

# Computer und Software 1

Maple

Grundkonzepte, Graphische Darstellung von Funktionen



# Organisatorisches

- Computer und Software 1 wird **nicht benotet (Studienleistung)**. Die Modulbeschreibung ist falsch, es gilt die Bachelorordnung !
- Es gibt eine **Klausur am Computer**:
  - Es müssen 2-3 kleinere Aufgaben mit Maple am Computer gelöst werden.
  - Je nach Teilnehmerzahl wird es mehrere Prüfungsgruppen geben (max. 20 Studenten pro Gruppe).

**Prüfungstermin: noch festzulegen, je nach Teilnehmerzahl**

# Computer-Algebra-Systeme

- **Gut geeignet für**
  - Manipulation von mathematischen Ausdrücken
  - Analytisches Lösen von Gleichungen, Ungleichungen, Gleichungssystemen
  - Analytisches Rechnen mit Vektoren, Matrizen
  - Grenzwertberechnung (Reihen, Folgen, mathematische Ausdrücke)
  - Differenzieren, Auswertung von (nicht zu komplizierten) unbestimmten Integralen
  - Lösen von einfachen Differentialgleichungen bzw. Differentialgleichungssystemen
  - Visualisierung diverser Funktionen/Lösungen
  - Rechnen mit beliebiger Genauigkeit
  - Schritt für Schritt Dokumentation einer Berechnung

# Computer-Algebra-Systeme

## Weniger geeignet für:

- Numerisches Rechnen
- Verarbeitung und Analyse von größeren Datenmengen
- Erstellung komplexer (mathematischer) Programme
- Erzeugung von publikationsreifen Abbildungen

# Maple-Alternativen

- Alternativen
  - Mathematica (kommerziell)
  - Maxima, <http://maxima.sourceforge.net>
  - Yacas (relativ neu)

