y=-x]
```

```
Out[1]= 2
```

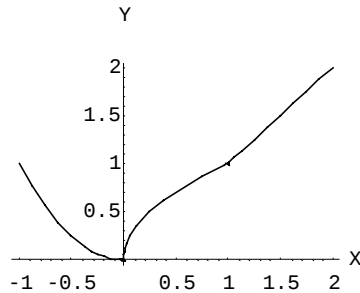
Die allgemeine Syntax:

$\text{If}[\text{ausdruck}, \text{dann}, \text{sonst}]$ Ist der Ausdruck wahr, wird <i>dann</i> ausgewertet, anderenfalls <i>sonst</i> .
--

Bei Funktionsdefinitionen mit Fallunterscheidung kann oft die Funktion **Which** verwendet werden, z.B.:

```
In[2]:= f[x_] := Which[x<0, x^2, x<1, Sqrt[x], x>=1, x]
```

```
In[3]:= Plot[f[x], {x, -1, 2}, AxesLabel -> {X, Y}]
```



```
Out[3]= -Graphics-
```

Die allgemeine Syntax der **Which**-Funktion:

$\text{Which}[\text{test1}, \text{wert1}, \text{test2}, \text{wert2}, \dots]$ Mathematica ermittelt für jedes Argument den ersten wahren <i>test</i> und liefert als Funktionswert den zugehörigen <i>wert</i> .
--

3.10.3 Schleifen-Konstruktionen

Die DO-Schleife hat in Mathematica die folgende allgemeine Syntax:

$\text{Do}[\text{ausdruck}, \{i, \text{imin}, \text{imax}, di\}]$	Der Ausdruck wird für $i = \text{imin}$ bis imax bei Schrittweite di ausgewertet.
---	--

Ist $\text{imin} = di = 1$ kann eine vereinfachte Syntax verwendet werden:

$\text{Do}[\text{ausdruck}, \{i, \text{imax}\}]$	Der Ausdruck wird für $i = 1$ bis imax ausgewertet.
--	--

In folgendem Beispiel wird mit Hilfe der Do-Schleife eine Funktion definiert, welche die Summe der quadrierten natürlichen Zahlen von 1 bis n ausgibt:

```
In[1]:= sumq1n[n_] := Module[{s}, s=0; Do[s=s+i^2, {i, n}]; s]
```

```
In[2]:= sumq1n[10]
```

```
Out[2]= 385
```

Weil die Do-Schleife keine Ausgabe erzeugt, wurde am Ende der Moduldefinition ein Kommando angehängt, das den aktuellen Wert der Variablen s ausgibt.

3.10.4 Ausgeben von Werten

Die Mathematica-Funktion **Print** dient dazu, die Werte von Ausdrücken auszugeben. In folgendem Beispiel geschieht dies innerhalb einer DO-Schleife:

```
In[1]:= Do[Print[n, " ", N[Log[n]/5]], {n,5}]
1  0
2  0.138629
3  0.219722
4  0.277259
5  0.321888
```

Nach der Ausgabe eines Print-Befehls wird automatisch eine neue Zeile begonnen. Innerhalb einer Print-Zeile können die Ausdrücke durch Leerzeichen getrennt werden.

Häufig ist es sinnvoll eine Liste mit Ergebnissen zu erzeugen, die anschließend weiterverarbeitet werden kann. Dazu dient die **Table**-Funktion:

```
In[2]:= res=Table[N[Log[n]/5], {n,5}]
Out[2]= {0, 0.138629, 0.219722, 0.277259, 0.321888}

In[3]:= Exp[res 5]
Out[3]= {1, 2., 3., 4., 5.}
```

3.11 Listen und Matrizen

Mathematica-Listen stellen Verallgemeinerungen von mehreren Standardbegriffen der Mathematik und der Informatik dar. In diesem Abschnitt lernen Sie die allerwichtigsten Operationen mit Listen bzw. Matrizen kennen. Eine vollständige Liste findet sich im Help Browser unter **Lists and Matrices**.

3.11.1 Erstellen von Wertetabellen

Wir haben Listen schon in Abschnitt 3.3.3 kennengelernt und z.B. gesehen, daß man Mathematica-Funktionen auf Listen von Objekten anwenden kann, wobei die Funktionen in der Regel elementenweise arbeiten, z.B. der Ableitungsoperator **D**:

```
In[1]:= Table[x^i, {i,0,5}]
Out[1]= {1, x, x^2, x^3, x^4, x^5}

In[2]:= D[%,x]
Out[2]= {0, 1, 2 x, 3 x^2, 4 x^3, 5 x^4}
```

Hinweis: Sollten Sie statt *Out[1]* eine andere Ausgabe erhalten haben, liegt dies an einer noch bestehenden Definition des Symbols *x*, die Sie mit „*x = .*“ aufheben können.

Die in obigem Beispiel verwandte **Table**-Funktion ist uns schon in Abschnitt 3.10.4 begegnet. Wir wollen ihre Möglichkeiten noch näher studieren. Unter Verwendung der normalen Mathematica-Laufindex-Schreibweise können mit **Table** auch **mehrdimensionale Tabellen** erstellt werden, z.B. solche Listen von Listen:

```
In[3]:= Table[x^(i+j), {i,1,4}, {j,1,4}]
Out[3]= {{x^2, x^3, x^4, x^5}, {x^3, x^4, x^5, x^6}, {x^4, x^5, x^6, x^7}, {x^5, x^6, x^7, x^8}}

In[4]:= x={12,10,6,5}; y={4,11,9,9}
Out[4]= {4, 11, 9, 9}
```

```
In[5]:= eh=Table[x[[i]] y[[j]]/33, {i,1,4}, {j,1,4}]
```

```
Out[5]= {{16/11, 4, 36/11, 36/11}, {40/33, 10/3, 30/11, 30/11}, {8/11, 2, 18/11, 18/11}, {20/33, 5/3, 15/11, 15/11}}
```

Mit Hilfe der Funktion **TableForm** können Sie sich eine Liste von Listen in Tabellenform anzeigen lassen:

```
In[6]:= TableForm[eh]
```

```
Out[6]//TableForm=
```

$\frac{16}{11}$	4	$\frac{36}{11}$	$\frac{36}{11}$
$\frac{40}{33}$	$\frac{10}{3}$	$\frac{30}{11}$	$\frac{30}{11}$
$\frac{8}{11}$	2	$\frac{18}{11}$	$\frac{18}{11}$
$\frac{20}{33}$	$\frac{5}{3}$	$\frac{15}{11}$	$\frac{15}{11}$

Die i -te Unterliste einer zweidimensionalen Liste ist natürlich eine eindimensionale Liste, z.B.:

```
In[7]:= eh[[1]]
```

```
Out[7]= {16/11, 4, 36/11, 36/11}
```

Um einzelne Listenelemente herauszuziehen, muß man doppelt indizieren:

```
In[8]:= eh[[1,2]]
```

```
Out[8]= 4
```

3.11.2 Vektoren und Matrizen

Nach den bisherigen Ausführungen dürfte klar sein, daß Vektoren und Matrizen in Mathematica als Listen dargestellt werden:

```
In[1]:= v1={x, y}
```

```
Out[1]= {x, y}
```

```
In[2]:= m1={{a, b}, {c, d}}
```

```
Out[2]= {{a, b}, {c, d}}
```

Mit den Funktionen **ColumnForm** und **MatrixForm** können die aus der Mathematik vertrauten Darstellungen erzeugt werden:

```
In[3]:= ColumnForm[v1]
```

```
Out[3]= x
        y
```

```
In[4]:= MatrixForm[m1]
```

```
Out[4]//MatrixForm=
```

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

3.11.2.1 Operationen mit Matrizen

3.11.2.1.1 "Matrix * Skalar" bzw. "Matrix + Skalar"

Die Operationen "Matrix * Skalar" bzw. "Matrix + Skalar" werden elementenweise ausgeführt:

```
In[1]:= p v1 + q
```

```
Out[1]= {q + p x, q + p y}
```

3.11.2.1.2 Matrixprodukt

In folgendem Beispiel wird die oben definierte Matrix *m1* mit dem Vektor *v1* nachmultipliziert:

```
In[1]:= MatrixForm[m1.v1]
```

```
Out[1]//MatrixForm=
```

$$\begin{pmatrix} a x + b y \\ c x + d y \end{pmatrix}$$

Als Operationszeichen für das Matrixprodukt wird also ein Punkt verwendet.

Die folgende Rechnung zeigt zweierlei:

- Grundsätzlich berechnet Mathematica das Produkt zweier Matrizen nach den üblichen Regeln, d.h. das Element (i, j) der Produktmatrix ist gerade das innere Produkt aus der i -ten Zeile der ersten Matrix mit der j -ten Spalte der zweiten Matrix.
- Beim Produkt eines Vektors mit einer Matrix wird der Vektor situationsadäquat als Zeile oder Spalte behandelt.

