

---

# Maple-Kurzeinführung

---

Peter Harmand, November 2000

## Inhaltsverzeichnis

Die ersten Schritte . . . . .	1
Symbolische und numerische Rechnung, Variablen . . . . .	2
Termumformungen, Lösung von Gleichungen, Listen . . . . .	4
Funktionen . . . . .	8
Differential- und Integralrechnung . . . . .	10
Folgen und Reihen . . . . .	12
Differentialgleichungen . . . . .	13

**Was ist Maple?** Maple ist ein mächtiges Computeralgebraprogramm, dessen Stärke in symbolischen Rechnungen liegt: Termumformungen, Ableitungen, Integrale, Lösen von Gleichungen. Durch zahlreiche vordefinierte Befehle können viele mathematische Aufgaben mit einzeiligen Kommandos gelöst werden. Besonders bemerkenswert sind auch die Graphikfähigkeiten von Maple zur Veranschaulichung von Funktionen.

Die Benutzung von Maple ist kein Ersatz für das *Verständnis* mathematischen Sachverhalte. Bei numerisch sehr rechenintensiven Problemen oder statistische Datenauswertungen mit einer großen Anzahl von Meßwerten stößt man an die Grenzen des Programms.

**Verschiedene Versionen** Die aktuelle - und in dieser Kurzanleitung behandelte - Version von Maple ist V.5. Fast alle Bücher zu Maple gehen noch von den Versionen V.3 und V.4 aus. Das ist kein großes Problem, wenn man sich durch kleine Syntaxänderungen und neue Befehle nicht verwirren läßt.

## Die ersten Schritte

**Maple-Prompt, Semikolon** Nach dem Aufruf meldet sich Maple mit einem leeren Arbeitsblatt (worksheet) und signalisiert seine Eingabebereitschaft durch

> (der Maple-Prompt).

Hier gibt man einen Befehl ein und schließt ihn durch

; oder : (Abschluß eines Befehls)

ab. Dabei unterdrückt die zweite Variante die Ausgabe des Ergebnisses. Mit (Enter) wird der Befehl dann ausgeführt, bei Abschluß mit ; das Ergebnis angezeigt und eine neue Eingabezeile mit > angeboten.

**Korrektur** Haben Sie einen Fehler bei der Eingabe gemacht (z.B. einen Malpunkt oder einen Klammer vergessen) oder möchten Sie die letzte Eingabe mit anderen Werten noch einmal ausführen, können Sie mit dem Cursor auf die alte Eingabezeile hochgehen, die notwendigen Änderungen vornehmen und (Enter) drücken; der Cursor muß dabei nicht am Zeilenende stehen.

**cut and paste** Mit der Windows-üblichen Methode zum Kopieren von Textstellen über die Zwi-

schenablage (Markieren - <Ctrl>-C - Cursor zur neuen Stelle - <Ctrl>-V) können Sie alte Eingabeteile auf die neue Kommandozeile bringen.

**Leerstellen, Kommentare** Auf der Befehlszeile kann man zur besseren Lesbarkeit der Eingabe Leerzeichen verwenden. Text hinter einem Doppelkreuz # wird ignoriert und gibt so die Möglichkeit für Kommentare oder Erklärungen. (Die bessere Lösung - Maples Textbereich - wird hier nicht erklärt.)

**Hilfe** Wenn Sie etwas Zeit haben, arbeiten Sie die *New User's Tour* aus dem Hilfe-Menü durch. Mit ?*Befehl* erhalten Sie Hilfe zu den einzelnen Befehlen auf der Kommandozeile.

## Symbolische und numerische Rechnung, Variablen

Maple rechnet mit rationalen Zahlen exakt und kann mit beliebig großen Zahlen und mit unbestimmten Ausdrücken umgehen.

> 1/2 + 1/3;

$$\frac{5}{6}$$

> 40!;

815915283247897734345611269596115894272000000000

> 2\*x + 1 - (1/4)\*x;

$$\frac{7}{4}x + 1$$

**Arithmetische Operationen** Bei den Rechenoperationen wird die Division durch / und das Potenzieren durch ^ eingegeben. ACHTUNG: Obwohl Maple Produkte im Ausgabebereich als  $xy$  anzeigt, braucht es das Malzeichen \* in der Eingabe.

**Reihenfolge der Bearbeitung** Die üblichen Regeln, z.B. Punktrechnung geht vor Strichrechnung, werden befolgt. Beim Potenzieren sind zwingend Klammern erforderlich, d.h.  $x^{y^z}$  als  $x^{(y^z)}$  und  $(x^y)^z$  als  $(x^y)^z$  eingeben. Mathematische Funktionen werden vor Potenzen ausgeführt;  $\sin(x)^2$  steht damit für  $\sin^2 x = (\sin x)^2$ . Im Zweifelsfall setze man lieber zusätzliche Klammern.

**Vordefinierte Funktionen** Alle gängigen Funktionen sind unter den üblichen Namen verfügbar; erwähnenswert sind nur  $\text{abs}(x)$  für  $|x|$  und  $\text{sqrt}(x)$  für  $\sqrt{x}$ .

**Pi, I, infinity** Maple kennt die symbolischen Konstanten Pi, I und infinity für  $\pi$ ,  $i$  (imaginäre Einheit) und  $\infty$ .

**Dezimalzahlen, evalf** Maple unterscheidet klar zwischen rationalen Zahlen, symbolischen Ausdrücken und exakter Arithmetik auf der eine Seite und Fließkommazahlen und approximativen Verfahren auf der anderen Seite. Sobald eine der beteiligten Zahlen eine Dezimalpunkt enthält, wird die Fließkommaarithmetik verwendet; so sind z.B. 0.5 und 1/2 oder 2.0 und 2 verschiedene Dinge.