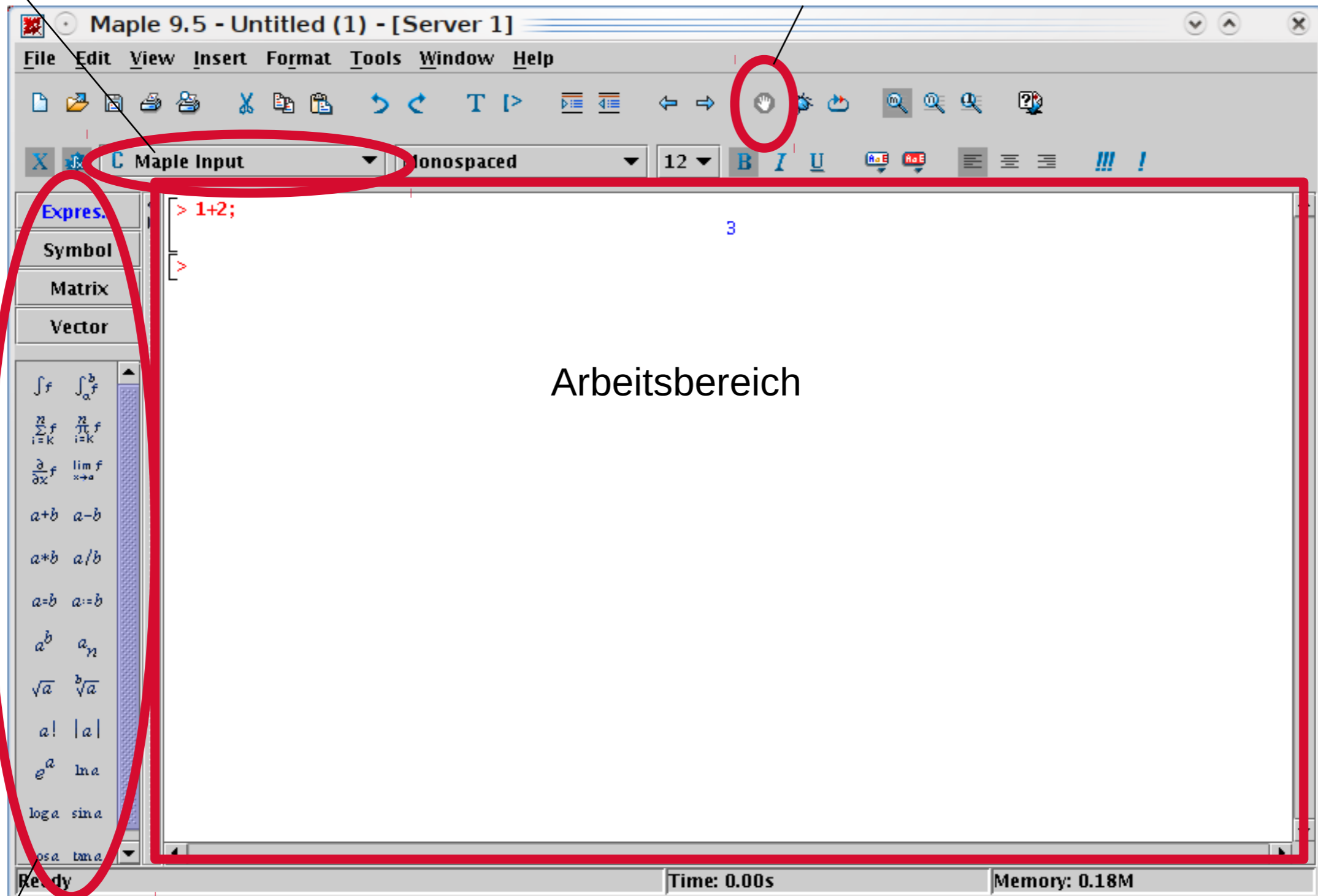
# Maple-Dokumentation

- Vorlesungsfolien
- Online Help in Maple
- Maple-Webseite (z.B. [Maple for Physics Students](#))
- Frank Y. Wang, Physics with MAPLE  
(breites Spektrum an fortgeschrittenen physikalischen Problemen, keine  
Programmdokumentation an sich)

# Startbildschirm

Aktueller Inputtyp

Abbrechen einer Berechnung



Formatierungshilfen

# Maple als Taschenrechner

Zurücksetzen

```
> restart;  
> 2*(3 + 5)/4;  
> 2^100;  
> sqrt(179);  
> evalf(%);  
> 4/5 + 8 /15;  
> evalf(%);
```

Jeder Befehl wird mit “;” abgeschlossen

```
> Digits := 30;  
> evalf(4/3);  
> sqrt(-4);  
> Pi;  
> evalf(Pi, 20);  
> exp(Pi);  
> evalf(%);
```

Anzahl der Stellen

Komplexe  
Einheitswurzel

Anzahl der Stellen, wenn  
abweichend von Digits

Numerische  
Auswertung  
erzwingen



# Arbeitsbereich

Einheiten können zu- und aufgeklappt werden

## [-] Fingerübungen mit MAPLE zum Aufwärmen

### [-] Triviale Berechnungen

Hier sollten ein Paar triviale Berechnungen aus didaktischen Gründen demonstriert werden.

Eine triviale Berechnung:

```
> 2*3 + 4/2;
```

8

Berechnung des Integrals

$\int \sin(x)^2 dx$  mit MAPLE

```
> int(sin(x)^2, x);
```

$-\frac{1}{2} \sin(x) \cos(x) + \frac{1}{2} x$

```
>
```

Kommentare/Dokumentation

### [-] Weitere triviale Berechnungen

```
> exp(-50);
```

$e^{(-50)}$

MAPLE-Projekte sollten immer **ausreichend** Dokumentiert werden!

Gliederung  
(Kapitel,  
Unterkapitel, etc.)