```
In[2]:= MatrixForm[v1.m1]
```

```
Out[2]//MatrixForm=
```

$$\begin{pmatrix} a x + c y \\ b x + d y \end{pmatrix}$$

Ein Vektor ist in Mathematica weder Zeile noch Spalte, sondern Liste. Angewandt auf zwei Vektoren liefert daher die Punkt-Operation unabhängig von der Reihenfolge der Argumente stets das *innere* Produkt, z.B.:

```
In[3]:= v2={1,2}
```

```
Out[3]= {1, 2}
```

```
In[4]:= v1 . v2
```

```
Out[4]= x + 2y
```

```
In[5]:= v2 . v1
```

```
Out[5]= x + 2y
```

Wie die obigen Beispiel gezeigt haben, gilt diese „Kommutativität“ in Mathematica keinesfalls generell für die Matrix-Multiplikation.

Wenn Sie für zwei Vektoren das *äußere* Produkt benötigen (Kronecker-Produkt), z.B.:

$$\begin{pmatrix} x \\ y \end{pmatrix} \begin{pmatrix} 1 & 2 \end{pmatrix}$$

müssen Sie die **Outer**-Funktion verwenden:

```
In[6]:= MatrixForm[Outer[Times, v1, v2]]
```

```
Out[6]//MatrixForm=
```

$$\begin{pmatrix} x & 2x \\ y & 2y \end{pmatrix}$$

Natürlich kann Mathematica auch das Kronecker-Produkt zweier Matrizen ausrechnen:

```
In[7]:= m2={{1, 2}, {3, 4}}
```

```
Out[7]= {{1, 2}, {3, 4}}
```

```
In[8]:= MatrixForm[m2]
```

```
Out[8]//MatrixForm=
```

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

```
In[9]:= MatrixForm[Outer[Times, m1, m2]]
```

```
Out[9]//MatrixForm=
```

$$\begin{pmatrix} \begin{pmatrix} a & 2a \\ 3a & 4a \end{pmatrix} & \begin{pmatrix} b & 2b \\ 3b & 4b \end{pmatrix} \\ \begin{pmatrix} c & 2c \\ 3c & 4c \end{pmatrix} & \begin{pmatrix} d & 2d \\ 3d & 4d \end{pmatrix} \end{pmatrix}$$

3.11.2.1.3 Transponierte Matrix

Die Transponierte zur Matrix $m1$ erhält man mit:

```
In[1]:= MatrixForm[Transpose[m1]]
```

```
Out[1]//MatrixForm=
```

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

3.11.2.1.4 Determinante einer Matrix

Die Funktion **Det** liefert die Determinante zur Matrix $m1$:

```
In[1]:= Det[m1]
```

```
Out[1]= -bc + ad
```

3.11.2.1.5 Inverse Matrix

Die Inverse zur Matrix $m1$ erhalten wir so:

```
In[1]:= MatrixForm[Inverse[m1]]
```

```
Out[1]//MatrixForm=
```

$$\begin{pmatrix} \frac{d}{-bc+ad} & -\frac{b}{-bc+ad} \\ -\frac{c}{-bc+ad} & \frac{a}{-bc+ad} \end{pmatrix}$$

3.11.2.1.6 Eigenwerte einer Matrix

Die Funktion **Eigenvalues** liefert uns die Eigenwerte der Matrix $m1$:

In[1]:= Eigenvalues[m1]

$$\text{Out[1]} = \left\{ \frac{1}{2} \left(a + d - \sqrt{a^2 + 4bc - 2ad + d^2} \right), \right. \\ \left. \frac{1}{2} \left(a + d + \sqrt{a^2 + 4bc - 2ad + d^2} \right) \right\}$$

Zum Auffrischen unseres Grundwissens in Linearer Algebra wollen wir das Ergebnis verifizieren:

$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \neq \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ ist Eigenvektor von $m1$, falls gilt:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - \lambda \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \Leftrightarrow \begin{pmatrix} a - \lambda & b \\ c & d - \lambda \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Dann muß aber gelten:

$$\det \begin{pmatrix} a - \lambda & b \\ c & d - \lambda \end{pmatrix} = 0 \Leftrightarrow (a - \lambda)(d - \lambda) - bc = 0 \Leftrightarrow ad - bc - (a + d)\lambda + \lambda^2 = 0$$

Durch Lösen der quadratischen Gleichung erhält man die Eigenwerte. Diese triviale Aufgabe überlassen wir wieder Mathematica:

In[2]:= Solve[a d - b c - (a + d) lambda + lambda^2 == 0, lambda]

$$\text{Out[2]} = \left\{ \left\{ \lambda \rightarrow \frac{1}{2} \left(a + d - \sqrt{a^2 + 4bc - 2ad + d^2} \right) \right\}, \right. \\ \left. \left\{ \lambda \rightarrow \frac{1}{2} \left(a + d + \sqrt{a^2 + 4bc - 2ad + d^2} \right) \right\} \right\}$$

3.11.2.1.7 Eigenvektoren einer Matrix

Mit **Eigenvalues** können wir schließlich auch noch Eigenwerte zu unserer Matrix $m1$ berechnen lassen:

In[1]:= Eigenvectors[m1]

$$\text{Out[1]} = \left\{ \left\{ -\frac{a + d + \sqrt{a^2 + 4bc - 2ad + d^2}}{2c}, 1 \right\}, \right. \\ \left. \left\{ -\frac{a + d - \sqrt{a^2 + 4bc - 2ad + d^2}}{2c}, 1 \right\} \right\}$$

Wir werden diese Lösung in Abschnitt 3.11.2.2 verifizieren.

3.11.2.1.8 Mathematica-Funktionen zum Generieren von Matrizen

In Abschnitt 3.11.1 haben wir schon die Funktion **Table** kennengelernt, die eine Matrix erzeugt, indem ein Ausdruck für bestimmte Indexkombinationen ausgewertet wird, z.B.:

In[1]:= Table[x y, {x, 1, 4}, {y, 1, 4}]

$$\text{Out[1]} = \{\{1, 2, 3, 4\}, \{2, 4, 6, 8\}, \{3, 6, 9, 12\}, \{4, 8, 12, 16\}\}$$

Mit der Funktion **Array** können Sie einen Vektor bzw. eine Matrix mit "indizierten" Variablen erstellen, z.B.:

```
In[2]:= Array[b,4]
```

```
Out[2]= {b[1], b[2], b[3], b[4]}
```

```
In[3]:= MatrixForm[Array[m23,{2,3}]]
```

```
Out[3]//MatrixForm=
```

$$\begin{pmatrix} m23[1, 1] & m23[1, 2] & m23[1, 3] \\ m23[2, 1] & m23[2, 2] & m23[2, 3] \end{pmatrix}$$

Die n -te Einheitsmatrix erhalten Sie mit der Funktion **IdentityMatrix**, z.B.:

```
In[4]:= MatrixForm[IdentityMatrix[3]]
```

```
Out[4]//MatrixForm=
```

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Mit der Funktion **DiagonalMatrix** können Sie aus einer Liste eine Diagonalmatrix erzeugen, z.B.:

```
In[5]:= DiagonalMatrix[{1,2,3}]
```

```
Out[5]= {{1, 0, 0}, {0, 2, 0}, {0, 0, 3}}
```

3.11.2.2 Lösung linearer Gleichungssysteme in Matrizenschreibweise

Mit Hilfe der Mathematica-Funktion **LinearSolve** läßt sich bequem der Lösungsvektor x zu einer Matrix-Gleichung $m \cdot x == b$ ermitteln. Zunächst soll ein **inhomogenes** Gleichungssystem mit numerischen Koeffizienten gelöst werden:

```
In[1]:= a={{1,-2,1,0},{1,-2,0,-1},{0,0,1,1}}
```

```
Out[1]= {{1, -2, 1, 0}, {1, -2, 0, -1}, {0, 0, 1, 1}}
```

```
In[2]:= MatrixForm[a]
```

```
Out[2]//MatrixForm=
```

$$\begin{pmatrix} 1 & -2 & 1 & 0 \\ 1 & -2 & 0 & -1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