> 0.5 + 1/3; # Vergleiche das erste Beispiel

.8333333333

> sqrt(2); # sqrt steht für square root

$$\sqrt{2}$$

> sqrt(2.0);

1.414213562

```
> evalf(Pi);
```

3.141592653

Im letzten Beispiel wurde `evalf(Ausdruck)` zur numerischen Auswertung eines exakten Ausdrucks verwendet.

**Variablen** In Maple werden Variablen mit

*Variablenname* := *Ausdruck*

definiert. Dabei muß der Variablenname mit einem Buchstaben beginnen und kann danach aus Buchstaben, Ziffern und Unterstrichen bestehen. ACHTUNG: Groß- und Kleinschreibung werden unterschieden.

Der Ausdruck, der einer Variablen zugewiesen werden kann, ist fast beliebig: Zahlen, algebraische Terme, Gleichungen, Funktionen, Folgen usw. sind möglich. Namen, denen bisher nichts zugewiesen ist, werden beim Aufruf so wieder ausgegeben; ist ein Name durch eine Zuweisung belegt, so wird dessen Bedeutung angezeigt. (Bei dieser Regel gibt es Ausnahmen: so werden z.B. Funktionsdefinitionen so nicht angezeigt.)

```
> a := 5; # Zahl
```

*a* := 5

```
> gl := x^2 - 2*x + 1 = tan(x); # Gleichung
```

*gl* :=  $x^2 - 2x + 1 = \tan(x)$

```
> poly := x^5 - 3*x + 4; # algebraischer Ausdruck
```

*poly* :=  $x^5 - 3x + 4$

```
> A; # Nicht belegte Variablen werden einfach zurückgegeben
```

*A*

```
> poly; # Bei belegten Namen wird die Bedeutung angezeigt
```

$x^5 - 3x + 4$

```
> x := 0;
```

*x* := 0

```
> poly;
```

4

In der letzten Zeile sieht man, daß definierte Namen in andere Variablen eingesetzt werden.

**Löschen von Variablen** Das Löschen oder Freigeben einer Variablen geht mit

*Variablenname* := '*Variablenname*' .

Mittels `restart`; löscht man alle bisher definierten Variablen.

```
> x := 'x': # Löschen von x
```

```
> x; # Kontrolle
```

*x*

```
> poly; # Wieder die allgemeine Definition
```

$x^5 - 3x + 4$

Auswertung von Ausdrücken Mit `eval(Ausdruck, x = a)` kann man einen *Ausdruck* an der Stelle  $x = a$  auswerten. Eine Alternative ist die unten beschriebene Definition als Funktion.

> `eval(poly, x=1);`

2

Komplexe Zahlen Verwendet man *I* für die imaginäre Einheit, so rechnet Maple problemlos mit komplexen Zahlen. Auch die Funktionen `Re`, `Im`, `abs` und `conjugate` für Realteil, Imaginärteil, Betrag und konjugiert komplexe Zahl stehen zur Verfügung.

> `(1 + 2*I) * (3 - 4*I);`

11 + 2I

> `Im(1 + 2*I);`

2

> `conjugate(1 + 2*I);`

1 - 2I

Bei der Verwendung von Unbestimmten in Ausdrücken wie  $z := a + I*b$  sind *a* und *b* für Maple *nicht* automatisch reell. Diese zusätzliche Annahme erreicht man mit `evalc(evaluate complex)`.

> `z := a + b*I;`

> `Re(z); # Maple läßt a,b in C zu`

$\text{Re}(a) - \text{Im}(b)$

> `evalc(Re(z));`

*a*

Ergebnisspeicher, Ditto Beim interaktiven Arbeiten mit Maple ist es sehr hilfreich, mit `%` auf das letzte Ergebnis zurückgreifen zu können. Es gibt auch `%%` und `%%%` für das vorletzte und das vorvorletzte Ergebnis. ACHTUNG: Bis Version V.4 muß `"` statt `%` verwendet werden.

> `a + b;`

$a + b$

> `%/2;`

$\frac{1}{2}a + \frac{1}{2}b$

## Termumformungen, Lösung von Gleichungen, Listen

Termumformungen Die Manipulation mathematischer Ausdrücke ist eines der natürlichen Einsatzgebiete von Computeralgebrasystemen. Leider ist das gelegentlich nicht ganz leicht - abgesehen davon, daß absolut nicht klar ist, was eine „einfache“ Form eines Ausdrucks ist. Die wesentlichen Befehle sind:

<code>simplify(Ausdruck)</code>	Maples <i>Versuch</i> zu vereinfachen
<code>expand(Ausdruck)</code>	Ausmultiplizieren von Produkten
<code>factor(Ausdruck)</code>	Ausfaktorisieren von gemeinsamen Termen
<code>collect(Ausdruck, x)</code>	Zusammenfassen der Koeffizienten von Potenzen von <i>x</i>
<code>sort(Ausdruck, x)</code>	Sortieren nach Potenzen von <i>x</i>

Faktorisieren von Polynomen bedeutet Nullstellenbestimmung – das ist bei einem Grad  $\geq 5$  im allgemeinen nicht möglich und funktioniert auch sonst nur im Reellen.

