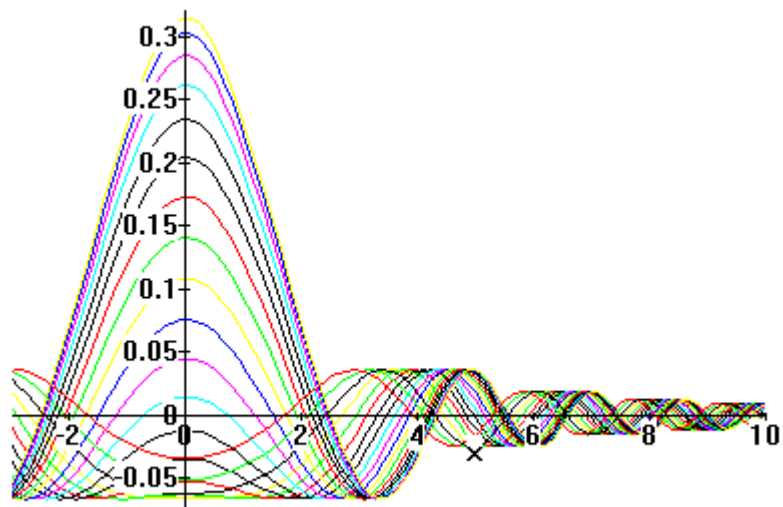


# Einführung in das Maple Computeralgebrasystem



# Einführung in das Maple Computeralgebrasystem

Universität Osnabrück  
- Rechenzentrum -  
Frank Elsner      F.Elsner@rz.uni-osnabrueck.de  
Albrechtstraße 28  
D-49076 Osnabrück

Version: 1.2  
Stand: 97/01/21

Maschinenlesbar (PostScript Format) per anonymous FTP:  
[ftp.rz.Uni-Osnabrueck.DE:pub/reports/rz/maple\\_tutorial-1.2.ps](ftp://rz.Uni-Osnabrueck.DE/pub/reports/rz/maple_tutorial-1.2.ps)

## Inhaltsverzeichnis

1. Einleitung.....	4
1.1. Zielsetzung.....	4
1.2. Typographische Konventionen .....	4
2. Überblick .....	5
2.1. Funktionsumfang .....	5
2.2. Architektur.....	6
3. Einfache Beispiele und Tips für Einsteiger .....	9
3.1. Einsteigen in Maple ... ..	9
3.2. Tips für Einsteiger .....	9
3.3. Maschinenlesbare Dokumente und Tutorials .....	10
4. Beschreibung der Benutzerschnittstelle .....	12
4.1. Elemente .....	12
4.2. Menüzeile und Arbeitsblatt.....	12
4.3. Hilfesystem .....	14
4.4. Regionen im Arbeitsblatt.....	14
4.5. Wichtige Menüpunkte und Funktionstasten .....	16
5. Datentypen.....	18
5.1. Ermitteln des Datentyps.....	19
6. Grafiken .....	21
6.1. Überblick .....	21
6.2. Einige Beispiele ... ..	23
6.3. Einfügen einer Grafik in ein Arbeitsblatt oder ein Dokument.....	24
6.4. Ausgeben einer Grafik in eine Grafik-Datei.....	24
6.5. Erzeugen und Abspielen einer Animation .....	24
7. Programmieren in Maple .....	26
8. Einlesen von Daten aus externen Dateien.....	28
9. Erzeugen von C, FORTRAN und LaTeX Code .....	29
10. Zusätzliche Pakete .....	30
11. Literaturhinweise .....	31
11.1. Weitere Informationsquellen .....	32

## 1. Einleitung

In diesem Kapitel wird die Zielsetzung und der Aufbau des Handbuches beschrieben.

### 1.1. Zielsetzung

Dieses Dokument wendet sich an Benutzer aller Fachrichtungen, die mathematische und statistische Aufgabenstellungen mit dem **Maple Computeralgebrasystem** bearbeiten wollen. Grundlegende Kenntnisse der Datenverarbeitung und der Bedienung graphischer Benutzeroberflächen wie z.B. Microsoft Windows werden vorausgesetzt, Programmierkenntnisse sind hilfreich, aber nicht unbedingt erforderlich.

Die Benutzerschnittstelle unter **Microsoft Windows** und einige interessante Aspekte des Funktionsumfang von **Maple** werden anhand einer kurzen Beschreibung schrittweise und mit kurzen erläuternden Beispielen vorgestellt. Dieses Dokument beschränkt sich auf grundlegende Informationen. Weiterführende Literatur ist im Kapitel **Literaturhinweise** aufgelistet.

### 1.2. Typographische Konventionen

Folgende typographischen Konventionen werden verwendet.

<b>Fett</b>	kennzeichnet <b>Maple</b> Kommandos und Funktionen, wichtige Textpassagen oder erstmalig genannte Begriffe. Beispiel: Die Funktion <b>sin</b> berechnet ...
Courier	kennzeichnet Namen von Dateien und Variablen. Beispiel: Die Datei <code>PLOT.EPS</code> ...
<i>Kursiv</i>	kennzeichnet Variablen (Platzhalter), die durch konkrete Werte zu ersetzen sind, und englische Fachbegriffe. Beispiel: Die Funktion $f(x)$ soll ...
□	Programmbeispiele sind zusätzlich durch eine Umrahmung gekennzeichnet. Eingaben sind dabei fett ausgezeichnet. Beispiel: <code>evalf(Pi,5); # Berechne Pi auf 5 # Stellen. 3.1416</code>
[...]	In Syntaxbeschreibungen bezeichnen eckige Klammern [...] Syntaxelemente, die nicht unbedingt eingegeben werden müssen (optionale Argumente).
# ...	In Programmen und Syntaxbeschreibungen bezeichnen Blöcke der Form # (Shell-like) einen Kommentar.
<i>File-&gt;Save</i>	Die Auswahl von Menüpunkten, z.B. durch Anklicken mit der Maus, wird durch Kursivschrift und durch Pfeile auf die entsprechenden Menüpunkte symbolisiert.

## 2. Überblick

In diesem Kapitel wird ein Überblick über den Funktionsumfang und die Architektur von **Maple** gegeben.

### 2.1. Funktionsumfang

Das **Maple** System (im folgenden mit **Maple** abgekürzt) gehört wie **Mathematica**, **Derive** oder **Axiom** zur Klasse der Computer-Algebra Systeme (*symbolic math package*, SMP).

**Maple** ist ein interaktives Werkzeug zur Lösung numerischer und symbolischer mathematischer Probleme und zur grafischen Darstellung der Ergebnisse. Es steht auf allen wichtigen Plattformen zur Verfügung ("vom 80386 bis zum Supercomputer").

**Maple** verwendet eine eigene, an Pascal angelehnte Programmiersprache und besitzt ca. 2500 vordefinierte Funktionen. Im Unterschied zu Programmiersprachen wie C, FORTRAN, Pascal oder APL und den zugehörigen numerischen Bibliotheken kann **Maple** auch Ausdrücke mit symbolischen Elementen auswerten oder vereinfachen.

Die Berechnung der Inversen einer 2x2 Matrix ist beispielsweise in C nur dann möglich, wenn alle Matrixelemente bekannt sind. **Maple** berechnet jedoch die Inverse auch in allgemeiner Form.

Lade das Paket **linalg** (Lineare Algebra) und definiere die allgemeine 2x2 Matrix A.

```
with(linalg):  
A:=matrix(2,2,[a,b,c,d]);
```

$$A := \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Berechne die Inverse von A.

```
inverse(A);
```

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

**Maple** beinhaltet Bibliotheksfunktionen für eine Vielzahl mathematischer Anwendungsgebiete, unter anderen:

- allgemeine Mathematik
- Kombinatorik
- Differential- und Integralrechnung
- Lineare Algebra
- Lineare Optimierung
- Statistik

Die berechneten Ergebnisse werden von **Maple** in mathematischer Notation<sup>1</sup> dargestellt, so z.B. Potenzen und Wurzeln als Ergebnis der Integration (**Maple** Funktion: `int`) der Funktion Wurzel aus  $(1-x^2)$  (**Maple** Darstellung: `sqrt(1-x^2)`):

Berechne das Integral über  $(1-x^2)^{1/2}$

```
int(sqrt(1-x^2),x);
```

Ergebnis:

$$\frac{1}{2}x\sqrt{1-x^2} + \frac{1}{2}\arcsin(x)$$

Zur Darstellung von mathematischen Funktionen und Datenpunkten wie z.B. Meßwerten stehen in **Maple** eine Vielzahl von graphischen Bibliotheksfunktionen zur Verfügung, unter anderen:

- x-y Grafiken
- x-y-z Grafiken
- parametrisierte Kurven und Flächen
- Konturlinien und -flächen
- 2D- und 3D-Animationen

Spezielle Pakete existieren zur Darstellung von geometrischen Objekten (*Plots Package*) und zur Darstellung von Lösungen einer Differentialgleichung (*DEplots Package*).