```
In[3]:= b={1,2,-1}
```

```
Out[3]= {1, 2, -1}
```

```
In[4]:= LinearSolve[a, b]
```

```
Out[4]= {2, 0, -1, 0}
```

Nun wollen wir ein **homogenes** Gleichungssystem lösen. Dazu prüfen wir nach, ob die in Abschnitt 3.11.2.1.7 behandelte Funktion **Eigenvectors** zum ersten Eigenwert der dortigen Matrix **m** ein korrektes Ergebnis geliefert hat. Zunächst müssen die dortigen Definitionen wieder hergestellt werden. Davon sind auch die Symbole **a** und **b** betroffen, die mittlerweile eine neue Definition als Matrix bzw. Vektor erfahren haben:

In[5]:= a=.

In[6]:= b=.

In[7]:= m={{a, b}, {c, d}}

Out[7]= {{a, b}, {c, d}}

In[8]:= Eigenvalues[m]

Out[8]= $\left\{ \frac{1}{2} \left(a + d - \sqrt{a^2 + 4 b c - 2 a d + d^2} \right), \right.$
 $\left. \frac{1}{2} \left(a + d + \sqrt{a^2 + 4 b c - 2 a d + d^2} \right) \right\}$

In[9]:= koef=m-%[[1]]IdentityMatrix[2]

Out[9]= $\left\{ \left\{ a + \frac{1}{2} \left(-a - d + \sqrt{a^2 + 4 b c - 2 a d + d^2} \right), b \right\}, \right.$
 $\left. \left\{ c, d + \frac{1}{2} \left(-a - d + \sqrt{a^2 + 4 b c - 2 a d + d^2} \right) \right\} \right\}$

In[10]:= MatrixForm[koeff]

Out[10]//MatrixForm=

$$\begin{pmatrix} a + \frac{1}{2} (-a - d + \sqrt{a^2 + 4 b c - 2 a d + d^2}) & b \\ c & d + \frac{1}{2} (-a - d + \sqrt{a^2 + 4 b c - 2 a d + d^2}) \end{pmatrix}$$

In[11]:= NullSpace[koeff]

Out[11]= $\left\{ \left\{ -\frac{-a + d + \sqrt{a^2 + 4 b c - 2 a d + d^2}}{2 c}, 1 \right\} \right\}$

Zur Lösung des homogenen Gleichungssystems wird hier die Funktion **NullSpace** benutzt, welche eine Liste mit Basisvektoren zum Kern der Matrix **koef** liefert.

3.11.3 Weitere Operationen mit Listen

Mathematica erlaubt noch zahlreiche weitere Operationen mit Listen, die mathematisch weniger relevant, aber für anwendungsbezogene Programme möglicherweise sehr nützlich sind, z.B.:

- Zugriff auf bestimmte Elemente einer Liste, z.B. auf das letzte Element, auf die ersten n Elemente
- Suche nach der Position eines Ausdrucks in einer Liste
- Einfügen, Anhängen oder Löschen von Listenelementen
- Elemente umordnen oder permutieren

3.12 Verwendung externer Dateien

3.12.1 Ergebnisse und Definitionen in externe Dateien schreiben

Sie können alle Ergebnisse (Inhalte von Ausgabezellen) sowie die Definitionen (von Variablen oder Funktionen) in Textform abzuspeichern. Per Voreinstellung werden dabei Mathematica-Ausdrücke geschrieben, die anschließend wieder als Mathematica-Eingaben zu verwenden sind. Man kann aber auch alternative Ausgabeformate anfordern.

3.12.1.1 Ausgabe im Mathematica-Format

3.12.1.1.1 Umleitung von Ergebnissen

In folgendem Beispiel wird eine mit der Funktion **Table** erzeugte Liste mit Hilfe des Umleitungsoperators ">>" in eine externe Datei geschrieben:

```
In[1]:= Table[x y, {x, 1, 4}, {y, 1, 4}]>>"c:\\user\\tabelle 1"
```

Die Datei **tabelle 1** hat anschließend den folgenden Inhalt:

{1, 2, 3, 4}, {2, 4, 6, 8}, {3, 6, 9, 12}, {4, 8, 12, 16}}
--

Hinsichtlich der Dateispezifikation ist bei der **Windows**-Version von Mathematica zu beachten:

- Wenn die Dateispezifikation Leerzeichen, eine Laufwerksangabe oder eine Pfadangabe enthält, muß sie mit doppelten Anführungszeichen (") begrenzt werden. Bei einfachen Dateinamen ohne Leerzeichen ist dies nicht nötig, z.B.:

```
In[1]:= Table[x y, {x, 1, 4}, {y, 1, 4}]>>tabelle1
```

Dann gelten die üblichen Voreinstellungen für Laufwerk und Pfad.

- Die umgekehrten Schrägstriche in den Windows-Pfadangaben müssen verdoppelt werden.

Am besten sorgt man für ein geeignetes Arbeitsverzeichnis, damit Laufwerks- und Pfadangaben überflüssig sind. Dies kann mit dem folgenden Mathematica-Befehl geschehen:

```
In[2]:= SetDirectory["c:\\user"]
```

Mit dem oben erklärten Umleitungsoperator ">>" wird *eine Zeile* in die Ausgabedatei geschrieben, wobei ein eventuell vorhandener alter Inhalt gelöscht wird. Wenn Sie die Ausgabedatei um eine Zeile *erweitern* möchten, ist der Umleitungsoperator ">>>" zu verwenden, z.B.:

```
In[3]:= Table[x y, {x, 1, 4}, {y, 1, 4}]>>>tabelle1
```

Natürlich ist die Ausgabeumleitung nicht auf die Funktion **Table** beschränkt, sondern klappt mit beliebigen Mathematica-Ausdrücken. Die allgemeine Syntax lautet:

<i>ausdruck</i> >> <i>dateispezifikation</i>	Das Ergebnis wird in die erste Zeile der angegebenen Datei geschrieben, wobei vorhandene Inhalte ggf. ersetzt werden.
<i>ausdruck</i> >>> <i>dateispezifikation</i>	Das Ergebnis wird als neue Zeile in die angegebene Datei geschrieben, wobei vorhandene Zeilen erhalten bleiben.

3.12.1.1.2 Sichern von Definitionen

Mit den eben beschriebenen Umleitungsoperatoren ">>" und ">>>" kann man Mathematica-Ergebnisse bequem abspeichern. Variablen- oder Funktionsdefinitionen sollten hingegen mit der Mathematica-Funktion **Save** gesichert werden, weil man die Definitionen dann später besonders leicht reaktivieren kann. In folgendem Beispiel

```
In[1]:= t = Table[x y, {x, 1, 4}, {y, 1, 4}]
```

```
Out[1]= {{1, 2, 3, 4}, {2, 4, 6, 8}, {3, 6, 9, 12}, {4, 8, 12, 16}}
```

```
In[2]:= Save["t_def", t]
```

wird die Definition der Variablen **t** in die externe Datei **t_def** geschrieben, die anschließend folgenden Inhalt hat:

t = {{1, 2, 3, 4}, {2, 4, 6, 8}, {3, 6, 9, 12}, {4, 8, 12, 16}}

Bezüglich der Dateispezifikation gelten im wesentlichen die Ausführungen aus dem vorherigen Abschnitt, wobei der Dateiname in der **Save**-Funktion allerdings stets in doppelte Hochkommata eingeschlossen werden muß. Weil im Beispiel auf Laufwerks- und Pfadangaben verzichtet wird, landet die Datei **t_def** also im momentan eingestellten Arbeitsverzeichnis.

Mit Hilfe des Umleitungsoperators "<<" kann eine so gespeicherte Definition später ganz leicht reaktiviert werden, z.B.:

```
In[3]:= << t_def
```

```
Out[3]= {{1, 2, 3, 4}, {2, 4, 6, 8}, {3, 6, 9, 12}, {4, 8, 12, 16}}
```

Die allgemeine Syntax der Funktion **Save** lautet:

Save["dateispezifikation", f, g, ...]	Die Definitionen der angegebenen Variablen und Funktionen werden in eine externe Datei geschrieben.
--	---

3.12.1.2 Ausgabe im Textformat

Bei allen bisher vorgestellten Ausgabemethoden werden Mathematica-Ausdrücke gesichert, die anschließend wieder als Mathematica-Eingaben verwendet werden können. Gelegentlich möchte man aber z.B. eine zweidimensionale Liste in einem einfachen Textformat ausgeben (ohne Mathematica-Syntaxelemente), z.B. zur Weiterverarbeitung mit einem anderen Programm. Dazu kann die Funktion **OutputForm** verwendet werden, z.B.:

```
In[1]:= t=Table[x y,{x,1,4},{y,1,4}]
```

```
Out[1]:= {{1, 2, 3, 4}, {2, 4, 6, 8}, {3, 6, 9, 12}, {4, 8, 12, 16}}
```

```
In[2]:= OutputForm[MatrixForm[t]]>>tabelle
```

Anschließend enthält die Datei **tabelle**:

1	2	3	4
2	4	6	8
3	6	9	12
4	8	12	16

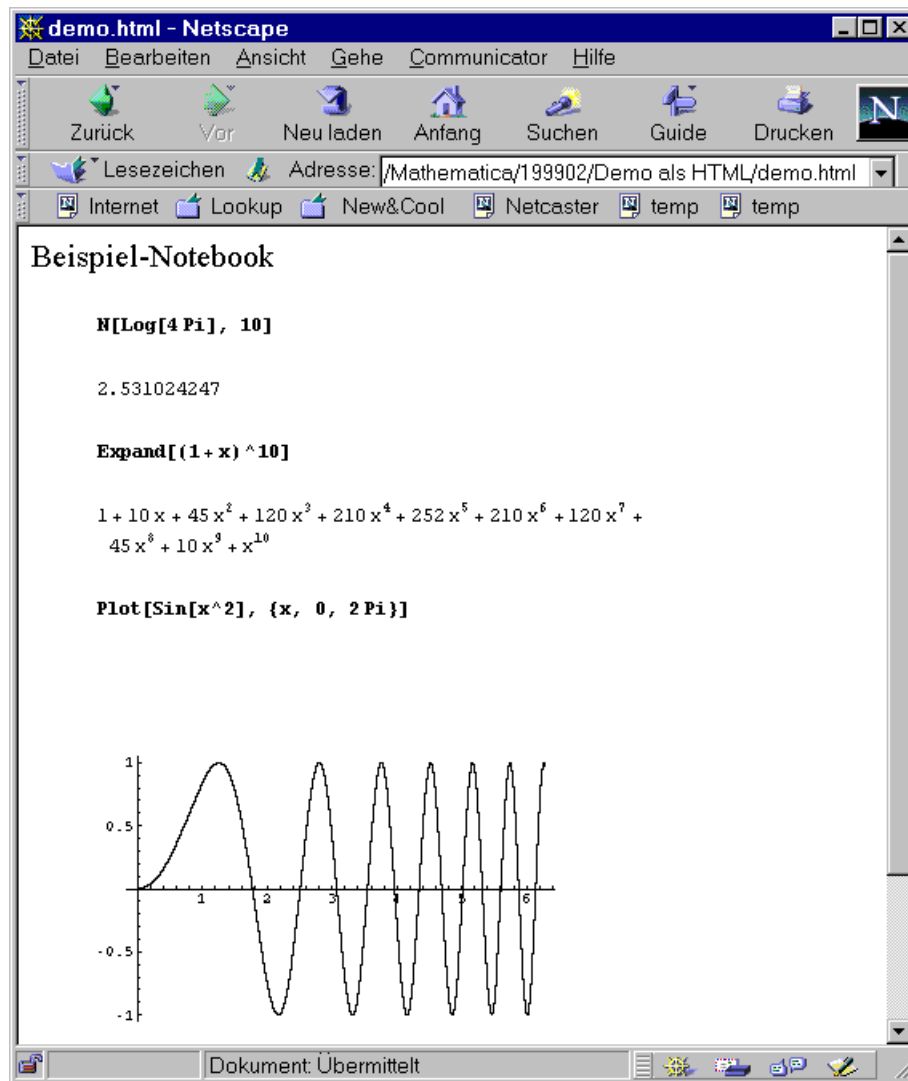
Im Hinblick auf spätere Abschnitte löschen wir die Definition der Variablen **t** wieder:

```
In[3]= Clear[t]
```

3.12.1.3 Weitere Export- bzw. Publikationsoptionen

Mathematica beherrscht noch zahlreiche weitere Exportoptionen, z.B.:

- Der Export von Grafiken wurde schon in Abschnitt 3.8.5 behandelt.
- Mit **TeXSave** kann ein Notizbuch im T_EX-Format gesichert werden.
- Mit **HTMLSave** kann ein Notizbuch im HTML-Format gesichert werden.
Die HTML-Version zu unserem Demo-Notebook sieht folgendermaßen aus:



Auf der HTML-Seite wird allerdings der Einsatz von GIF-Dateien etwas übertrieben.

3.12.2 Aus externen Dateien lesen

3.12.2.1 Mathematica-Ausdrücke einlesen

Die oben gesicherte Definition der Variablen **t** kann mit Hilfe des Umleitungsoperators "<<" bequem aus der Datei reaktiviert werden:

```
In[1]:= << t_def
```

```
Out[1]= {{1, 2, 3, 4}, {2, 4, 6, 8}, {3, 6, 9, 12}, {4, 8, 12, 16}}
```

Der Umleitungsoperator "<<", den wir auch beim Einlesen von Mathematica-Paketen verwenden (siehe Abschnitt 3.4) bewirkt, daß die Mathematica-Ausdrücke in der Eingabedatei *ausgewertet* werden.

Wenn Sie Dateiinhalte nur *anzeigen* lassen wollen, können Sie die folgende Anweisung verwenden:

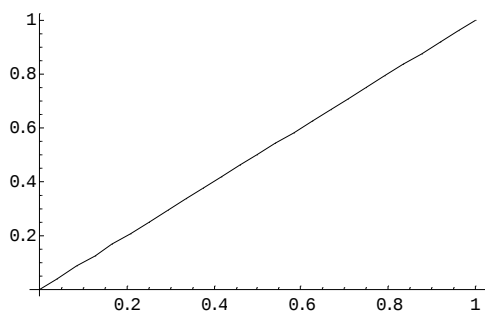
```
In[2]:= !!t_def
```

```
t = {{1, 2, 3, 4}, {2, 4, 6, 8}, {3, 6, 9, 12}, {4, 8, 12, 16}}
```

Um eine per Ausgabeumleitung im Mathematica-Format gesicherte Grafik erneut anzuzeigen, kann die **Show**-Funktion verwendet werden, z.B.:

```
In[3]:= Plot[x,{x,0,1}] >> gerade
```

```
In[4]:= Show[<<gerade]
```



```
Out[4]= -Graphics-
```

3.12.2.2 Aus Textdateien lesen

Die Funktion **ReadList** ermöglicht es, aus einer Textdatei Daten (z.B. Zahlen) in eine Mathematica-Liste einzulesen. Die Beispieldatei **test.dat** enthält zwei Zeilen mit tabulator-getrennten Zahlen:

```
In[1]:= !!test.dat
```

```
16  9  4  1
4   3  2  1
```

```
In[2]:= ReadList["test.dat", Number]
```

```
Out[2]= {16, 9, 4, 1, 4, 3, 2, 1}
```

Mit **Number** wird der Datentyp angegeben. Alternative Datentypen sind z.B. **Character** oder **String**. Im vorliegenden Beispiel wird man es normalerweise bevorzugen, die Eingabematrix in eine Liste von Listen zu überführen, die in Mathematica zur Darstellung von Matrizen verwendet wird. Dazu müssen wir die **ReadList**-Funktion um eine Option erweitern:

```
In[3]:= ma=ReadList["test.dat", Number, RecordLists->True]
```

```
Out[3]= {{16, 9, 4, 1}, {4, 3, 2, 1}}
```

4 Zusätzliche Hinweise zur Notizbuch-Benutzeroberfläche

4.1 Hilfen beim Erstellen und Ausführen von Mathematica-Ausdrücken

4.1.1 Funktionsbezeichnungen automatisch vervollständigen lassen

Die Bezeichnungen von Funktionen (mathematica-intern oder benutzerdefiniert) müssen nicht unbedingt komplett eingetippt werden, sondern können von Mathematica vervollständigt werden. Dazu verwendet man nach dem Eintippen des Wortanfangs den Menübefehl **Input > Complete Selection** oder besser die folgende Tastenkombination:

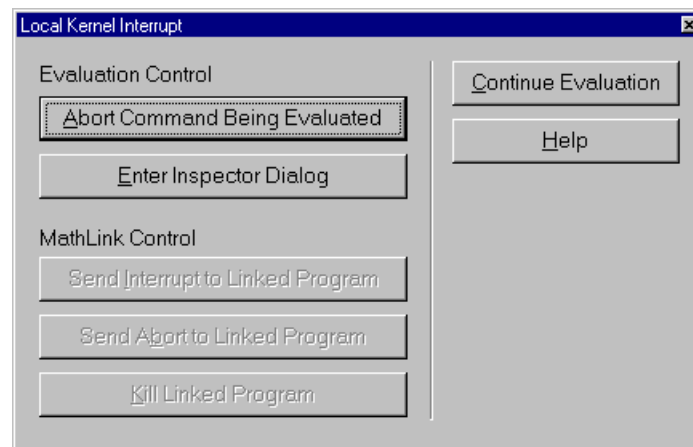
Windows	Macintosh	X
<Strg><K>	<Befehl><K>	<Mod1><K>

Ist keine eindeutige Expansion möglich, werden alle Optionen in einem Fenster angeboten, das z.B. bei der Windows-Version nach Eingabe des Wortanfangs **Para** so aussieht:



4.1.2 Mathematica unterbrechen

Sie können eine laufende Berechnung mit dem Menübefehl **Kernel > Interrupt Evaluation** bzw. mit der Tastenkombination **<Alt>+<, >** unterbrechen und dann in einer Dialogbox über das weitere Vorgehen entscheiden:



Es kann einige Zeit (z.B. sogar mehrere Minuten) dauern, bis Mathematica auf Ihre Unterbrechung reagiert.

4.2 Zellenattribute und -stile

4.2.1 Zellenattribute

Über **Cell > Cell Properties** können verschiedene Attribute von **markierten** Zellen ein- und ausgeschaltet werden, u.a.:

4.2.1.1 Editable

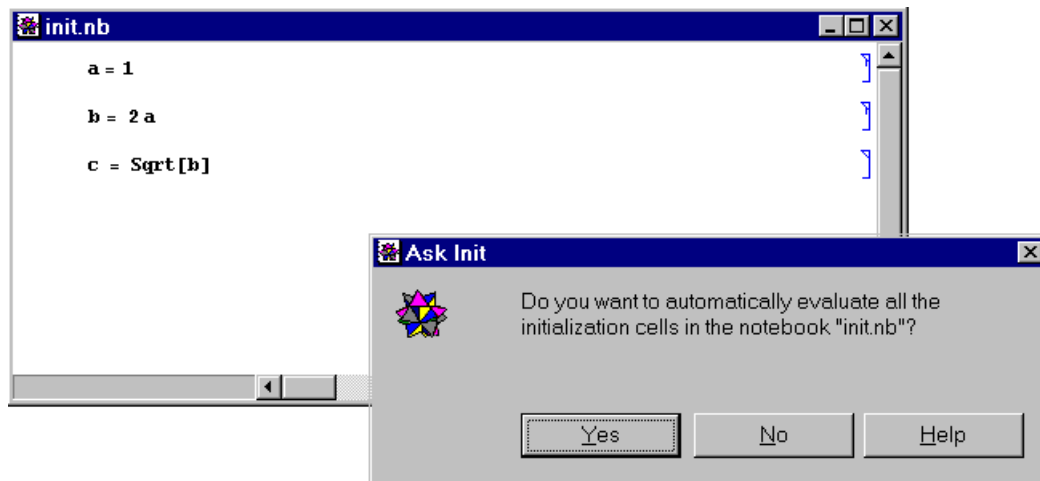
Man kann z.B. eine Input-Zelle vor unbeabsichtigten Veränderungen schützen, indem man ihr das Attribut **Editable** entzieht. Dieser Zustand wird durch ein "X" am oberen Rand der Zellenklammer gekennzeichnet:



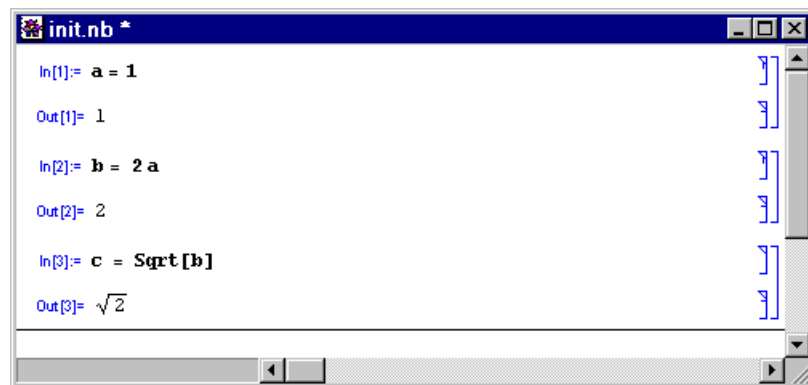
Das Markieren einer Zelle oder Zellengruppe erledigt man übrigens per Mausklick auf die zugehörige Klammer.

4.2.1.2 Initialization

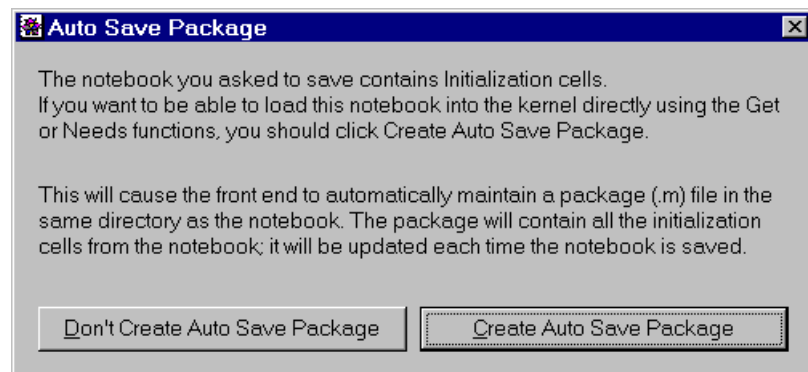
Die Initialisierungszellen eines Notebooks werden von Mathematica automatisch ausgeführt (nach Anfrage), sobald für irgend eine Zelle des Notebooks die Auswertung angefordert oder der Menübefehl **Kernel > Evaluation > Evaluate Initialization** ausgeführt wird. Damit kann man sicher stellen, daß gewisse Definitionen bei der Arbeit mit einem Notebook grundsätzlich zur Verfügung stehen. Das folgende, frisch geladene Notebook enthält zwei einleitende Initialisierungszellen, durch ein "I" in der Zellenklammer gekennzeichnet:



Läßt man die *dritte* Zelle ausführen und bestätigt die Auswertung der Initialisierungszellen, wird der Wert von *c* unter Berücksichtigung der Definitionen von *a* und *b* berechnet:



Beim ersten Speichern eines Notebooks mit Initialisierungszellen bietet Mathematica das Erstellen einer zugehörigen Package-Datei mit dem Inhalt aller Initialisierungszellen an:



Für die oben beschriebene Arbeitsweise mit Initialisierungszellen ist diese Datei *nicht* erforderlich. Was eine Package-Datei enthält, und wozu sie dienen kann, verrät das zum obigen Beispiel erzeugte Exemplar:

```
(*****
This file was generated automatically by the Mathematica front end.
It contains Initialization cells from a Notebook file, which typically
will have the same name as this file except ending in ".nb" instead of
".m".
```