Maples Antwort

Eingabe

## Wiederverwendung von Ergebnissen

> 4/3;

$$\frac{4}{3}$$

Verwendung eines Ergebnisses mit '%'

> % \* 2;

$$\frac{8}{3}$$

Verwendung des vorletzten Ergebnisses mit '%%'

> %%^2;

$$\frac{16}{9}$$

Verwendung des vorvorletzten Ergebnisses mit '%%%'

> %%% - 1;

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

**Achtung!**

**Letztes Ergebnis =  
Ergebnis der zuletzt  
ausgeführten**

**Berechnung**, die nicht  
unbedingt die davor  
stehende Berechnung  
ist!

- Verwendung von '%' sollte eher vermieden werden
- Wichtige Berechnungen sollten in Variablen gespeichert und über die Variablennamen wiederverwendet werden.

## Auswertung von Ausdrücken

- Maple versucht die Auswertung in Dezimalformat (Genauigkeitsverlust), soweit wie möglich zu vermeiden:

`> 1/3 - 3/2;`

$$\frac{-7}{6}$$

`> sqrt(Pi);`

$$\sqrt{\pi}$$

`> sqrt(42);`

$$\sqrt{42}$$

- Explizite Auswertung kann über `evalf()` erfolgen bzw. erzwungen werden.
- Operationen mit Dezimalzahlen werden automatisch ausgewertet:

```
> evalf(sqrt(Pi), 4); 1.773
                        Anzahl der Stellen explizit beim Aufruf
> Digits := 15;      Digits := 15
                        Anzahl der Stellen global setzen
> evalf(sqrt(Pi));
1.77245385090552
> sqrt(42.0);
6.48074069840786
```

## Eingabe

- Eingabe von Text oberhalb des Prompts:  
Strg-Shift-K bzw. Insert|Paragraph|Before cursor
- Eingabe von Text unterhalb des Prompts:  
Strg-Shift-J bzw. Insert|Paragraph|After cursor
- Eingabe von mathematischen Formel während Texteingabe:  
Strg-R

- Mathematische Zeichen können links angeklickt werden. Einzelne Positionen ("?",) werden via TAB selektiert und ausgefüllt werden.

The screenshot shows the Maple 9.5 interface. The main window title is "Maple 9.5 - /local/home/aradi/MapleIntro.mw - [Server 1]". The menu bar includes File, Edit, View, Insert, Format, Tools, Window, and Help. The toolbar contains various icons for file operations and editing. The main input area contains the text `> restart;` in red. To the left, the "Expressions" palette is open, showing various mathematical symbols like integrals, sums, limits, and algebraic expressions. A red circle highlights the palette, with the label "Eingabehilfen" below it. Another red circle highlights the input field, with the label "Eingabe für ausgewähltes Feld" to its right. A third red circle highlights a question mark in the input field, with the label "Ausgewähltes Feld" below it. The text "Texteingabe mit Integral:  $\int ? d?$ " is positioned between the input field and the palette.