Das Titelbild zeigt eine Familie gedämpfter Cosinus Funktionen<sup>2</sup>. Zur Berechnung und Ausgabe wurden folgende Kommandos verwendet:

Erzeuge eine Menge von Cosinusfunktionen (parametrisiert über  $a$ ) und zeichne ihre Graphen im Intervall von 3 bis 10.

```
s:={};          # leere Menge
for i from 2 by 2 to 40 do
  a:=0.1*i;
  s:=s union
    {cos((x^2+a^2)/4)/(x^2+a^2+Pi)};
od:
plot(s, x=-3..10);
```

## 2.2. Architektur

**Maple** besteht aus einer **Benutzerschnittstelle**, einem in C geschriebenen **Kern**, der mathematische Berechnungen ausführt, und einer **Bibliothek** von Funktionen, die bei Bedarf vom Kern geladen werden.

Eine Maple Sitzung vollzieht sich in einem interaktiven Dialog zwischen dem Benutzer, der "Fragen" stellt, und dem Maple System, das "Antworten" auf diese Fragen gibt.

---

<sup>1</sup> Dies erfordert allerdings eine graphische Benutzeroberfläche wie z.B. Motif oder Windows.

<sup>2</sup> Alle Maple Plots und Formeln in diesem Dokument sind über die Windows Zwischenablage im Bitmap Format eingefügt worden. Die Qualität der Darstellung in einer Maple Sitzung ist weitaus besser.

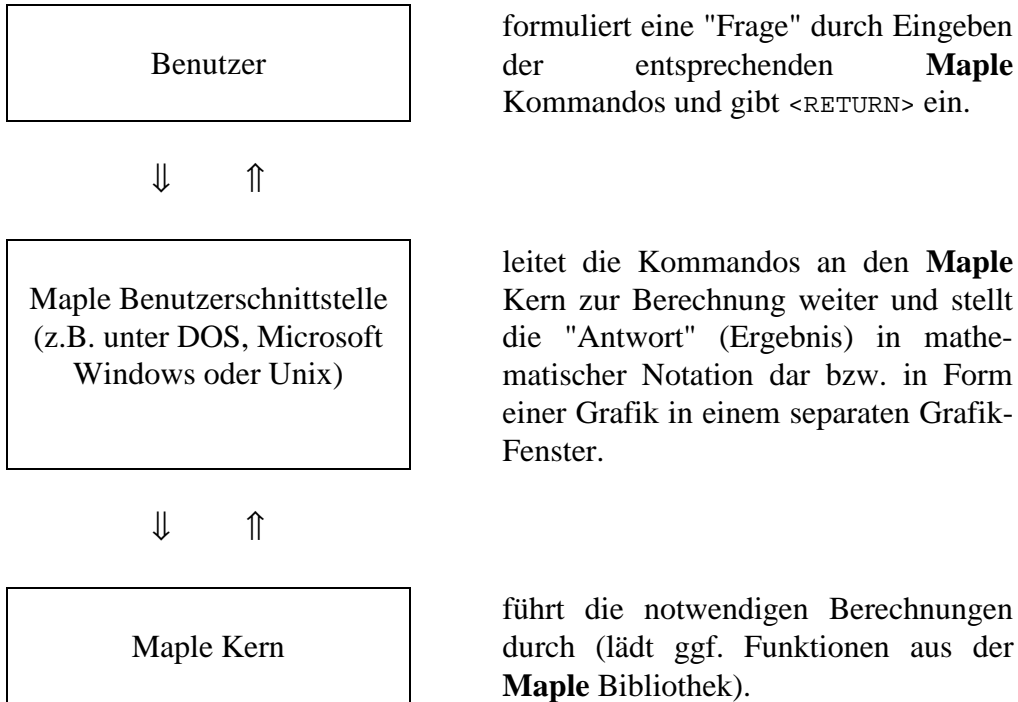
## Überblick

---

Im einzelnen sieht das Zusammenspiel folgendermaßen aus:

## Überblick

---



Diese Architektur hat z.B. folgende Vorteile:

- Durch die Trennung von Benutzerschnittstelle und mathematischen Kern kann eine Benutzerschnittstelle optimal auf eine Plattform angepaßt werden, z.B. durch Ausnutzung vorhandener Technologien wie OSF/Motif oder Microsoft Windows.
- Der mathematische Kern und die Bibliothek sind vom Funktionsumfang auf allen Plattformen identisch, die Bibliothek ist sogar als Datei plattformübergreifend ohne Änderungen austauschbar (Maple .m Format).

### 3. Einfache Beispiele und Tips für Einsteiger

In diesem Kapitel werden einige einfache Aufgabenstellungen und korrespondierende Maple Kommandos behandelt. Zusätzlich werden häufig auftauchende "Startschwierigkeiten" genannt und Lösungen vorgestellt.

#### 3.1. Einsteigen in Maple ...

Die ursprüngliche Problemstellung wird als "Frage" formuliert und die "Antwort" von Maple im Anschluß berechnet und angezeigt:

Problem	Frage/Antwort
Addiere 5 und $7^4$ .	<code>5 + 7^4;</code> 2406
Löse die Gleichung $x^2-2=0$ .	<code>solve(x^2 - 2 = 0, x);</code> $\sqrt{2}, \sqrt{2}$
Berechne die numerischen Werte der Lösung.	<code>evalf(");</code> 1.414, -1.414
Löse die Gleichung $ax+b=0$ .	<code>solve(a*x+b=0,x);</code> -b/a
Substituiere in der Lösung $a=5.0$ und $b=10.0$ .	<code>subs(a=5.0,b=10.0,");</code> -2.000000000
Differenziere $\sin(x)\cos(x)$ nach $x$ .	<code>diff(sin(x)*cos(x),x);</code> $\cos(x)^2 - \sin(x)^2$
Berechne den Grenzwert von $\sin(x)/x$ für $x \rightarrow 0$ .	<code>limit(sin(x)/x,x=0);</code> 1
Berechne den Grenzwert von $\text{Tangens}(x)$ für $x \rightarrow \pi/2$	<code>limit(tan(x),x=Pi/2);</code> undefined

#### Bemerkungen:

- Jede "Frage" (Kommando) muß mit einem Semikolon ; abgeschlossen werden. Erst dann beginnt **Maple** mit der "Beantwortung" (Berechnung und Anzeige des Ergebnisses).
- Der letzte berechnete Ausdruck kann zur Vermeidung von Schreiarbeit mit einem Anführungszeichen " abgekürzt werden.
- Die mathematische Konstante  $\pi$  wird in Maple mit `Pi` bezeichnet, ein undefiniertes Ergebnis (Weiß ich nicht!) mit `undefined`. Die arithmetischen Operatoren werden mit `+`, `-`, `*`, `/` und `^` (Potenz) bezeichnet.
- Dezimalstellen werden durch einen Punkt (nicht durch ein Komma) abgetrennt.