> `expand((x+y)^3);`

$$x^3 + 3x^2y + 3xy^2 + y^3$$

> `a^2 - b^2;`

$$a^2 - b^2$$

> `% = factor(%);`

$$a^2 - b^2 = (a - b)(a + b)$$

> `simplify(sin(x)^2 + cos(x)^2);`

$$1$$

> `a/b + c/d: % = simplify(%);`

$$\frac{a}{b} + \frac{c}{d} = \frac{ad + cb}{bd}$$

> `p1 := x^4 + x^3 - 2*x^2:`

> `p2 := a*x^2 + x - 3*a:`

> `p1*p2;`

$$(x^4 + x^3 - 2x^2)(ax^2 + x - 3a)$$

> `P := expand(%);`

$$P := x^6a + x^5 - 5x^4a + x^5a + x^4 - 3x^3a - 2x^3 + 6ax^2$$

> `collect(P,x); # Potenzen von x zusammenfassen`

$$x^6a + 6x^2a + (a+1)x^5 + (-5a+1)x^4 + (-3a-2)x^3$$

> `sort(%,x); # Nach Potenzen von x sortieren`

$$ax^6 + (a+1)x^5 + (-5a+1)x^4 + (-3a-2)x^3 + 6ax^2$$

Lösen von Gleichungen

Der Befehl für das symbolische Lösen von Gleichungen ist:

`solve(Gleichung, Variable)`

Die Angabe der Variablen, nach der aufgelöst werden soll, kann entfallen, wenn die Gleichung nur eine Unbestimmte enthält. Auch komplexe Lösungen und mehrfache Nullstellen werden gefunden. Maple benutzt bei komplizierten Ausdrücken die %1-Notation für mehrfach auftretende Teilterme und antwortet mit *RootOf(..)*, wenn es nichts herausfindet.

> `solve(x^2 + p*x + q = 0, x);`

$$-\frac{1}{2}p + \frac{1}{2}\sqrt{p^2 - 4q}, -\frac{1}{2}p - \frac{1}{2}\sqrt{p^2 - 4q}$$

> `solve(x^3 - 3*x^2 + x - 3 = 0);`

$$3, I, -I$$

> `# Hier findet Maple in der Standardeinstellung nur eine Lösung`

> `solve(sin(x) = 0);`

$$0$$

> `# Hier findet Maple auch die triviale Lösung x=0 nicht`

> `solve(x = tan(x));`

$$\text{RootOf}(\tan(\_Z) - \_Z)$$

Systeme von Gleichungen, Ungleichungen Mit dem `solve`-Befehl lassen sich auch Systeme von Gleichungen lösen, wenn man die Gleichungen und die Variablen in Mengenklammern schreibt.

```
> solve({2*x - y = 10, x + a*y = b}, {x,y});
      {y = 2 * (b - 5) / (1 + 2*a), x = (10*a + b) / (1 + 2*a)}
> solve({28000 = b*500^c, 175 = b*0.5^c}, {b,c});
      {b = 291.2103401, c = .7347066609}
> solve(1-s^2 >= 1+s);
      RealRange(-1, 0)
```

An dem letzten Beispiel sieht man, daß auch einfache Ungleichungen mit `solve` gelöst werden können; die Antwort ist Maples Schreibweise für das Intervall  $[-1, 0]$ .

Numerische Lösung von Gleichungen

Mit dem Befehl

`fsolve(Gleichung, Variable, Bereich)`

steht ein leistungsfähiges Kommando zur numerischen Lösung von Gleichungen zur Verfügung. Die Angabe des Bereichs, in dem eine Lösung gesucht werden soll, kann zwar entfallen, ist aber angeraten. ACHTUNG: Bei komplizierten Gleichungen ist es *sehr* empfehlenswert, sich vorher mit einem oder mehreren Plots über die Struktur des Problems und die Lage der Lösungen zu informieren. Nur eine intelligente Kombination von Plots, eigenen Überlegungen und `fsolve` wird bei komplizierten Gleichungen alle Lösungen liefern; siehe *Maple zu Kapitel 3*.

```
> fsolve(exp(x) = 1/x, x);
      .5671432904
> fsolve(tan(x) = x, x, Pi..(3/2)*Pi);
      4.493409458
```

Folgen und Listen

Eine endliche Folge ist für Maple eine durch Kommata getrennte Aufzählung von Objekten. Mit einer in eckigen Klammern `[ ]` eingeschlossenen Folge vereinbart man eine Liste.

Der Zugriff auf einzelne Elemente oder Bereiche in Folgen oder Listen ist über die Angabe des Index(bereichs) in eckigen Klammern möglich; das erste Element hat den Index 1.

```
> folge := 1,2,2,a,x^2;
      folge := 1, 2, 2, a, x^2
> liste := [3, b=c, sin];
      liste := [ 3, b = c, sin ]
> liste[2]; folge[1..3];
      b = c
      1, 2, 2
> solve(x^2 - 7*x + 12 = 0); # Bei mehreren Lösungen einer Gleichung
> # ist das Ergebnis eine Folge
      4, 3
> lsg := %: lsg[1]; lsg[2];
      4
```