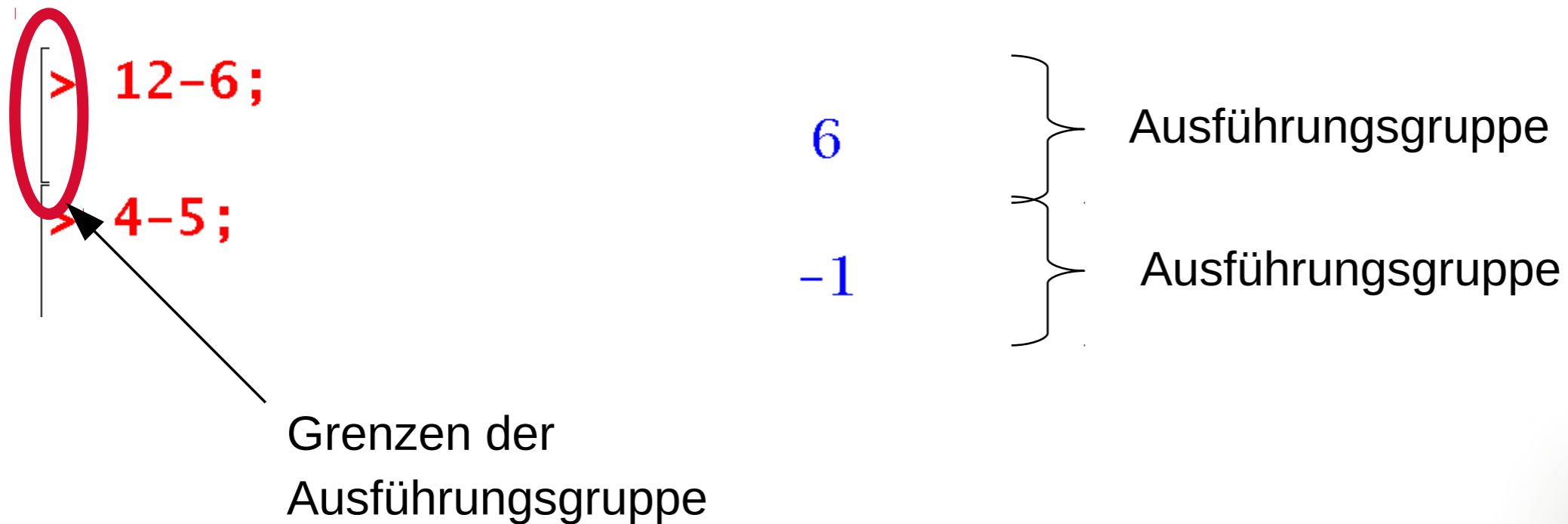
- Alternativ können Formeln in Maple-Notation eingegeben werden:

Textbeispiel:  $\int_a^b \sin(x)^2 dx$

`[Strg-R]int(sin(x)^2,  
x=a..b[Enter]`

# Ausführungsgruppen

Ausführungsgruppe =  
Eingabe und (wenn ausgeführt)  
Ergebnis  
und evtl. Dokumentation



- Benachbarte Ausführungsgruppen werden mit **F4 (Edit | Split or Join | Join Exec. Group)** zusammengeführt (nachstehende mit aktueller vereint)
- Ergebnisse erst wenn alle Befehle der Ausführungsgruppe ausgeführt:

```
> 12-6;
> 4-5;
6
-1
```

- Alternativ: Zeilen mit **[Shift-Enter]** (statt **[Enter]**) abschließen, dann kann eine weitere Zeile eingegeben werden:

```
> 12-6; ← [Shift-Enter]
4-5; ← [Enter]
6
-1
```

- Vereinte Gruppen können mit **F3 (Edit | Split or Join | Split Exec. Group)** getrennt werden.

## Unterdrückung des Ergebnisses / Hilfe

- Mit **Doppelpunkt** kann die Darstellung des Ergebnisses **unterdrückt** werden:

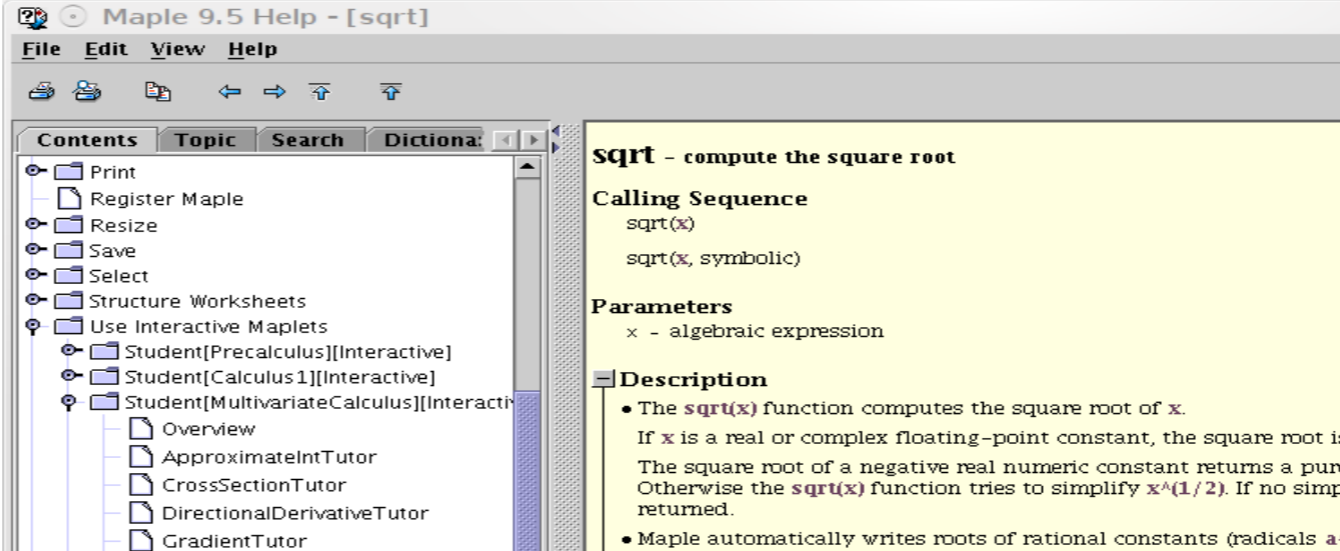
```
> a := 5:
> a;
```

5

← Ergebnis unterdrückt  
(Operation trotzdem ausgeführt!)

- **Hilfe** zu einem Befehl kann mit vorangestelltem **Fragezeichen** angezeigt werden

```
> ?sqrt
```



The screenshot shows the Maple 9.5 Help window for the `sqrt` function. The window title is "Maple 9.5 Help - [sqrt]". The main content area displays the following information:

- sqrt - compute the square root**
- Calling Sequence**
  - `sqrt(x)`
  - `sqrt(x, symbolic)`
- Parameters**
  - `x` - algebraic expression
- Description**
  - The `sqrt(x)` function computes the square root of `x`.
  - If `x` is a real or complex floating-point constant, the square root is computed numerically.
  - The square root of a negative real numeric constant returns a purely imaginary value.
  - Otherwise the `sqrt(x)` function tries to simplify  $x^{1/2}$ . If no simplification is possible, the function returns `sqrt(x)`.
  - Maple automatically writes roots of rational constants (radicals) as `a^(1/n)`.

- Weitere Hilfe über das Help-Menü



## Wichtigste Eingabebefehle

Ctrl-J	Einfügen einer neuen Befehlsgruppe nach der aktuellen Gruppe
Ctrl-K	Einfügen einer neuen Befehlsgruppe vor der aktuellen Gruppe
F4	Zusammenführen zweier Befehlsgruppen
F3	Befehlsgruppe aufteilen
Shift-Ctrl-J	Einfügen eines (Text-)Kommentars vor dem Cursor
Shift-Ctrl-K	Einfügen eines (Text-)Kommentars nach dem Cursor
Ctrl-Del	Ausgewählte Gruppe/Kommentar/Eingabe/Ausgabe löschen.
Insert   Section	Neues Kapitel anlegen
Insert   Subsection	Neues Unterkapitel anlegen
Ctrl-R	Wechseln zur Math.mode während der Texteingabe
Ctrl-T	Wechseln zur Textmode während mathematischer Eingabe

# Konstanten

Wichtigste Konstanten:

<b>Pi</b>	3.14....	<b>e</b>	Eulersche Zahl
<b>I</b>	Komplexe Einheitswurzel	<b>infinity</b>	Unendlich

- Konstanten (und Maple im Allg.) sind gegen **Groß- und Kleinschreibung** empfindlich:

```

> cos(pi);
cos(π)
> evalf(cos(pi));
cos(π)
> cos(Pi);
-1