#### 3.2. Tips für Einsteiger

- Schließen Sie jedes **Maple** Kommando mit einem Semikolon ab. Beispiel: `evalf(Pi);`
- Weisen Sie Werte in der Form `x := expression` zu. Beispiel: `x := 5.0;`  
(Achtung: **Mathematica** verwendet nur das Gleichheitszeichen für Zuweisungen)
- Geben Sie das Multiplikationszeichen `*` (Potenzierungsoperator `^`) an, wenn Sie Ausdrücke miteinander multiplizieren (potenzieren) wollen wie z.B. in den Ausdrücken `3*x` und `3^7`. (Achtung: **Mathematica** erkennt z.B. `3 x` als `3*x`, **Maple** meldet einen Syntaxfehler).
- Beachten Sie, daß (wie z.B. unter Unix) Groß- und Kleinschreibung signifikant ist. Beispiel: `sin(x)` ist nicht gleich `sin(X)`
- Kommentieren Sie Ihre Kommandos, entweder durch `# comment` oder durch Textregionen in Arbeitsblättern (siehe unten).
- Beachten Sie den Unterschied zwischen Variablen (*symbols*), denen Sie bereits einen Wert zugewiesen haben, und Variablen, die für sich stehen. Führen Sie ggf. explizit das Kommando `x:='x'` aus, um die Variable (das Symbol) `x` wieder als frei zu kennzeichnen. Das Kommando `anames()` listet alle bisher verwendeten Variablennamen.
- Geben Sie das Kommando `evalf(expr,digits)` an, um den Ausdruck `expr` auf `digits` Ziffern genau numerisch auszuwerten. Geben Sie das Kommando `Digits:=10` ein, um alle nachfolgenden numerischen Berechnungen automatisch auf 10 Stellen genau durchzuführen.
- Verwenden Sie für Zeichenketten mit Sonderzeichen aller Art die Schreibweise `'string'`. (Fast alle Sonderzeichen haben für **Maple** eine spezielle Bedeutung.)  
Beispiel: `read('userdefs1.m');`
- Kürzen Sie den letzten berechneten Ausdruck zur Vermeidung von Schreiarbeit mit `"` ab. Beispiel: `x:=sqrt(2); evalf(");`
- Geben Sie das `restart` Kommando ein, um **alle** bisher durchgeführten Definitionen zu löschen.
- Beachten Sie, daß im Kern jeweils nur die zuletzt berechneten Werte gespeichert werden, auch wenn Sie vorher berechnete Werte noch am Bildschirm sehen.  
Beispiel: `x:= 5; ...; x := 'x'; x; # x is no longer set to 5.`

### 3.3. Maschinenlesbare Dokumente und Tutorials

Besonders empfehlenswert zum Kennenlernen sind die folgenden maschinenlesbaren Dokumente und Tutorials, die zum Lieferumfang von **Maple** gehören:

- `QUIKTOUR.MS`  
(Überblick über **Maple**)
- `SUMMARY.MS`  
(Neue Möglichkeiten in Maple V Release x)
- Online Tutorial

Falls Sie vor dem Weiterlesen die genannten Dokumenten oder das Online Tutorial bearbeiten wollen, gehen Sie folgendermaßen vor (1, 2, 2a, 4 bzw. 1, 3, 3a, 4):

1. Starten Sie **Maple** (z.B durch Anklicken des **Maple** Icons).

## Einfache Beispiele und Tips für Einsteiger

---

2. Laden Sie die Datei (hier z.B. QUIKTOUR.MS) durch Auswahl der Menüpunkt *File*, Unterpunkt *Open*, Verzeichnis: C:\MAPLEV2\LIB o.ä..
- 2a. Folgen Sie den Anweisungen im Arbeitsblatt QUIKTOUR.MS.
3. Geben Sie ggf. in einer Eingabezeile dasKommando `tutorial();` ein.
- 3a. Folgen Sie nun den Anweisungen des Tutorials.
4. Beenden Sie ggf. **Maple** durch Auswahl der Menüpunkt *File*, Unterpunkt *Exit*.

### 4. Beschreibung der Benutzerschnittstelle

Im folgenden wird die grafische Benutzerschnittstelle von **Maple** für Microsoft Windows behandelt, die auch weitgehend der Benutzerschnittstelle von **Maple für X/Motif** entspricht. Für weniger gut ausgerüstete PCs existiert alternativ eine zeilenorientierte Benutzerschnittstelle unter DOS mit geringen Speicherplatzanforderungen (2MB RAM Minimum).

#### 4.1. Elemente

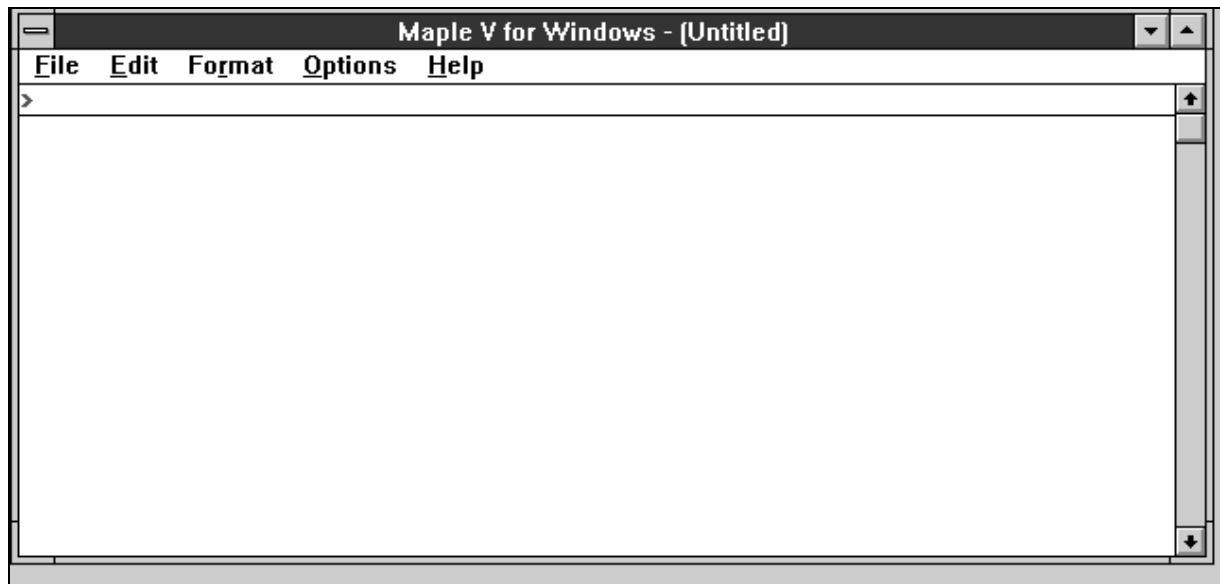
Sie arbeiten wie unter Microsoft Windows gewohnt in einem Fenstersystem mit folgenden Elementen:

Element	dient ...
Menüzeile (Menu Bar)	zur Auswahl von Menüpunkten.
Arbeitsblatt (Worksheet Window)	zum Erstellen eines integrierten Dokuments mit Text, Formeln und Grafiken.
Statusanzeige (Status Window)	zur Anzeige von Statusinformationen, optional.
Hilfe-Fenster (Help Window)	zur Anzeige von Hilfeinformationen, Menüpunkt <i>Help</i> .
2D-Grafik-Fenster (2D Plot Window)	zur Anzeige von 2D Plots, wird automatisch nach einem <b>plot</b> Kommando erzeugt.
3D-Grafik-Fenster (3D Plot Window)	zur Anzeige von 3D Plots, wird automatisch nach einem <b>plot3d</b> Kommando erzeugt.
Animations-Fenster (Animation Window)	zur Anzeige von Animationen, wird automatisch nach einem <b>animate</b> oder <b>animate3d</b> Kommando erzeugt.