**Vergrößern von Folgen und Listen** Folgen können durch Voranstellen oder Anhängen von Ausdrücken vergrößert werden. Mit `op` kann man eine Liste in eine Folge umwandeln (und dann erweitern). Die Anzahl der Elemente einer Liste bestimmt man mit `nops` (number of operands).

```
> folgeneu := 0, folge;
```

$$\text{folgeneu} := 0, 1, 2, 2, a, x^2$$

```
> op(liste);
```

$$3, b = c, \sin$$

```
> liste := [op(liste), 25]; # Verlängern einer Liste
```

$$\text{liste} := [3, b = c, \sin, 25]$$

```
> nops(liste);
```

4

**Der seq-Befehl** Eine sehr leistungsfähige und bequeme Möglichkeit zur Erzeugung von Folgen bietet

$$\text{seq}(\text{Ausdruck\_in\_}k, k=n..m)$$

Dabei werden in *Ausdruck\_in\_k* für  $k$  nacheinander die Werte  $n, n+1, n+2, \dots, m$  eingesetzt.

```
> seq(n^2, n=1..6);
```

$$1, 4, 9, 16, 25, 36$$

```
> seq([k, (-1)^k/k], k=1..5); # Erzeugung von Listen für plot
```

$$\left[1, -1\right], \left[2, \frac{1}{2}\right], \left[3, -\frac{1}{3}\right], \left[4, \frac{1}{4}\right], \left[5, -\frac{1}{5}\right]$$

**Schöne Ausgabe** Will man z.B. bei der Lösung von Gleichungen eine schön formatierte Ausgabe des Ergebnisses, so kann man dies ``x`` = erreichen. Für typographisch aufwendigere Operationen bietet Maple bei einigen Befehlen sog. träge Versionen an, die nur das entsprechende Symbol ausgeben, aber nichts berechnen; z.B. ist `Limit` die träge Version von `limit`, `Int` die von `int`. Zur Ausgabe von mehreren Variablen *nebeneinander* kann man `print(Var_1, Var_2, ...)` benutzen.

```
> `x` = solve(5*x - 1 = a, x);
```

$$x = \frac{1}{5} + \frac{1}{5}a$$

```
> p := 1: q := 2: print(p,q,r);
```

$$1, 2, r$$

```
> int(x^2, x=0..1); # Ein bestimmtes Integral; siehe unten
```

$$\frac{1}{3}$$

```
> Int(x^2, x=0..1); # Darstellung und keine Rechnung
```

$$\int_0^1 x^2 dx$$

```
> Int(x^2, x=0..1) = int(x^2, x=0..1); # Die Kombination liefert
> # eine schöne Ausgabe
```

$$\int_0^1 x^2 dx = \frac{1}{3}$$

## Funktionen

Definition von Funktionen Mit der Syntax

```
f := x -> Ausdruck_in_x
```

definiert man Funktionen  $f : \mathbb{R} \rightarrow \mathbb{R}$ . (Der Pfeil wird mit einem Minus- und einem Größerzeichen eingegeben.)

```
> f := x -> x^2 + cos(x);
```

$$f := x \rightarrow x^2 + \cos(x)$$

```
> `f(0)` = f(0); f(sqrt(y));
```

$$f(0) = 1$$

$$y + \cos(\sqrt{y})$$

```
> f; # Für die Anzeige der Definition reicht f allein nicht
```

$$f$$

```
> eval(f); # .. man braucht eval( )
```

$$x \rightarrow x^2 + \cos(x)$$

Zur Definition von stückweise definierten Funktionen siehe *Maple zu Kapitel 2*.

Umwandlung in Funktion Mit `unapply(A, x)` kann man einen Ausdruck  $A$  in eine Funktion von  $x$  umwandeln.

```
> C := a*x^2 + b*sin(x):
```

```
> f := unapply(C, x); # Ausdruck zu Funktion machen
```

$$f := x \rightarrow a x^2 + b \sin(x)$$

Plots Der Befehl zum Zeichnen des Graphen einer Funktion  $f$  im Intervall  $[a, b]$  ist

```
plot(f(x), x = a..b).
```

Der `plot`-Befehl erlaubt eine Vielzahl von zusätzlichen Optionen, die unten zum Teil erklärt werden. Graphiken in worksheets sind anklickbar und dann durch Ziehen am Rahmen skalierbar sowie über die Zwischenablage in andere Programme, z.B. Word, kopierbar. Bei angeklickten Graphiken erhält man eine zusätzliche Symbolleiste mit der verschiedene Optionen, z.B. die Darstellung der Achsen oder die Maßstäbe auf den beiden Achsen, einstellbar sind.

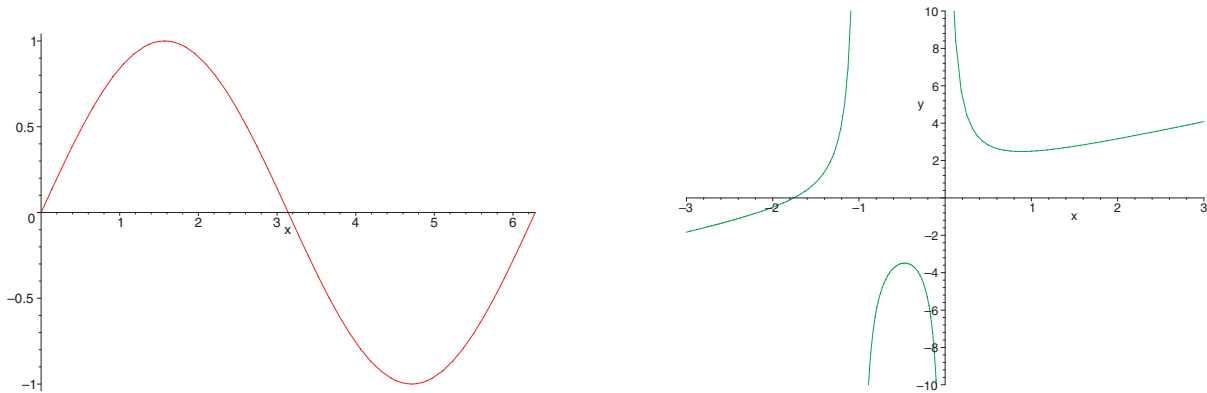
```
> plot(sin(x), x=0..2*Pi); # Ausdruck unten links
```

```
> g := x -> x + 1 + 1/(x*(x+1));
```

$$g := x \rightarrow x + 1 + \frac{1}{x(x+1)}$$



```
> # Festlegung des Bereichs auf der y-Achse
> # Bei Unstetigkeiten den Graph nicht verbinden
> plot(g(x), x=-3..3, y=-10..10, discont=true);
```