```

- Komplexe Zahlen werden als **Realteil + I \* Imaginärteil** eingegeben:

```

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

```

# Variablen

- Variablen sind **Namen**, die ein Objekt/Ausdruck repräsentieren
- Der **Wert** einer Variable (im Gegensatz zu einer Konstante) **läßt sich verändern**

- Variablen werden mit **':='** zugewiesen  
(**Bitten nicht mit '=' verwechseln!**)
- **Variable** steht **links**, **Wert** steht **rechts**

**OK**

```
> a := 12;  
> a;  
12
```

```
a := 12  
12
```

**Falsch**

```
> b = 4;  
> b;
```

```
b = 4  
b
```

# Variablentypen

Matrix:

```
> a := Matrix([[1, 2], [3, 4]]);
```

$$a := \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

Sequenz:

```
> a := 1, 2, 3;
```

$$a := 1, 2, 3$$

Liste:

```
> a := [ 1, 2, 3 ];
```

$$a := [1, 2, 3]$$

Menge:

```
> a := { 1, 2, 3 };
```

$$a := \{1, 2, 3\}$$

- Typ einer Variable kann mit der Funktion **whattype** abgefragt werden:

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

$$a := 3$$

```
> whattype(a);
```

*integer*

```
> a := x - y = 12;
```

$$a := x - y = 12$$

```
> whattype(a);
```

=

# Variablenzuweisung rückgängig machen

## restart

(alle Variablenzuweisungen)

```
> a := 12:
  b := 3:
> a; b;
12
3

> restart;
> a; b;
a
b
```

## explizite Zuweisung

(nur ausgewählte Zuweisungen)

```
> a := 12:
  b := 3:
> a; b;
12
3

> a := 'a';
a := a

> a; b;
a
3
```

Gute Praxis: **restart** als ersten Befehl eines Maple-Arbeitsblattes

## Sequenz (Folge)

- Sequenz: Eine Abfolge von Objekten (Typ willkürlich)

```
> a := 1, 2, 3;
a := 1, 2, 3
```

- Wird eine Sequenzvariable verwendet, so hat das den selben Effekt, als hätte man die einzelnen Einträge explizit hingeschrieben:

Berechnung des Restes bei der Teilung von 3 durch 2 (mit Hilfe der Funktion modp):

```
> modp(3, 2);
1
> a := 3, 2;
a := 3, 2
> modp(a);
1
```

- Eine Sequenz nach fester Regel kann mit der Funktion **seq** erzeugt werden

```
> seq(x^n, n=1..10);
x, x2, x3, x4, x5, x6, x7, x8, x9, x10
```

## Liste

- Eine festgelegte Abfolge von Elementen (Typ willkürlich)

```
> a := [ 1, 2, 3 ];
      a := [1, 2, 3]
> b := [ x, x^2, x^3 ];
      b := [x, x^2, x^3]
```

- Maple behält die Reihenfolge, wie sie im Input eingegeben wurde.
- Ein **Element kann mehrmals vorkommen**

```
> c := [ 1, 4, -1, 1 ];
      c := [1, 4, -1, 1]
```

- Die Elemente der Liste können auch einzeln manipuliert werden:

```
> c[1];
      1
> c[1] := 9;
      c1 := 9
> c;
      [9, 4, -1, 1]
> c[2] * c[3];
      -4
```

**Erstes Element wird mit 1 indiziert!**

## Menge

- Eine Menge von Objekten
- Reihenfolge nicht relevant, und deswegen nicht zwingend beibehalten!

```
> d := { 2, -1, 8};  
d := {-1, 2, 8}
```

- Jedes Element kann nur einmal vorkommen (Duplikate ausgefiltert)

```
> f := { -1, 4, 3, -1, 2 };  
f := {-1, 2, 3, 4}
```