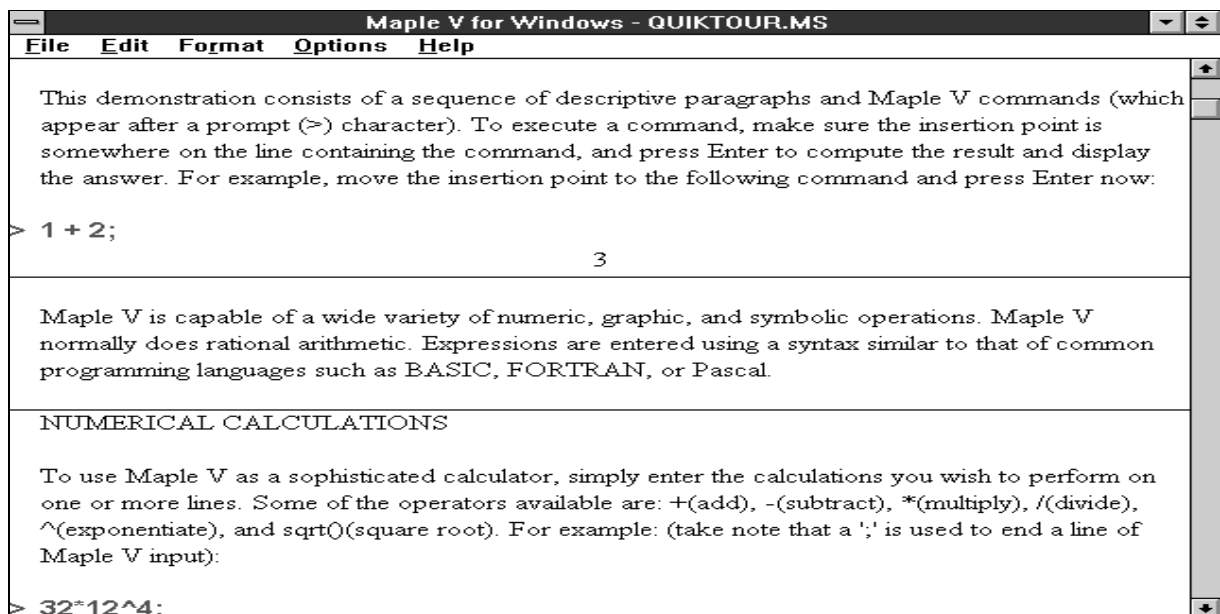
#### 4.2. Menüzeile und Arbeitsblatt

Die folgende Abbildung ist ein "Schnappschuß" des Bildschirms nach dem Starten einer Maple Sitzung. Er zeigt die Menüzeile mit den Menüpunkten *File*, *Edit*, ..., *Help*, und ein bis auf eine Eingaberegion, leeres Arbeitsblatt. Die Eingaberegion enthält per Voreinstellung das Zeichen '>' als Eingabeaufforderung.

## Beschreibung der Benutzerschnittstelle



Nach dem Laden eines Arbeitsblattes (*Worksheet*) verändert der Bildschirm sein Aussehen folgendermaßen:



### Bemerkungen:

- Innerhalb einer **Maple** Sitzung arbeiten Sie mit einem Arbeitsblatt (*Worksheet*), das in voneinander getrennten Bereichen (*Regions*) Kommandos, Ergebnisse und begleitenden Text bzw. Grafiken enthält. Die Abbildung zeigt ein Arbeitsblatt, das Eingabe-, Ausgabe- und Textregionen enthält.
- Sie können ein Arbeitsblatt jederzeit über den Menüpunkt *File->Save* abspeichern bzw. über den Menüpunkt *File->Print* ausdrucken und besitzen damit ein vollständiges Protokoll Ihrer Sitzung.
- Durch Laden eines Arbeitsblattes können Sie die entsprechende **Maple** Sitzung zu beliebigen Zeiten fortsetzen.
- Arbeitsblätter sind plattform-unabhängig; d.h. ein z.B unter Microsoft Windows erstelltes Arbeitsblatt kann unter UNIX eingelesen und weiterbearbeitet werden.

### 4.3. Hilfesystem

Eine hervorragende englische Beschreibung der Benutzerschnittstelle erhalten Sie im **Maple** Hilfe System.

Wählen Sie folgende Menüpunkte aus: *Help->Interface Help*. In dem nun angezeigten Inhaltsverzeichnis können Sie über Verweise (*Hyperlinks*) folgende Themen auswählen:

- Using Maple V in the Windows Environment
- Types of Maple Windows
- Maple Menu Commands
- Working with Worksheets
- Plotting in Maple
- Maple's Help Facility

### 4.4. Regionen im Arbeitsblatt

Sie arbeiten unter Windows mit einem Arbeitsblatt (*worksheet*), das Sie in folgende Typen von Regionen oder Bereichen (*regions*) einteilen können:

<b>Region</b>		<b>dient zur ...</b>
<b>Eingaberegion</b>	<i>Input Region</i>	zur Eingabe von Kommandos an den Maple Kern
<b>Ausgaberegion</b>	<i>Output Region</i>	zur Ausgabe der Berechnungen des Maple Kerns
<b>Textregion</b>	<i>Text Region</i>	zur Eingabe von überleitendem oder erläuterndem Text
<b>Grafikregion</b>	<i>Graphics Region</i>	zum Einfügen von beliebigen Grafiken aus der Zwischenablage

#### **Bemerkungen:**

- Beim Start von Maple wird zunächst nur eine *Input Region* angezeigt. Nach Eingabe eines Kommandos wird das berechnete Ergebnis von Maple automatisch in einer *Output Region* angezeigt; d.h. auf eine Eingaberegion folgt immer eine Ausgaberegion.
- Andere Regionen werden nur auf Anforderung erzeugt, z.B. durch Auswahl eines entsprechenden Menüpunktes bzw. durch Einfügen (*Paste*) von Grafiken aus der Zwischenablage oder durch Umwandlung einer Eingaberegion in eine Textregion.
- Eine Grafik-Region kann komplett kopiert oder gelöscht werden. Die in ihr enthaltene Bitmap Grafik kann nicht direkt bearbeitet werden. Hierzu können Sie allerdings ein Zeichenprogramm verwenden, das Bitmap Grafiken über die

## Beschreibung der Benutzerschnittstelle

---

Zwischenablage einlesen und bearbeiten kann. Andere Möglichkeiten werden im Abschnitt **Speichern von Grafiken** behandelt.

### 4.5. Wichtige Menüpunkte und Funktionstasten

Die folgende Aufstellung gibt einen kurzen Überblick über das Menüsystem:

Menüpunkt *File*:

<i>New</i>	öffnet ein neues Arbeitsblatt ( <i>Worksheet</i> )
<i>Open</i>	öffnet ein vorhandenes <i>Worksheet</i> (vorgeschriebene Endung: <i>ms</i> , z.B. <i>summary.ms</i> )
<i>Save</i>	speichert ein <i>Worksheet</i>
<i>Save as ...</i>	speichert ein <i>Worksheet</i> unter einem neuen Namen
<i>Print</i>	druckt das aktuelle <i>Worksheet</i>
<i>Exit</i>	beendet die Maple Sitzung

Menüpunkt *Edit*:

<i>Cut</i>	schneidet den markierten Bereich aus und stellt ihn in die Zwischenablage
<i>Copy</i>	kopiert den markierten Bereich in die Zwischenablage
<i>Paste</i>	kopiert den Inhalt der Zwischenablage (Text oder Grafik) an die Cursorposition

Menüpunkt *Format*:

<i>Input (Text) Region</i>	(F5)	verwandelt eine Text-Region in eine Eingabe-Region und umgekehrt (Wechselschalter).
<i>Split Region</i>	(F3)	teilt eine Region an der Cursorposition.
<i>Join Region</i>	(F4)	verbindet zwei Regionen.
<i>Insert Page Break</i>		fügt einen erzwungenen Seitenumbruch ein.
<i>Insert New Region</i>		fügt eine neue Region ein.

Menüpunkt *Options*:

<i>Show Separators</i>	(F9)	zeigt (keine) Trennlinien zwischen Regionen an.
<i>Show Prompts</i>		zeigt eine (keine) Eingabeaufforderung in jeder Eingabe-Region an.

## Beschreibung der Benutzerschnittstelle

---

<i>Continuous/Insert/Insert at End Mode</i>	wechselt zwischen den 3 möglichen Modi für die Platzierung von Ergebnissen bei erneuter Auswertung von Ausdrücken..
---	---

Menüpunkt *Help*:

<i>Browser ...</i>	öffnet das <i>Browser Window</i> , von dem aus Syntaxbeschreibungen aller Funktionen ausgewählt werden können.
<i>Interface Help ...</i>	öffnet das <i>Interface Help Window</i> .