### Optionen des Plot-Befehls

Die folgenden Optionen können im `plot`-Befehl verwendet werden:

<code>y = c..d</code>	Festlegung des Bereichs auf der $y$ -Achse
<code>discont=true</code>	Unterdrückung von senkrechten Linien bei Unstetigkeiten
<code>numpoints=n</code>	feinere Unterteilung für genaueren Plot
<code>scaling=constrained</code>	gleiche Maßstäbe auf beiden Achsen
<code>color=Farbe</code>	$Farbe \in \{\text{red, green, blue, yellow, ..}\}$ siehe <code>?plot, color</code> für eine vollständige Liste
<code>linestyle=n</code>	Strichelung, $n \in \{1, 2, 3, 4, 5\}$
<code>thickness=n</code>	Strichdicke, $n \in \{0, 1, 2, 3\}$

### Mehrere Plots

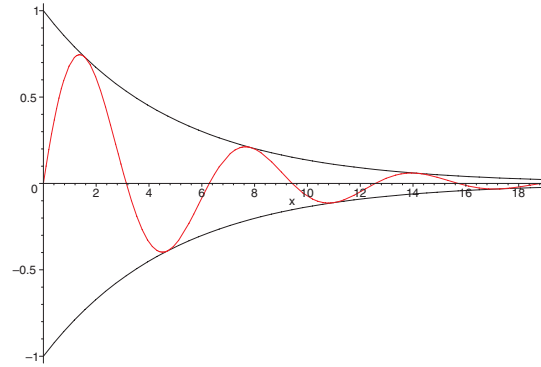
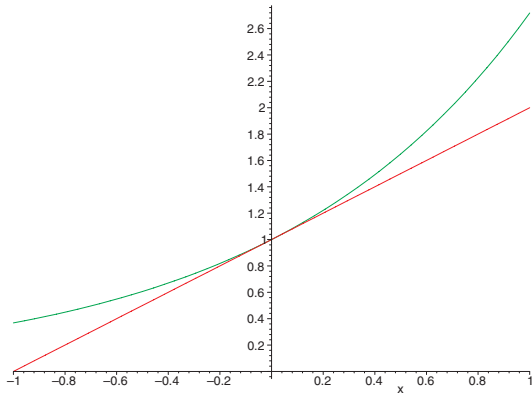
Die einfachste Möglichkeit ist `plot({f(x), g(x)}, ...)`. Für komplexere Überlagerungen benutze man

```
P1 := plot(...): P2 := plot(...):
with(plots): # Braucht man nur einmal pro worksheet
display([P1, P2]);
```

Die Plots werden zunächst als  $P1$  und  $P2$  gespeichert; beachten Sie die Doppelpunkte. Mit `with(...)` lädt man zusätzliche Befehle aus externen Bibliotheken. Der `display`-Befehl aus der `plots`-Bibliothek stellt dann  $P1$  und  $P2$  in einem Bild dar. Das geht natürlich auch für mehr als zwei Plots.

```
> plot({exp(x), 1 + x}, x=-1..1);          # 2 Graphen in einem Plot

> P1 := plot(exp(-0.2*x)*sin(x), x=0..6*Pi, color=red, thickness=2):
> P2 := plot({exp(-.2*x), -exp(-.2*x)}, x=0..6*Pi, color=black):
> with(plots): display([P1, P2]);
```



Für die Möglichkeiten zur Beschriftung von Plots siehe *Maple zu Kapitel 2*.

## Differential- und Integralrechnung

**Grenzwerte** Den Grenzwert einer Funktion  $f$  für  $x \rightarrow x_0$  bestimmt man mit

$$\text{limit}(f(x), x = x_0).$$

Für  $x_0$  ist auch *infinity* zulässig. Die zusätzlichen Optionen *left* und *right* erlauben die Berechnung einseitiger Grenzwerte.

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

*undefined*

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

$\infty$

> # Mit träger Version für schöne Ausgabe

> `Limit(exp(x), x=-infinity) = limit(exp(x), x=-infinity);`

$$\lim_{x \rightarrow (-\infty)} e^x = 0$$

**Ableitung** Die Befehle zum Differenzieren einer Funktion sind

$$\text{diff}(f(x), x) \quad \text{oder} \quad D(f).$$

Dabei ist die erste Version für Ausdrücke evtl. intuitiver als der D-Operator für Funktionen. Bei dem trägen Kommando `Diff` benutzt Maple die Leibnizsche Notation mit einem runden »d«.