```
This file is intended to be loaded into the Mathematica kernel using
the package loading commands Get or Needs. Doing so is equivalent to
using the Evaluate Initialization Cells menu command in the front end.
```

DO NOT EDIT THIS FILE. This entire file is regenerated automatically each time the parent Notebook file is saved in the Mathematica front end. Any changes you make to this file will be overwritten.

*****)

a=1

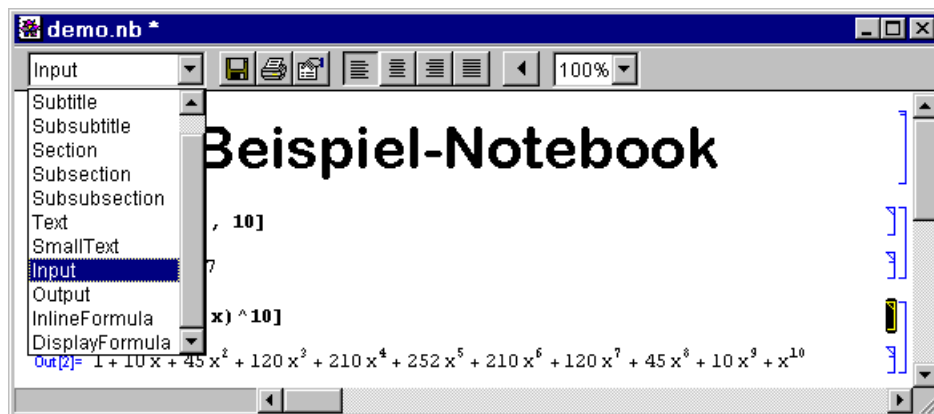
b= 2 a

4.2.2 Zellenstile

Jede Zelle hat einen bestimmten Stil, gekennzeichnet durch eine Kombination von Zellenattributen und Textformatierungen. Sie können für die markierten Zellen nach dem Menübefehl **Format > Style** aus dem folgenden Menü einen Zellenstil auswählen:

Title	At+1
Subtitle	At+2
Subsubtitle	At+3
Section	At+4
Subsection	At+5
Subsubsection	At+6
Text	At+7
SmallText	At+8
✓ Input	At+9
Output	
InlineFormula	
DisplayFormula	
Other...	At+0

Wer mit **Format > Show ToolBar** die Symbolleiste zum Notebook-Fenster eingeschaltet hat, kann auch über die dortige versteckte Liste einen Zellenstil auswählen, z.B.:



Wie in obigem Beispiel mit einer einleitenden Textzelle zu sehen ist, wirken sich manche Textstile auch auf die Gestalt der Zellenklammer aus, wobei wir uns die Einzelheiten aber sparen wollen.

4.2.3 Style Sheets

Das Aussehen der einzelnen Zellenstile hängt vom aktiven **Style Sheet** ab. Mathematica bringt etliche allgemein verfügbare Style Sheets mit, die Sie einem Notebook über **Format > Style Sheet** zuweisen können. Das am Anfang des Manuskriptes gezeigte Beispiel-Notebook sieht nach Zuweisung des Style Sheets **Report** (und Abschaltung der Zellennumerierung mit **Kernel > Show In/Out Names**) folgendermaßen aus:

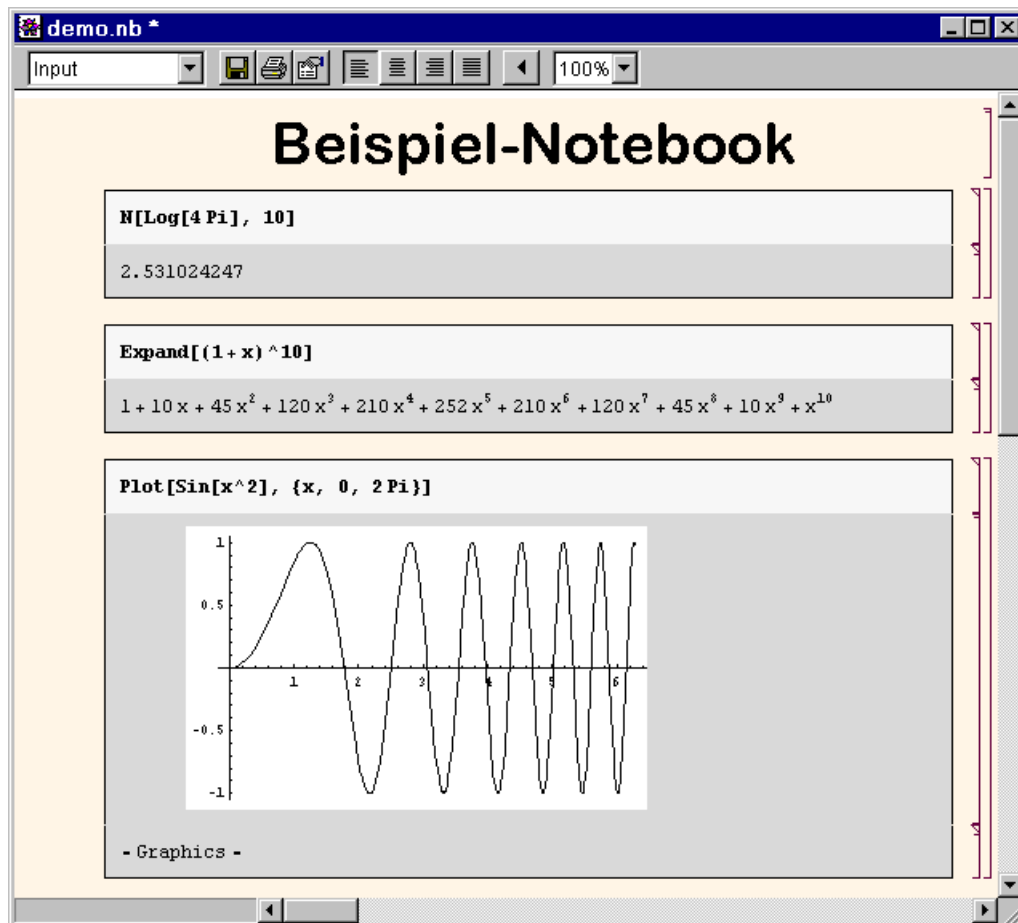


Abbildung 5 Beispiel-Notebook mit Style Sheet *Report* und abgeschalteter Zellennumerierung

Wer ein eigenes Style Sheet gestalten möchte, kann dies nach **Format > Edit Style Sheet** tun, wobei folgende Möglichkeiten angeboten werden:

- Ein allgemein verfügbares Style Sheet ändern.
- Ein privates Style Sheet für das aktuelle Notebook erstellen.

4.3 Verfahren für Gruppen von Zellen

Zunächst müssen Sie **Cell > Cell Grouping > Manual Grouping** die manuelle Gruppierung aktivieren.

4.3.1 Zellen zu einer Gruppe zusammenfassen

Markieren Sie alle Zellen bzw. Zellengruppen, und wählen Sie den Menübefehl **Cell > Cell Grouping > Group Cells**. Am rechten Fensterrand erscheint eine zusätzliche Gruppenklammer, die alle einbezogenen Zellen bzw. Zellengruppen umfaßt.

Leider mußte ich eine Ungereimtheit beobachten, wenn der erste Gruppierungsaspirant selbst bereits eine Gruppe war: Die vorbestehende Gruppierung wurde kurzerhand aufgehoben, die zugehörigen Zellen wurden der neuen, umfassenderen Gruppe als „Einzelgänger“ eingegliedert.

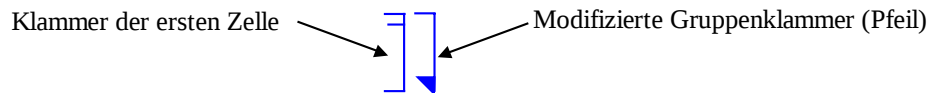
4.3.2 Eine Gruppierung aufheben

Markieren Sie die Gruppenklammer, und wählen Sie den Menübefehl **Cell > Cell Grouping > Ungroup Cells**.

4.3.3 Zellengruppen schließen und öffnen

Eine Gruppe von Zellen kann per Doppelklick auf Gruppenklammer aus- bzw. wieder eingeblendet werden. Alternativ können Sie auch den Menübefehle **Cell > Cell Grouping > Open/Close Group** benutzen.

Eine eingepackte Gruppe wird im Notizbuch durch ihre erste Zelle vertreten, die sinnvollerweise eine Überschrift enthalten sollte, und in der Klammernzone folgendermaßen gekennzeichnet:



Wenn die erste Zelle jeder Gruppe eine Überschrift enthält, kann durch Schließen aller Gruppen eine Gliederungsansicht des Notizbuchs hergestellt werden.

Mehr Übersicht kann man auch mit den folgenden Befehlen gewinnen, die auf alle markierten Gruppen wirken:

Cell > Cell Grouping > Open All Subgroups
Cell > Cell Grouping > Close All Subgroups

4.3.4 Zellen aufteilen

Mit dem Menübefehl **Cell > Devide Cell** kann eine markierte Zelle u.a. folgendermaßen aufgeteilt werden:

i) Einfügemarke (|) befindet sich irgendwo innerhalb der Zelle

Zellinhalt vor der Einfügemarke	=	erste neue Zelle
Zellinhalt nach der Einfügemarke	=	zweite neue Zelle

ii) Eine Zeichenfolge ist markiert

Zellinhalt vor der Markierung	=	erste neue Zelle
Markierung	=	zweite neue Zelle
Zellinhalt nach der Markierung	=	dritte neue Zelle

4.3.5 Zellen zusammenlegen

Mehrere markierte Zellen können mit **Cell > Merge Cells** zusammengelegt werden, wobei die Attribute und Stile (siehe oben) der neuen Zelle von der ersten markierten Zelle übernommen werden.

5 Literatur und sonstige Informationsquellen

Im Manuskript werden folgende Publikationen zitiert:

Endl, K. & Luh, W. (1983). *Anaysis II* (6. Aufl.). Wiesbaden: AULA-Verlag.

Hettich, R. (1989). *Numerische Lineare Algebra*. Trier: Vorlesungsmitschrift

Heuser, H. (1986). *Lehrbuch der Analysis, Teil 1*. Stuttgart: Teubner.

Martin, E. (1996). *Mathematica 3.0. Standard Add-On Packages*. . Champaign, IL: Wolfram Media.

Wolfram, S. (1996). *The Mathematica Book* (3rd ed.). Champaign, IL: Wolfram Media.

Die Originalhandbücher zu Mathematica sind in elektronischer Form im Help Browser verfügbar (siehe Abschnitt 3.1). Wer gedruckte Informationen bevorzugt, kann die Handbücher aber auch in der Benutzerberatung des Rechenzentrums kurzzeitig ausleihen.

Als Einstieg für die diversen Internet-Ressourcen zu Mathematica eignet sich die folgende WWW-Seite:

<http://www.wolfram.com>

6 Anhang

6.1 X-Mathematica auf der RZSUN00 benutzen

In diesem Abschnitt wird beschrieben, wie Sie mit der auf dem UNIX-Rechner RZSUN00 installierten X-Version von Mathematica 3.0 arbeiten können.

Bei der RZSUN00 handelt es sich um einen SUN-SPARC-Server 1000 unter dem Betriebssystem SOLARIS (ein UNIX-SVR4-Dialekt) mit mehreren Prozessoren und reichlich Hauptspeicher.

6.1.1 Voraussetzungen für das Arbeiten mit Mathematica auf der RZSUN00

6.1.1.1 EDV-Kenntnisse

Beim Arbeiten mit der Mathematica-Version auf der RZSUN00 sind zwei Rechner im Spiel:

- **Die RZSUN00 selbst**

Hier läuft das Programm Mathematica unter dem Betriebssystem SOLARIS (ein UNIX-Dialekt).

- **Ihr Arbeitsplatzrechner**

Dieser Rechner läuft oft ebenfalls unter UNIX, eventuell aber auch unter einem anderen Betriebssystem, z.B. unter DOS/Windows, OS/2 oder Mac-OS. Bei einem PC unter Windows NT ist jedoch die Verwendung der UNIX-Installation von Mathematica wenig sinnvoll, weil auch die MS-Windows-Version benutzt werden kann (s.o.).

Im folgenden wird vorausgesetzt, daß Sie elementare Fertigkeiten im Umgang mit den beiden Rechnern besitzen. Bei Bedarf können Sie in der Skriptenstelle des URT (Raum E-046) eine preiswerte Einführung in das Betriebssystem UNIX erwerben.

6.1.1.2 Benutzerverwaltung

Um mit der RZSUN00 arbeiten zu können, müssen Sie sich nötigenfalls im Raum E-046 als Benutzer für dieses System registrieren lassen.

6.1.1.3 Anforderungen an Ihren Arbeitsplatzrechner

Damit Sie die Notizbuch-Oberfläche von Mathematica verwenden können, muß Ihr Arbeitsplatzrechner die X-Server-Funktionalität besitzen. Dies ist bei UNIX-Systemen in der Regel gegeben. Rechner unter einem anderen Betriebssystem benötigen meistens eine X-Server-Zusatzsoftware. Bei einem PC unter MS-Windows (3.x, 95, NT) eignet sich dafür z.B. das Programm **XVision**, das an der Universität Trier als Campuslizenz verfügbar ist.

Frei zugängliche Arbeitsplatzrechner unter UNIX (BSD/OS) finden Sie im Raum V-5. Wie für die RZSUN00 gilt auch für diese UNIX-Arbeitsplatzrechner, daß nur eingetragene Benutzer damit arbeiten können.

Wenn Sie auf der RZSUN00 mit der **textorientierten** Oberfläche von Mathematica arbeiten wollen, benötigt Ihr Arbeitsplatzrechner nur einen Terminal-Zugang zur RZSUN00, z.B. über den Internetdienst **Telnet**. Dieser Zugang steht auf praktisch jedem Rechner im Campusnetz der Universität zur Verfügung. Die Anforderungen an den Arbeitsplatzrechner sind minimal, so daß z.B. auch ein PC-AT mit 640 KB RAM ohne weiteres verwendet werden könnte. Auch externe Rechner, die nicht in einem Gebäude der Universität Trier installiert sind, können via Internet oder über Modem bzw. ISDN eine Telnet-Verbindung zur RZSUN00 herstellen, so daß Sie z.B. auch vom heimischen PC aus mit der Mathematica-Version auf der RZSUN00 arbeiten können.

6.1.2 Eine Sitzung mit X-Mathematica auf der RZSUN00

Weil aktuelle PCs an der Universität Trier das Betriebssystem Windows NT verwenden und damit die MS-Windows-Version von Mathematica nutzen können, wird der Zugriff auf die X-Version von Mathematica nur für UNIX-Arbeitsplatzrechner beschrieben.

6.1.2.1 Vorbereitungen auf dem UNIX-Arbeitsplatzrechner

Ist ein modern konfigurierter UNIX-Rechner frei für den nächsten Benutzer, so präsentiert er ein Login-Fenster zur Gesprächseröffnung. Schicken Sie Ihren Benutzernamen und Ihr Paßwort mit **<Enter>** ab.

In der Version 3.0 unterstützt Mathematica bei der Eingabe einige Sonderzeichen (z.B. π), die bisher umschrieben werden mußten (z.B. Pi). Dazu benötigt das Programm spezielle Zeichensätze, die beim Einsatz der X-Version dynamisch von der RZSUN00 bezogen werden können. Die X-Server-Software auf Ihrem Arbeitsplatzrechner muß allerdings darüber informiert werden, daß die RZSUN00 über den TCP-Port 7100 als Fontserver ansprechbar ist.

Ferner muß der X-Server auf Ihrem Arbeitsplatzrechner die RZSUN00 als X-Host akzeptieren. In der Regel müssen Sie dazu ein **xhost-Kommando** abschicken, z.B. (Benutzereingaben fett):