## 5. Datentypen

Eine großer Vorteil von **Maple** gegenüber konventionellen Programmiersprachen ist die einfache Behandlung unterschiedlicher Datentypen. Jeder Operator und jede Funktion ist in der Lage, beliebige Kombinationen von Argumenten beliebiger Datentypen zu verarbeiten<sup>3</sup>. Sie können z.B. rationale Zahlen addieren und erhalten als Ergebnis wieder eine rationale Zahl:

Addiere  $4/7$  und  $9/2$ .

$$4/7 + 9/2;$$

$$\frac{71}{14}$$

Sie können aber auch unbestimmte Integrale symbolisch oder bestimmte Integrale symbolisch (falls möglich) oder approximativ numerisch berechnen. **Maple** liefert einen Ausdruck einfach als Ergebnis zurück, wenn es ihn nicht weiter vereinfachen oder berechnen kann (siehe auch: `?assume`):

Berechne das Integral über

$\sin(t^2)$  im Intervall von 0 bis unendlich.

Berechne das bestimmte Integral in beliebigen Grenzen über

$\exp(-t^3)$ .

`int(sin(t^2), t=0..infinity);`

$$\frac{1}{4}\sqrt{2}\sqrt{\pi}$$

`int(exp(-t^3), t=lb..ub);`

$$\int_{lb}^{ub} e^{-t^3} dt$$

Für das Verständnis der Arbeitsweise von **Maple** ist es notwendig, einen Überblick über wichtige Datentypen und Funktionen zu ihrer Verarbeitung zu kennen. Insbesondere sind hier die Datentypen **Sequenz**, **Liste** und **Menge** zu nennen.

Die folgende Tabelle listet zunächst einige der nicht weiter zerlegbaren (atomaren) Datentypen auf:

Datentyp	Beispiel	Bemerkung
Ganze Zahlen	5;	Ganze Zahlen dürfen "beliebig" groß werden.
Rationale Zahlen	4/7;	Rationale Arithmetik ist exakt.
Fließkommazahlen	5.4E-7	Arithmetik mit Fließkommazahlen kann "beliebig" genau durchgeführt werden.

---

<sup>3</sup> Einigermaßen sinnvoll sollte das Argument allerdings sein, z.B. ist der Ausdruck `sqrt("Zahl")` auch für Maple nicht interpretierbar und verursacht einen Syntaxfehler.

## Datentypen

---

Zeichenketten	"Hallo World"	
Konstanten	E oder $\exp(1)$ I oder $\sqrt{-1}$ [-]infinity	

In den zusammengesetzten Datentypen **Sequenz**, **Liste** und **Menge** können Sie beliebige andere Datentypen zusammenfassen, z.B. auch Listen von Listen oder Mengen von Mengen.

Datentyp	Schreibweise	Beispiel
Sequenz	a, b, c, d, ...	S1 := 1, -1, 1, -1;
Liste	[a, b, c, d, ...]	L1 = [S1];
Menge	{a, b, c, d, ...}	M1 := {S1};

Die Verwendung von zusammengesetzten Datentypen soll an folgenden Beispielen demonstriert werden:

```

Erzeuge die Sequenz S2 der Zahlen 1, 2, 3, 4.  S2:=seq(i^2,i=1..4);
                                                1, 2, 3, 4
Wende auf jedes Element der Sequenz S2 die  map(x^2, S2);
Funktion x^2 an.                             1, 4, 9, 16
Erzeuge die Mengen M2 und M3.                M2:={a,b,c,d,1};
                                                M3:={a,c,4};
Schneide die Mengen M2 und M3.               M2 intersect M3;
                                                {a,c}
Verwandle die Liste L1 in eine Sequenz S3.   L1:=[a,b,c];
                                                S3:=op(L1) # op extracts 1. arg.
                                                a,b,c
    
```

Folgende weiteren Datentypen sind u.a. in **Maple** bekannt:

Datentyp	Schreibweise	Beispiel
Logische Ausdrücke (Ergebnis: Wahrheitswert)	true, false Operatoren: and, or, not, =, <>, < <=	if ( a > b ) ...;
Bereiche	begin..end	int(x^2, x=0..2);
Tabellen	a[index] := ...	Temp[Monday] := 15*C;
Matrizen	matrix(#rows, #cols, list)	with(linalg): matrix(2,2,[1,0,0,1]);

### 5.1. Ermitteln des Datentyps

Die wesentlichen Funktionen zum Untersuchen des Typs eines Ausdrucks und seiner Argumente lauten: **type**, **whattype**, **op**, **nops**

## Datentypen

---

Die Funktion **type** vergleicht einen Ausdruck mit einem vorgegebenen Datentyp und liefert den Wahrheitswert `true`, wenn der Ausdruck dem vorgegebenem Datentyp entspricht, die Funktion **whattype** liefert den Datentyp eines Ausdrucks (genauer: des Kopfes eines Ausdrucks). Die Funktion **op** dient zur Extraktion beliebiger Operanden (Argumente) eines Ausdruckes, die Funktion **nops** die Anzahl Argumente.

Die genannten Funktionen können z.B. in Prozeduren verwendet werden, um abhängig vom Datentyp der Argumente geeignete Aktionen auszuführen.

## 6. Grafiken

In diesem Kapitel werden einige Funktionen zum Erzeugen, Bearbeiten und Abspeichern von Grafiken behandelt, sowie einige Optionen zum Gestalten von Grafiken.

### 6.1. Überblick

Sie können mit Funktionen aus der **Maple** Bibliothek Grafiken und Animationen:

- in einem separaten Grafik-Fenster erzeugen,
- interaktiv bearbeiten (z.B. rotieren),
- mit anderen Grafiken überlagern,
- in die Windows Zwischenablage kopieren und in ein Dokument einfügen, insbesondere natürlich auch in das aktuelle Maple Arbeitsblatt<sup>4</sup>.
- in einem Grafik-Format (z.B. PostScript) in eine Datei abspeichern,

Die Funktionen lassen sich in folgende Gebiete einteilen (siehe auch **Help Browser**):

Gebiet	Beispiele für Funktion	
2D	plot display densityplot fieldplot gradplot conformalplot	
3D	plot3d display3d fieldplot3d gradplot3d contourplot surfdata spacecurve	
2D Animation	animate display	
3D Animation	animate3d display3d	
Differentialgleichungen	DEplot[1 2] dfieldplot phaseportrait	
Statistik	boxplot histogram	

Die folgende Übersicht zeigt die Syntax einiger der oben genannten Funktionen zum Erzeugen von Abbildungen und Animationen (*plot functions*)<sup>5</sup>:

---

<sup>4</sup> Für die Windows Version 5.2 gilt hier allerdings die Einschränkung, daß der Austausch nur im Bitmap Format, also in relativ schlechter Qualität, erfolgt.

<sup>5</sup> Bis auf `plot` und `plot3d` befinden sich alle Funktionen in Paketen (*packages*), die einmalig explizit geladen werden müssen, z.B. das `plot` Paket durch: `with(plots):`

## Grafiken

---

Syntax	Beschreibung
<code>animate(f(x,t), x=a..b, t=c..d, frames=n [, ...]);</code>	erzeugt eine Animation von $f(x,t)$ . (Es werden zunächst $n$ Graphen von $f(x)$ für $t=c, t=c+(d-c)/n, \dots, t=d$ erzeugt und dann in schneller Folge abgespielt.)
<code>animate3d(f(x,y,t), ...)</code>	erzeugt eine Animation von $f(x,y,t)$ .
<code>display({p1, p2, ...}, options)</code>	zeigt eine vorher erzeugte 2D-Struktur bzw. eine Menge von 2D-Strukturen an (z.B. für Darstellen mit veränderten Optionen oder Überlagernnützlich).
<code>display3d({p1, p2, ...}, options)</code>	zeigt eine 3D-Struktur bzw. eine Menge von 3D-Strukturen an.
<code>plot(f(x), x=a..b, options);</code>	erzeugt den Graphen von $f(x)$ .
<code>plot3d(f(x,y), x=a..b, y=a..b, options);</code>	erzeugt das Funktionsgebirge von $f(x,y)$

Das Aussehen einer Grafik kann über Optionen beeinflusst werden<sup>6</sup>. Die Voreinstellungen für die folgenden Beispiele sind jeweils fett ausgezeichnet.