> `Diff(x^2, x) = diff(x^2, x);`

$$\frac{\partial}{\partial x} x^2 = 2x$$

> `D(tan);`

$$1 + \tan^2$$

> `diff((f*g)(x), x);` # Die Produktregel mit `diff`

$$\left( \frac{\partial}{\partial x} f(x) \right) g(x) + f(x) \left( \frac{\partial}{\partial x} g(x) \right)$$

Höhere Ableitungen Zur Berechnung der  $n$ -ten Ableitung braucht man `diff` oder `D` nicht mehrfach anwenden, sondern kann dies mit `diff(f(x), x$n)` oder `(D@@n)(f)` bewerkstelligen.

```
> diff(exp(sin(x^3)), x$3);
```

$$-27 \cos(x^3) x^6 e^{\sin(x^3)} - 54 \sin(x^3) x^3 e^{\sin(x^3)} - 81 \sin(x^3) x^6 \cos(x^3) e^{\sin(x^3)} + 6 \cos(x^3) e^{\sin(x^3)} + 54 \cos(x^3)^2 x^3 e^{\sin(x^3)} + 27 \cos(x^3)^3 x^6 e^{\sin(x^3)}$$

Taylor-Entwicklung Das Taylorpolynom der Ordnung  $n$  von  $f(x)$  für den Entwicklungspunkt  $x_0$  berechnet man mit

```
taylor(f(x), x=x0, n).
```

Dabei ist die Ordnung (normalerweise) der Grad +1. Zum Entfernen des O-Terms benutze man `convert(Ausdruck, polynom)`.

```
> t2 := taylor(cos(x), x=0, 3);
```

$$t2 := 1 - \frac{1}{2} x^2 + O(x^3)$$

```
> p2 := convert(t2, polynom);
```

$$p2 := 1 - \frac{1}{2} x^2$$

Integration Maples Befehle zur Bestimmung von bestimmten und unbestimmten Integralen sind

```
int(f(x), x = a..b)    und    int(f(x), x).
```

Mit der ersten Version können auch uneigentliche Integrale berechnet werden.

```
> int(x^2 + 9, x=1..3); # Ein einfaches bestimmtes Integral
```

$$\frac{80}{3}$$

```
> # Beispiel 3.41 (1) mit träger Version
```

```
> Int(x*cos(x), x) = int(x*cos(x), x);
```

$$\int x \cos(x) dx = \cos(x) + x \sin(x)$$

```
> Int(exp(-x), x=0..infinity) = int(exp(-x), x=0..infinity);
```

$$\int_0^{\infty} e^{-x} dx = 1$$

```
> # Die Rückgabe der Eingabe im folgenden Beispiel bedeutet,
```

```
> # daß Maple das Integral nicht exakt auswerten kann.
```

```
> int(cos(x^5), x=0..1);
```

$$\int_0^1 \cos(x^5) dx$$

```
> evalf(%);    # numerische Berechnung
```

$$.9564853782$$

## Folgen und Reihen

**Definition, Grenzwerte** Folgen behandelt Maple wie Funktionen. Definition und Grenzwerte funktionieren für (explizit definierte) Folgen wie oben beschrieben. Für rekursiv definierte Folgen siehe *Maple zu Kapitel 2*.

```
> Limit(1/n, n=infinity) = limit(1/n, n=infinity);
```

$$\lim_{n \rightarrow \infty} \frac{1}{n} = 0$$

```
> b := n -> (n^2 + 3*n)/(2*n^2 + 1); # Beispiel 1.41 (1)
```

$$b := n \rightarrow \frac{n^2 + 3n}{2n^2 + 1}$$

```
> limit(b(n), n=infinity);
```

$$\frac{1}{2}$$

```
> limit((-1)^n, n=infinity);
```

-1..1

Die letzte Ausgabe bedeutet, daß Maple keinen Grenzwert findet, aber feststellen kann, daß alle Häufungspunkte im Intervall  $[-1, 1]$  liegen.

**Graphische Darstellung von Folgen** Zunächst muß man mit einem geeigneten seq-Befehl eine Liste der Form  $L = [[x_1, y_1], [x_2, y_2], \dots, [x_n, y_n]]$  erzeugen. Die Darstellung der Folge (als Punktmenge) geht dann mit

```
plot(L, style=point).
```

Ohne die Option `style=point` werden die Punkte durch Geradenstücke verbunden. Zusätzlich kann man mit

```
symbol=box oder symbol=circle oder symbol=diamond
```

die Darstellung der „Punkte“ beeinflussen.

```
> A := seq([n, (1 + 1/n)^n], n=1..50):
```

```
> plot([A], style=point); # Einzelne Punkte
```

```
> b := n -> (-1)^n/n;
```

$$b := n \rightarrow \frac{(-1)^n}{n}$$

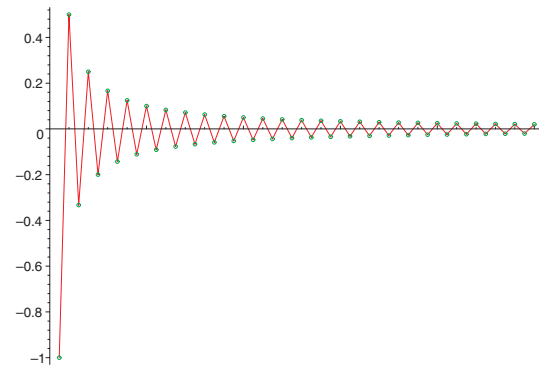
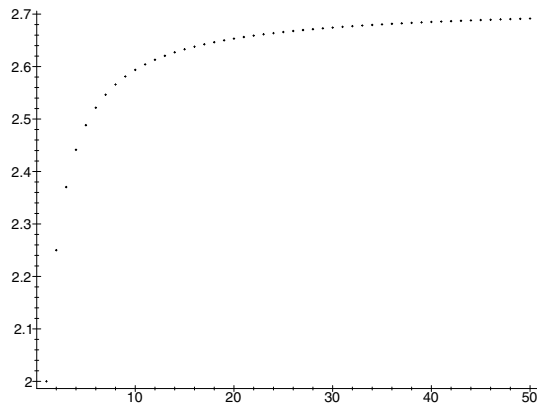
```
> L := seq([n, b(n)], n=1..50):
```

```
> # Die einzelnen Punkte als grüne Kreise
```

```
> P1 := plot([L], style=point, symbol=circle, color=green):
```

```
> P2 := plot([L]): # Streckenzug
```

```
> display([P1, P2]);
```



Reihen Endliche und unendliche Reihen bearbeitet man mit

$$\text{sum}(a_k, k = 1..n) \quad \text{oder} \quad \text{sum}(a_k, k = 1..infinity).$$

Bei unendlichen Reihen mit Unbestimmten, insb. bei Potenzreihen, achtet Maple leider nicht auf die Konvergenz!