```
bash$ xhost rzsun00
```

Eröffnen Sie schließlich eine Telnet-Verbindung zur RZSUN00:

```
bash$ telnet rzsun00
```

6.1.2.2 Mathematica auf der RZSUN00 starten und beenden

Melden Sie sich mit Ihren Benutzerkennndaten *für die RZSUN00* an, z.B.:

```
login: rzsp007  
Password:
```

Sie müssen auf der RZSUN00 folgende **Umgebungsbedingungen** sicherstellen:

- In der Environmentvariablen DISPLAY muß der Internetname Ihres Arbeitsplatzrechners angegeben sein, sonst erhalten Sie die Fehlermeldung "Can't open display". Vereinbaren Sie daher wie in folgenden Beispielen (mit dem Internetnamen „dummy13“) die DISPLAY-Variable:

– **c-shell:**

```
rzsun00% setenv DISPLAY dummy13:0.0
```

– **bash:**

```
RZSUN00:~ >export DISPLAY=dummy13:0.0
```

- Die Umgebungsvariablen OPENWINHOME muß auf /usr/openwin/bin gesetzt werden, was in der **bash** folgendermaßen geschehen kann:

```
RZSUN00:~ >export OPENWINHOME=/usr/openwin/bin
```

Ob die Bedingungen erfüllt sind, können Sie mit dem Kommando **env** überprüfen, das alle Umgebungsvariablen anzeigt.

Wer Mathematica auf der RZSUN00 häufiger benutzt, wird die Kommandos zur Definition von Umgebungsvariablen natürlich in eine geeignete Konfigurationsdatei eintragen, bei Bourne-Shell-Varianten (z.B. bash) also in die Datei **.profile**.

Die Notizbuch-Oberfläche von Mathematica wird folgendermaßen gestartet:

```
rzsun00% mathematica
```

und mit **File > Exit** beendet.

Die textorientierte Oberfläche wird mit:

```
rzsun00% math
```

gestartet und mit **<Ctrl>+<D>** beendet.

6.1.2.3 Verhalten nach dem Beenden von Mathematica

Nachdem Sie Mathematica mit:

File > Exit

verlassen haben, ist Ihre Telnet-Verbindung zur RZSUN00 immer noch aktiv. Beenden Sie diese ebenfalls, indem Sie im zugehörigen Dialogfenster die Tastenkombination

<Strg>+<D>

drücken.

Jetzt sind Sie nur noch mit dem lokalen UNIX-Rechner verbunden. Wenn Sie Ihre Arbeit an diesem Rechner ebenfalls beenden wollen, beachten Sie bitte folgendes: Ihr Kontakt mit dem UNIX-System wird **nicht** beendet, wenn Sie z.B. das aktuelle Dialogfenster mit einem weiteren **<Strg>+<D>** schließen. Ein Unbefugter könnte auf dem Rechner unter Ihrem Namen und damit eventuell zu Ihrem Nachteil mühelos ein neues Dialogfenster eröffnen. Beenden Sie daher unbedingt Ihr Gespräch mit dem lokalen UNIX-Rechner. Auf den Rechnern in V-5 müssen Sie auf das Logout-Symbol am unteren Bildschirmrand klicken.

6.2 Wichtige Tastaturbelegungen

Auf dem Macintosh erhalten Sie die beim Arbeiten mit Mathematica unverzichtbaren Klammern folgendermaßen:

[	Alt (Wahl) + <5>
]	Alt (Wahl) + <6>
	Alt (Wahl) + <7>
{	Alt (Wahl) + <8>
}	Alt (Wahl) + <9>

Tastatur-Äquivalenzen zwischen den Mathematica-Versionen für Windows, Macintosh und X-Window:

Windows	Macintosh	Solaris
<Strg>	<Befehl>	<Mod1> (bei meinem X-Server: Alt)

7 Stichwortverzeichnis

\$FormatType	38
\$TextStyle	37
%	14
<<	55, 56
>>	54
>>>	54

3

3D-Aussichtspunkt	36
3D-Oberflächendiagramme	36

A

Ableitung	
partielle	17
totale	17
unbekannter Funktionen	18
Abs	13
AbsolutePointSize	40
Adobe Illustrator File	32
AI 32	
Algebraische Gleichungen	22
Anfangswertproblem	25
Apart	16
Arbeitsverzeichnis einstellen	54
Arg	13
Arithmetische Operationen	10
Array	51
Arrays	15
Ausdruck	
Auswertungsserienfolge	11
Ausgeben von Werten	46
Äußeres Produkt	49
Auswertung anfordern	7
Auswertungsserienfolge	11

B

Backslash	54
Bedingte Anweisungen	45
Benutzeroberfläche	6
Bestimmtes Integral	19
BMP	32

C

Clear	44
Coefficient	45
ColumnForm	48
Conjugate	13
ConstrainedMin	43
ContourPlot	34
Contours	35

D

D (partielle Ableitung)	17
Dateien	
lesen aus	56
schreiben in	53
Datenlisten	40
Definitionen von Symbolen	
anzeigen	44

löschen	44
DensityPlot	35
Determinante einer Matrix	50
DiagonalMatrix	52
Dichtediagramme	35
Differentialgleichungen	25
Differenzieren	17
DISPLAY	65
Doppelsummen	20
Do-Schleifen	46
DSolve	25
Dt (totale Ableitung)	17

E

Editable	58
Eigenvektoren	51
Eigenwerte einer Matrix	50
Einheitsmatrix	52
Eliminierung	24
EMF	32
Encapsulated PostScript File	32
Enhanced Metafile	32
EPS	32
Ersetzungsregeln	22
Eulersche Zahl	28
Expand	16
ExpandAll	16
Export einer Grafik	32
via Zwischenablage	32
Exportieren von Mathematica-Objekten	55

F

Factor	16
Fallunterscheidung	45
Felder	15
FindMinimum	43
FindRoot	42
Fontserver	65
Frame	31
Funktionen	
definieren	44

G

Genauigkeit	11
Geometrische Reihe	19
Gleichungen	22
algebraische	22
polynomiale	22
transzendente	23
Gleichungssystem	
homogenes	52
inhomogenes	52
Gleichungssysteme	23
Grafik	
Export in eine Datei	32
Größe ändern	31
übertragen via Zwischenablage	32
verschieben	31
Grafikanweisungen	40
GraphicsArray	34
Grenzwerte	26

Gruppierung	
aufheben.....	61
erstellen.....	61

H

Help Browser	12, 15
HTMLSave.....	55

I

IdentityMatrix	52
If 45	
Im 13	
Indizieren	48
Initialization	58
Inneres Produkt	49
InputForm	29
Input-Zelle	7
Input-Zellen	7
Integral	
bestimmtes	19
unbestimmtes	18
Integrate	18
Inverse Matrix	50

K

Kernel.....	6
Klammern	
eckige.....	12
geschweifte	15
runde.....	11
Komplexe Zahlen.....	13
Konstanten	13
Konturdiagramme	34
Kronecker Produkt	49
Kuh-Schaf-Beispiel	43

L

Limit.....	26
Lineare Gleichungssysteme	23
Lineare Gleichungssysteme	52
Lineare Programmierung.....	44
LinearSolve	52
Listen	15, 47, 57
Elemente herausziehen.....	48
Unterlisten	48
ListPlot.....	40
logische Operatoren	22

M

Macintosh	
Tastaturbelegung.....	66
Mathematische Notation	26
Standardform	28
Traditionelle Form	28
Mathlink.....	6
MatrixForm.....	48
Matrixprodukt	49
Matrizen	15
Mesh	35
Microsoft Windows Bitmap	32
Module	45

N

N 11	
NIntegrate.....	41
Notizbücher	7
NProduct	41
NSolve.....	42
NSum	41
NT	6
NullSpace	53
Numeric.....	11
Numerische Berechnungen.....	10
Numerische Mathematik	41

O

Operatoren	
logische	22
relationale.....	21
Optimierung	
mit Nebenbedingungen	43
ohne Nebenbedingungen.....	43
Optionen für Grafiken	31
Outer	49
OutputForm.....	29, 55
Output-Zelle	7

P

Package-Datei.....	59
Pakete	6, 15
Paletten.....	27
ParametricPlot	39
ParametricPlot3D	39
Parametrische Diagramme.....	39
Plot.....	30
PlotJoined.....	40
PlotLabel	31
PlotPoints	33, 35
PlotRange.....	33
PlotStyle	40
Polynom-Gleichungen.....	22, 23
Postfix-Schreibweise	11
Postscript.....	30
Potenzreihen.....	26
PowerExpand	17
Print-Funktion	46
Print-Zelle	46
Produkte	20

R

Re 13	
ReadList	57
Reduce.....	24
Reine Funktion.....	25
relationale Operatoren	21
Relationen	21
Remove	15, 44
Root.....	23
RZSUN00.....	64

S

Save.....	54
Schleifen.....	46
Schriftarten	

in Grafiken	37
Series	26
SetDirectory	54
Show	32, 56
Simplify	16
Solve	22
StandardForm	28
StandardForm	29, 38
StyleForm	38
Summen	19
Symbolleiste	9

T

Tabellen	47
mehrdimensionale	47
Table	47
TableForm	48
Tastaturbelegungen	66
Taylor-Reihen	26
Telnet	64
TeXSave	55
TraditionalForm	29, 38
Traditionelle Form	28
Transponierte Matrix	50
Transzendente Gleichungen	23

U

Umleitungsoperator	54
Unbestimmtes Integral	18
UNIX	6, 64
Unterbrechen	58
Unterliste	48
Unterprogramme	45

V

Variablen	14
-----------------	----

globale	45
indizierte	51
lokale	45
Variablennamen	14
Vergleich	21
Vergleichsoperatoren	21
ViewPoint	36
Vollständige Lösungen	25

W

Werte ausgeben	46
Which	46
Windows NT	6

X

xhost-Kommando	65
X-Server	64

Z

Zell-Einfügemarke	7
Zellen	7
aufteilen	62
gesperrte	8
Input	7
Output	7
zusammenlegen	62
Zellen gruppieren	61
Zellenattribute	58
Editable	58
Initialization	58
Zellengruppen schließen und öffnen	62
Zellennumerierung	
abschalten	8
Zellenstile	60
Zweidimensionale Notation	27