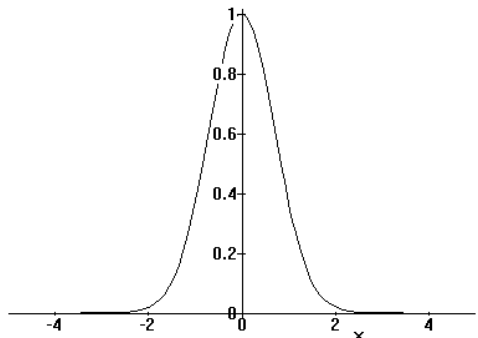
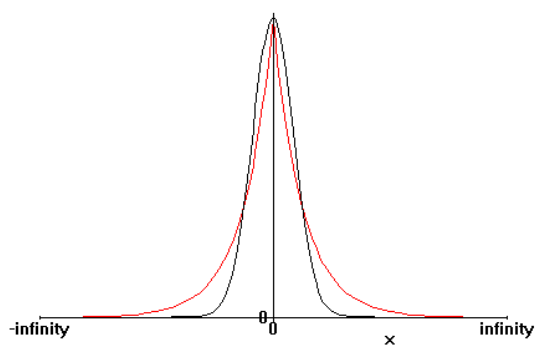
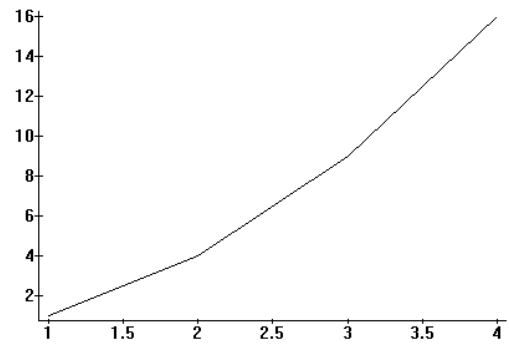
Option	Beispiel
<code>style=<b>LINE</b>   PATCH   POINT</code>	<code>plot(sin(x), style=POINT);</code> # Linientyp: Punkte
<code>scaling=<b>UNCONSTRAINED</b>   CONSTRAINED</code>	
<code>numpoints=n</code>	<code>plot(sin(x)/x, x=0..1, numpoints=300);</code> # Anzahl Stuetzstellen

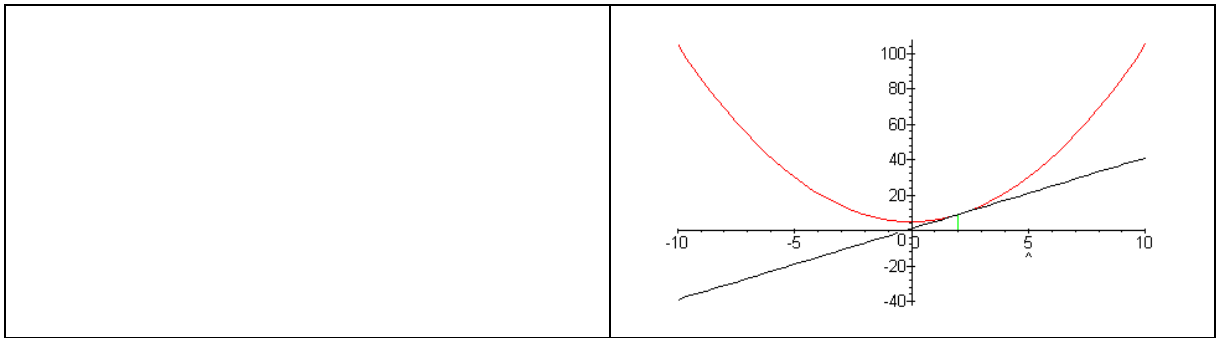
Die folgenden Abschnitte behandeln einige Beispiele. Weitere Informationen erhalten Sie im Hilfesystem über die Menüpunkte *Help*-> *Browser* -> *Graphics* -> ... oder direkt über das Kommando `?topic`, wie z.B. `?plot`.

---

<sup>6</sup> Die Optionen treffen nicht auf alle Plot Funktionen zu, z.B. ist die Option `frames` nur für `animate` und `animate3d` zulässig.

6.2. Einige Beispiele ...

<p>Zeichne die Funktion <math>\exp(-x^2)</math> im Intervall von -5 bis 5.</p>	<pre>plot(exp(-x^2), x=-5..5);</pre>
<p>Bemerkungen:</p> <ul style="list-style-type: none"> <li>Die Grafik wird gemäß den Voreinstellungen (<i>defaults</i>) oder den gewählten Optionen dargestellt.</li> </ul>	
<p>Zeichne die Graphen der Funktionen <math>\exp(-x^2)</math> und <math>\exp(- x )</math> über die reelle Achse.</p>	<pre>plot({exp(-x^2), exp(-abs(x))} x=-infinity, infinity, title='Graphen von exp(-x^2) und exp(- x )');</pre>
<p>Bemerkungen:</p> <ul style="list-style-type: none"> <li>Die Darstellung des Graphs kann über Optionen (programmiert oder interaktiv) in vielfältiger Weise modifiziert werden, z.B. durch Hinzufügen eines Titels und Achsenbeschriftungen.</li> <li>Durch Angabe einer Menge von Funktionen können mehrere Graphen überlagert werden, z.B.  <pre>plot({sin, cos}, ...);</pre> </li> </ul>	<p>Graphen von <math>\exp(-x^2)</math> und <math>\exp(- x )</math></p> 
<p>Zeichne ein Liniendiagramm des Vektors v.</p>	<pre>n:=4; v:=array(1..n); S1:=NULL; for i to n do;   v[i]:=i^2;   S1:=S1, i, v[i]; od; plot([S1]);</pre>
<p>Bemerkungen:</p> <ul style="list-style-type: none"> <li><code>plot</code> erwartet Datenpunkte in Form einer Liste:  <math>[x_1, y_1, x_2, y_2, \dots, x_n, y_n]</math></li> <li>Per Voreinstellung wird eine Linie durch die Datenpunkte gelegt.</li> <li>Dieses Beispiel läßt sich einfach auf den Fall umschreiben, daß die Daten als Meßpunkte in einer externen Datei vorliegen, siehe Kapitel <b>Verarbeiten formatierter Daten</b>.</li> </ul>	
<p>Zeichne die Funktion <math>x^2+5</math> und die Tangente für den Punkt <math>x=2</math>.</p>	<pre>with(student): showtangent(x^2+5, x=2);</pre>



### 6.3. Einfügen einer Grafik in ein Arbeitsblatt oder ein Dokument

Gehen Sie zum Einfügen einer Grafik in das aktuelle Maple Arbeitsblatt oder in ein anderes Dokument folgendermaßen vor:

1. Kopieren Sie die Grafik über den Menüpunkt *Edit->Copy* in die Windows Zwischenablage.
2. Wechseln Sie in Ihre **Maple** Sitzung oder zu einer anderen Windows Anwendung und fügen Sie die Grafik über den Menüpunkt *Paste* (oder ähnlich) in das aktuelle **Maple** Arbeitsblatt oder ein Dokument ein. **Maple** erzeugt beim Einfügen einer Grafik in das aktuelle Arbeitsblatt automatisch eine neue Grafik-Region.

### 6.4. Ausgeben einer Grafik in eine Grafik-Datei

Sie können eine Grafik auch direkt in einer Grafik-Datei abspeichern.