> `sum(q^k, k=0..n);` # 1.27

$$\frac{q^{(n+1)}}{q-1} - \frac{1}{q-1}$$

> # Beispiel 1.50 mit träger Version

> `Sum(1/n^2, n=1..infinity) = sum(1/n^2, n=1..infinity);`

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{1}{6} \pi^2$$

> # Die geometrische Reihe (Beispiel 1.44)

> # Die Reihe ist nur für  $\text{abs}(q) < 1$  konvergent;

> # die Darstellung ist auch etwas ungewöhnlich.

> `sum(q^n, n=0..infinity);`

$$-\frac{1}{q-1}$$

## Differentialgleichungen

Lösung von Differentialgleichungen Der Befehl zur Lösung von Differentialgleichungen ist `dsolve(Dgl)` bzw. `dsolve({Dgl, AnfWert}, y(t))`.

Die erste Variante liefert die allgemeine Lösung mit freien Parametern `_C1, ...`; die zweite Form ist für die Bearbeitung von Anfangswertproblemen da. In der Differentialgleichung `Dgl` kann man die `diff-` oder die `D-`Notation für die Ableitung benutzen. Bei den Anfangswerten muß man `D` verwenden.

> ## Inhomogene lineare Dgl mit konstanten Koeffizienten

> `dgl := diff(y(t), t) = a*y(t) + b;`

$$dgl := \frac{\partial}{\partial t} y(t) = a y(t) + b$$

> `dsolve(dgl);`

$$y(t) = -\frac{b}{a} + e^{(at)} \_C1$$

```
> ## Logistische Dgl mit Anfangsbedingung
> dgl := diff(y(t),t) = a*y(t) - b*y(t)^2;
      dgl :=  $\frac{\partial}{\partial t} y(t) = a y(t) - b y(t)^2$ 
```

```
> dsolve({dgl, y(0)=2}, y(t));
```

$$y(t) = \frac{a}{b + \frac{1}{2} e^{(-at)} (a - 2b)}$$

```
> # Extrahieren der Lösung
> # rhs liefert die rechte Seite (right hand side) der obigen Gleichung
> f := unapply(rhs(%),t);
```

$$f := t \rightarrow \frac{a}{b + \frac{1}{2} e^{(-at)} (a - 2b)}$$

```
> f(0);
```

2

```
> ## Eine Schwingungsgleichung
```

```
> Sgl := (D@@2)(y)(t) = -2*D(y)(t) - 2*y(t);
```

$$Sgl := (D^{(2)})(y)(t) = -2D(y)(t) - 2y(t)$$

```
> AnfBed := y(0) = 5, D(y)(0) = 1;
```

```
> dsolve({Sgl,AnfBed}, y(t));
```

$$y(t) = 5 e^{(-t)} \cos(t) + 6 e^{(-t)} \sin(t)$$

Systeme von  
Differential-  
gleichungen

Mit der Version

`dsolve({Sys}, {Var})` bzw. `dsolve({Sys,AnfWerte}, {Var})`

kann Maple auch die allgemeine Lösung bzw. die Lösung eines Anfangswertproblems für ein System von (einfachen) Differentialgleichungen bestimmen.

```
> ## Ein lineares System
```

```
> Sys := D(u)(t) = 3*u(t) - 2*v(t), D(v)(t) = 2*u(t) - 2*v(t);
```

$$Sys := D(u)(t) = 3u(t) - 2v(t), D(v)(t) = 2u(t) - 2v(t)$$

```
> # Lösung eines Anfangswertproblems für dieses lineare System
```

```
> dsolve({Sys, u(0)=8, v(0)=15}, {u(t), v(t)});
```

$$\{u(t) = \frac{2}{3} e^{(2t)} + \frac{22}{3} e^{(-t)}, v(t) = \frac{44}{3} e^{(-t)} + \frac{1}{3} e^{(2t)}\}$$

Richtungsfelder, Lösungskurven Ein Richtungsfeld für eine Differentialgleichung der Form  $y' = f(t, y)$  zusammen mit einigen Lösungskurven erzeugt man mit

```
with(DEtools): # Braucht man nur einmal pro worksheet
DEplot(Dgl, Var, t-Bereich, AnfWerte, y-Bereich, Optionen) .
```

Dabei ist

*Dgl* die Differentialgleichung  
*Var* der Name der abhängigen Variablen (häufig also  $y$ )  
*t-Bereich* der Bereich der unabhängigen Variablen (häufig also  $t=a..b$ )  
*AnfWerte* eine Liste von Anfangswerten in der Form  $[[y(t_0) = y_0], [y(t_1) = y_1], \dots]$   
*y-Bereich* der Bereich der abhängigen Variablen (häufig also  $y=c..d$ )

Bei den Optionen muß man `arrows=line` verwenden. Außerdem ist `thickness=1` und `linecolor=Farbe` angeraten, da man sonst dicke gelbe Graphen bekommt. Wenn die Lösungskurven zu eckig aussehen, muß man noch `stepsize` setzen; die Standardeinstellung ist hierbei  $(b - a)/20$ .