Liste alle verfügbaren Grafik-Formate auf.  
Leite alle folgenden Grafiken in die Datei `plot1.eps` um<sup>7</sup>. Verwende als Grafik-Format `postscript`<sup>8</sup>.  
Erzeuge eine beliebige Grafik.  
Leite die Grafik-Ausgabe wieder auf den Bildschirm (hier: UNIX X11 Terminal).

```
?plot[device]
interface
  (plotdevice=postscript,
   plotoutput='plot1.ps');

plot(...); # plots to graphics file
interface
  (plotdevice=x11);
```

### 6.5. Erzeugen und Abspielen einer Animation

---

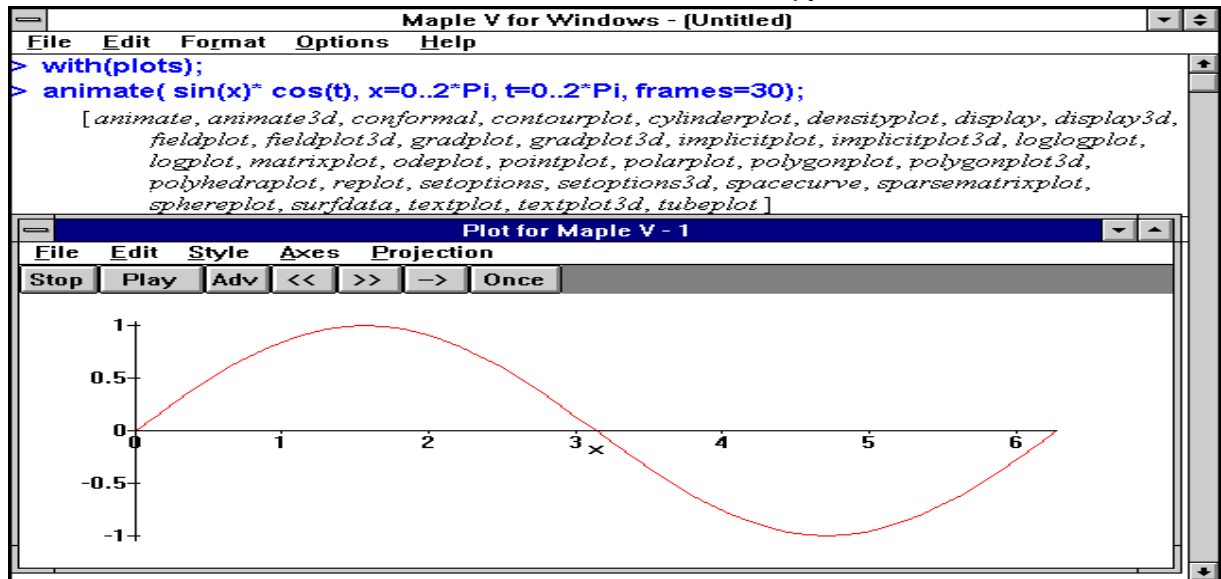
<sup>7</sup> In der Windows Version 5.2 funktioniert diese Vorgehensweise leider nicht, da ausschließlich Windows Druckertreiber verwendet werden. Umgehung: Verwenden Sie die DOS Version.

<sup>8</sup> Die PostScript Graphik ist leider sehr eigenwillig skaliert, siehe z.B. [7].

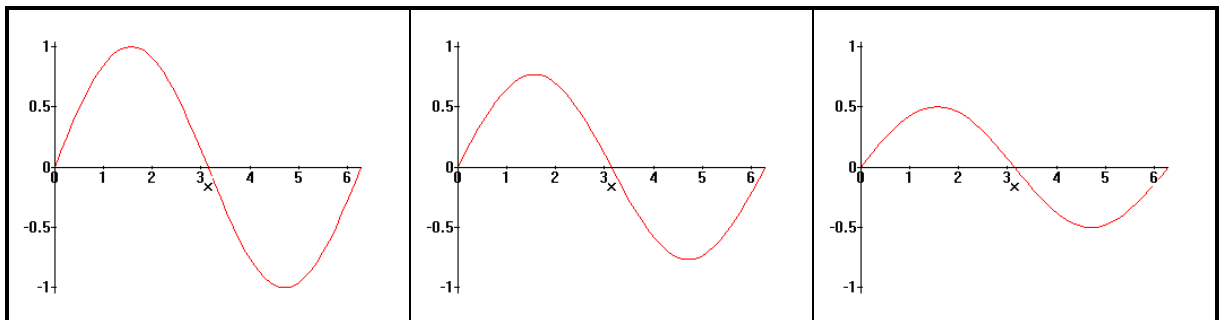
## Grafiken

Erzeuge eine Animation der Funktion  
 $f(x)=\sin(x)*\cos(t)$  (Schwebung).

```
animate(sin(x)*cos(t),  
x=0..2*Pi,  
t=0..Pi/4,  
frames=8);
```



Die Benutzerführung im *Animation Window* entspricht der eines Videorecorders, d.h. die Sequenz von Graphen kann z.B. vorwärts oder rückwärts abgespielt werden und angehalten werden. Aus Platzgründen werden im folgenden nur 3 Momentaufnahmen gezeigt:



## 7. Programmieren in Maple

Sie können Maple Kommandos mittels einer eingebauten **Maple Programmiersprache** zu Maple Funktionen bzw. Prozeduren zusammenfassen. Die Maple Programmiersprache ist Pascal-ähnlich. Es ist empfehlenswert, für jede Funktion oder Prozedur einen Hilfetext zu erstellen.

(In Maple vordefinierte Funktionen (*built-ins*) oder Bibliotheksroutinen werden hier als Kommandos bezeichnet, um sie von benutzerdefinierten Funktionen oder Prozeduren zu unterscheiden.)

### Definieren eigener Funktionen

Syntax:	Beispiel:
<pre>f := (x,y,z, ...) -&gt;   expression using x,y,z, ...;</pre>	<pre>f := (x,y) -&gt; (x^2+y^2) /   (1+x^2+y^2): f(a,a);   2*a^2/(1+2*a^2)</pre>

### Definieren eigener Prozeduren

Syntax:	Beispiel:
<pre>f := proc(formal_parameter1   [: type],   formal_parameter2   [: type], ...)   local_variable1, ...;   statement(s); end: # : or ;</pre>	<pre>f := proc(g,x,n:integer)   exp_t:=taylor(g(x), x=0,n);   exp_p:=convert(exp_t,polynomial); end: f(sin, x, 5);   x - x^3/6</pre>

### Vorhandene Sprachelemente

Syntax:	Beispiel:
<pre>while condition do   ... od;</pre>	<pre>a:=7; b:=49; while b &lt;&gt; 0 do   d := irem(a,b); a:=b; b:=d; od;</pre>
<pre>for i from start   by change to finish do   ... od;</pre>	<pre>for i from 1 to 10 do   print(i, isprime(i)); od;</pre>
<pre>for x in expression do   ... od;</pre>	<pre>for x in [1,2,3,4,5] do   print(x, sum(i, i=1..x)) od;</pre>

## Programmieren in Maple

---

<pre>if condition1 then   ... elif condition2 then   ... else   ... fi;</pre>	<pre>if a &gt; 0 then   ..print('Expr is positive.');</pre> <pre>elif a &lt; 0 then   ..print('Expr is negative.');</pre> <pre>else   ..print('Expr is zero.');</pre> <pre>fi;</pre>
---	--

### Erzeugen von Hilfetexten

Syntax:	Beispiel:
<pre>'help/text/function' := TEXT( 'text string 1', 'text string 2', ..., 'text string n');</pre>	<pre>'help/text/f' := TEXT('', 'HELP FOR:', 'f - ...', 'CALLING SEQUENCE:', 'f(g, x, order)', 'PARAMETER(S):', ... 'SYNOPSIS:', ..., 'EXAMPLE(S):', ..., 'SEE ALSO:', ..., ):</pre> <pre>?f # opens new help window for f. # Text be copied and pasted.</pre>

### Speichern und Laden von Definitionen (u.a. Funktionen und Prozeduren)

Syntax:	Beispiel:
<pre>save('filename'); # saves def. ... read('filename'); # loads def.</pre>	<pre>save 'D:/MAPLE/USRLIB/FUNCL.M'; ... read 'D:/MAPLE/USRLIB/FUNCL.M';</pre>

### Bemerkung:

- Die Dateiendung `.m` bewirkt, daß die Funktion in einem komprimierten Format gespeichert wird.
- Sie können die Funktion `unapply` verwenden, um einen Ausdruck in eine Funktion umzuwandeln

Verwandeln Sie den Ausdruck  $p$  in eine Funktion  $f$  mit Argument  $x$ .

```
p:=x^2+7*x+5; # p polynomial
f:=unapply(p,x); # f function
f(1); # function eval.
```