Will man nur ein Richtungsfeld, so gibt man keine Anfangswerte an. Mit der Option `arrows=none`, erhält man nur die Lösungskurve(n).

```
> ## Richtungsfeld mit Lösungskurven
> dg12 := diff(y(t), t) = y(t) - t^2;
```

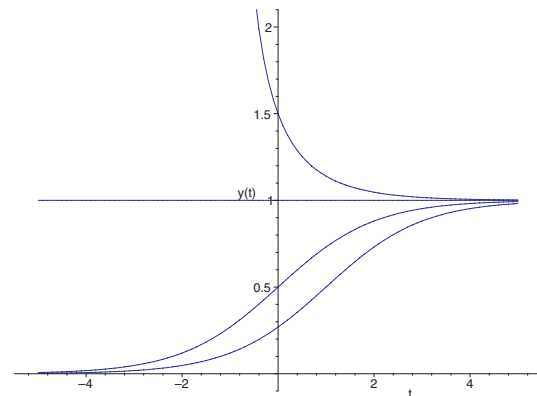
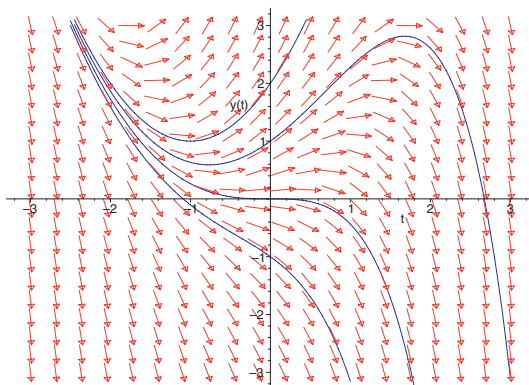
$$dg12 := \frac{\partial}{\partial t} y(t) = y(t) - t^2$$

```
> AnfBed := [[y(0)=-1], [y(0)=0], [y(0)=1], [y(0)=2]]:
> Optionen := arrows=line, linecolor=blue, thickness=1, stepsize=0.1:
> DEplot(dg12, y, t=-3..3, AnfBed, y=-3..3, Optionen);
```

```
> ## Lösungskurven für die logistische Differentialgleichung
> Dg11 := diff(y(t), t) = y(t)*(1 - y(t));
```

$$Dg11 := \frac{\partial}{\partial t} y(t) = y(t) (1 - y(t))$$

```
> AnfWerte := [[0,.5], [1,.5], [0,1], [0,1.5]]:
> Optionen := arrows=none, linecolor=blue, thickness=1, stepsize=0.1:
> DEplot(Dg11, y, t=-5..5, AnfWerte, y=0..2, Optionen);
```



Phasenraum, Trajektorien Für ein autonomes System der Form  $u' = f(u, v)$ ,  $v' = g(u, v)$  erzeugt man mit

```
DEplot(Sys, Var, t-Bereich, AnfWerte, u-Bereich, v-Bereich, Optionen)
```

einen Vektorfeld-Plot des Phasenraums zusammen mit einigen Trajektorien. Dabei ist

*Sys* das System in der Form  $[Dgl1, Dgl2]$   
*Var* eine Liste der Namen der abhängigen Variablen (häufig  $[u, v]$ )  
*AnfWerte* eine Liste von Anfangswerten in der Form  
 $[[u(t_0) = u_0, v(t_0) = v_0], [u(t_1) = u_1, v(t_1) = v_1], \dots]$

Man sollte `arrows=medium` für Pfeile wählen. Die anderen Parameter und Optionen sind wie oben.

```
> ## Vektorfeld mit Trajektorien (Phasenraum, d.h. u-v-Ebene)
> sys := diff(u(t),t) = -u(t)/2 - v(t), diff(v(t),t) = u(t) - v(t)/2;
      sys :=  $\frac{\partial}{\partial t} u(t) = -\frac{1}{2} u(t) - v(t), \frac{\partial}{\partial t} v(t) = u(t) - \frac{1}{2} v(t)$ 
> AnfWerte := [[u(0)=15,v(0)=15], [u(0)=0,v(0)=-15]]:
> Optionen := arrows=medium, linecolor=blue, thickness=1, stepsize=0.05:
> # Gute Werte für t= .. und stepsize= erfordern etwas Ausprobieren
> DEplot([sys], [u,v], t=0..10, AnfWerte, u=-17..17, v=-17..17, Optionen);
```

*(t, u)- und (t, v)-Darstellung* Statt des Phasenraums, d.h. der  $(u, v)$ -Darstellung, erreicht man mit einer geeigneten Wahl (siehe Beispiel) der `scene`-Option einen Plot der Lösungen als Funktion der unabhängigen Variablen.

```
> ## Graph der Lösungen als Funktion von der unabhängigen Variablen t
> AnfWert := [[u(0)=15,v(0)=15]]:
> Opt := thickness=1, stepsize=0.05:
> Pu := DEplot([sys], [u,v], t=0..10, AnfWert, u=-17..17, v=-17..17,
>   Opt, linecolor=blue, scene=[t,u]): # u in blau
> Pv := DEplot([sys], [u,v], t=0..10, AnfWert, u=-17..17, v=-17..17,
>   Opt, linecolor=green, scene=[t,v]): # v in grün
> display([Pu,Pv], view=[0..10,-8..17], labels=["t","u,v"]);
```