## 8. Einlesen von Daten aus externen Dateien

Zum Einlesen und Ausgeben von formatierten Daten aus Dateien stehen folgende Funktionen zur Verfügung:

Funktion	Bedeutung	Beispiel
String := readline(file)	liest eine Zeile einer Datei.	readline('data', integer)
List := sscanf(String, format)	liest einen formatierten Datenwert (analog zu C).	sscanf(readline(...), 'd %d')
List := readdata(file, type, #col)	liest eine Tabelle ein (Datenwerte in Zeilen und Spalten).	readdata('data', float, 4)
printf(format, expr)	gibt Datenwerte formatiert aus (analog zu C).	printf('d\n', x);

Das folgende Beispielprogramm liest eine 3x3-Matrix mit reellen Zahlen aus einer externen Datei ein, berechnet die Transponierte und gibt sie unter Kontrolle von `printf` in eine externe Datei aus:

```
readlib(readdata):
data:=readdata('c:/datafile.in', float, 3):
A:=convert(data, matrix):
A_t:=linalg[transpose](A):
writeto('c:/datafile.out'): # writes to file from now on.
for i from 1 to 3 do
  printf('%8.3f %8.3f %8.3f\n', A_t[i,1], A_t[i,2], A_t[i,3]);
od;
writeto(terminal): # writes to terminal from now on.
```

## 9. Erzeugen von C, FORTRAN und LaTeX Code

Zum Erzeugen von FORTRAN, C und LaTeX Code stehen folgende Funktionen zur Verfügung:

Prozedur	Aufgabe	Beispiel
<code>fortran(expr, [filename='file', optimized])</code>	wandelt einen Ausdruck in FORTRAN Syntax um.	<code>D1:=Int(x^3/(1+x),x): fortran(D1);</code>
<code>readlib(C): C(expr,[...])</code>	wandelt einen Ausdruck in C Syntax um.	<code>D1:=Int(x^3/(1+x),x): C(D1, optimized);</code>
<code>latex(expr, 'filename')</code>	wandelt einen Ausdruck in LaTeX Syntax um.	<code>D1:=Int(x^3/(1+x),x): latex(D1, 'latex.out');</code>
<code>precision := double; # single</code>	legt den Datentyp für die folgenden Umwandlungen fest.	

Das folgende Beispielprogramm berechnet die Ableitung von  $\sin(x) \cdot \log(\log(x))$  und gibt sie als LaTeX Code aus:

```

expr1 := diff(sin(x)*log(log(x)),x);
latex(expr1);

```

Das Ergebnis lautet:

$$\cos(x) \ln(\ln(x)) + \frac{\sin(x)}{x \ln(x)}$$

### Bemerkung:

- Das Ergebnis kann direkt in eine Datei geschrieben werden (siehe Argument `filename` bzw. Funktion `writeto`).

### 10. Zusätzliche Pakete

Sie können sich alle verfügbaren **Pakete** (Zusammenfassung von Prozeduren eines Anwendungsgebietes) durch das Kommando `?packages` auflisten lassen:

```
?packages
approx:      Numerical Approximation
combinat:    combinatorial functions
DEtools:     Differential Equation Tools
diffforms:   differential forms
Gauss:       create domains of computation
GaussInt:    Gaussian Integers
geom3d:      three-dimensional Euclidean geometry
geometry:    two-dimensional Euclidean geometry
grobner:     Grobner bases
group:       permutation and finitely-presented groups
liesymm:     Lie symmetries
linalg:      linear algebra
logic:       Boolean logic
networks:    Graph Networks
np:          Newman-Penrose formalism
numtheory:   number theory
orthopoly:   orthogonal polynomials
padic:      Padic Numbers
plots:       graphics package
powseries:   formal power series
projgeom:    projective geometry
simplex:      linear optimization
stats:       statistics
student:     student calculus
totorder:    total orders on names

For information, type:
?<package>
where <package> is from the above list.
This will give a list of the functions available in the package.
To cause all functions in a package to be defined in the session,
type:
  with(<package>); # verbose loading
  with(<package>): # silent loading
```

Das folgende Beispielprogramm verwendet das Paket `DEtools` zur graphischen Darstellung der Lösung der Differentialgleichung  $y^{(4)} + 2y^{(2)} - \cos(x) = 3$  für  $x$  zwischen 0 und 2:

```
with(DEtools:
a1 := diff(y(x),x$4)+2*diff(y(x),x$2) -cos(x)=3;
DEplot(a1,[x,y],0..2,[[1,1,2,3,0]]);
```

Sie können ferner auf Pakete und Beispiele zurückgreifen, die weltweit von Maple Benutzern programmiert und in einer zentralen **Maple Share Bibliothek** gesammelt werden

## 11. Literaturhinweise

Folgende Bücher behandeln das Maple Computeralgebra System und Anwendungen:

1. *Auer, J. W.*  
**Linear Algebra with Applications**  
Prentice Hall Publishing
2. *Auer, J. W.*  
**Maple Solutions Manual for Linear Algebra with Applications**  
Prentice Hall Publishing
3. *Char, B. W. et al*  
**First Leaves: A Tutorial Introduction to Maple V**  
Springer Verlag
4. *Char, B. W. et al*  
**Maple V Language Reference Manual**  
Springer Verlag
5. *Char, B. W. et al*  
**Maple V Library Reference Manual**  
Springer Verlag
6. *Gloggengießer, H.*  
**Maple V Software für Mathematiker**  
Markt&Technik Verlag
7. *M. Kofler,*  
**Maple V Release 2, Einführung und Leitfaden für den Praktiker;**  
Addison-Wesley, 1994

Es gibt ferner in der **Maple Share Library** zahlreiche einführende Beschreibungen für **Maple**, teilweise im TeX Format (\*.tex), teilweise im Arbeitsblatt Format (\*.ms).

Ferner erscheint ca. 4 Mal im Jahr folgende Zeitschrift:

1. Maple Technical Newsletter  
Birkhäuser Publishing

In unregelmäßigen Abständen werden folgende Broschüren von **Waterloo Maple Software**, dem Entwicklungsteam von Maple, veröffentlicht:

1. Maple Technical Tips  
Waterloo Maple Software
2. The Maple Roots Report  
Waterloo Maple Software

## 11.1. Weitere Informationsquellen

- Maple Online Help System
- Maple Support
- Maple Share Bibliothek
- Maple E-Mail Liste
- News Group

### Maple Online Help System

Das Maple Online Help System enthält Hilfe-Informationen zum gesamten Funktionsumfang. Der Aufruf erfolgt über das Kommando `?<topic>` bzw. über den Menüpunkt *Help*.

### Maple Support

Sie können Fragen oder Probleme direkt an den Maple Support senden. Die E-Mail Adresse lautet:

```
support@maplesoft.on.ca
```

### Maple Share Bibliothek

Die Maple Share Bibliothek mit Beiträgen aus der Maple Benutzerschaft ist im RS/6000 Cluster des RZ verfügbar:

Computersystem:	titan.rz.Uni-Osnabrueck.DE
Verzeichnis:	usr/misc/maple-5.*/share*

Ferner liegt sie auf folgenden Computersystemen mit Anonymous FTP Server:

Internet	Internet address	Location	Directory
=====	=====	=====	=====
==			
129.97.140.58	daisy.uwaterloo.ca	Waterloo	maple
192.93.2.54	ftp.inria.fr	France	lang/maple
192.16.184.250	canb.can.nl	Holland	pub/maple-
ftplib			
129.132.101.33	neptune.inf.ethz.ch	Zurich	maple

### Maple E-Mail Liste

Die Maple E-Mail Liste dient zum weltweiten Informations- und Meinungsaustausch für Maple Benutzer. Senden Sie eine kurze E-Mail mit der Bitte um Aufnahme an folgende Adresse:

mail maple_group@daisy.uwaterloo.ca
Subject: I want to subscribe to the Maple E-Mail List.

### News Group

## Literaturhinweise

---

Folgende News Group beschäftigt sich allgemein mit symbolischer Mathematik:

<code>sci.math.symbolic</code>
--------------------------------