- Auf die einzelnen Elemente kann ähnlich wie bei Listen zugegriffen werden:  
(Der Index muss der von Maple aufgestellten Reihenfolge entsprechen!)

```
> f;  
{-1, 2, 3, 4}  
  
> f[1];  
-1
```

- Die einzelnen Elemente können jedoch nicht verändert werden!

```
> f[1] := 12;  
Error, cannot reassign the entries in a set
```



## Zusammenhang zw. Menge, Sequenz, Liste

- Sowohl Liste (`[]`) als auch Menge (`{}`) sind **Operatoren**, die die in ihnen übergebene **Operanden** zu einer Liste bzw. Menge **verknüpfen**.

`{ 1, 2, 3 }`: Operator: `{}` Operanden: 1, 2, 3 Ergebnis: `{1, 2, 3}` (Menge)  
`[ 1, 2, 3 ]`: Operator: `[]` Operanden: 1, 2, 3 Ergebnis: `[ 1, 2, 3 ]` (Liste)

- Die Anzahl der Operanden lässt sich via **nops** ermitteln:

<pre>&gt; f := [1, 2, 3]; &gt; nops(f);</pre>	<pre>f := [1, 2, 3] 3</pre>	<pre>&gt; g := { -1, 4, 3, 6 }; &gt; nops(g);</pre>	<pre>g := {-1, 3, 4, 6} 4</pre>
---	-----------------------------	---	---------------------------------

- Die Operanden können via **op** als eine Sequenz abgefragt werden:

<pre>&gt; op(f);</pre>	<pre>1, 2, 3</pre>	<pre>&gt; op(g);</pre>	<pre>-1, 3, 4, 6</pre>
------------------------	--------------------	------------------------	------------------------

## Konversion zwischen Menge und Liste

- Über eine Sequenz lassen sich Listen in Mengen konvertieren und zurück  
Beispiel: Anzahl der Elemente in einer Liste **ohne Duplikate** bestimmen:

```
> f := [ -1, 4, 3, -1 ];  
           f := [-1, 4, 3, -1]  
> tmp := op(f);  
           tmp := -1, 4, 3, -1  
> g := { tmp };  
           g := {-1, 3, 4}  
> nops(f);  
           4  
> nops(g);  
           3
```

## Funktionen

- Die meisten mathematischen Funktionen und Operationen, die man während des Physikstudiums kennen lernt, sind in Maple implementiert.

Trig. Funktionen und Inverse	sin, cos, tan, asin, acos, atan
Hyper. trig. Funktionen:	sinh, cosh, tanh, asinh, acosh, atanh
Absolutwert:	abs
Quadratwurzel:	sqrt
Exponentialfunktion	exp
Logarithmus (natürlicher, 10):	log, log10
Runden, Aufrunden, Abrunden :	round, floor, ceil
Fakultät	factorial oder ! (z.B: 4!)
Besselfunktionen	Bessel, BesselJ, BesselK, BesselY
:	

- Bei der Suche nach einer Funktion, einfach im [Maple-Help](#) nachschauen.

## Summe / Produkt

- Summen- und Produktoperatoren sind auch implementiert:

```
> sum(1/x^i, i=1..3);
```

$$\frac{1}{x} + \frac{1}{x^2} + \frac{1}{x^3}$$

```
> product(1/x^i, i=1..3);
```

$$\frac{1}{x^6}$$

## Benutzerdefinierte Funktionen

- Man kann auch selber Funktionen im Maple definieren.
- Benutzerdefinierte Funktionen werden mit dem “->” (zwei Zeichen, “-” und “>”) Operator definiert:

```
> myFunc := x -> sin(x) * cos(x);
      myFunc := x → sin(x) cos(x)
> myFunc(Pi/3);
       $\frac{1}{4} \sqrt{3}$ 
```

- Bei Funktionen mit mehreren Variablen müssen die Variablen vor dem “->” Operator in Klammern stehen:

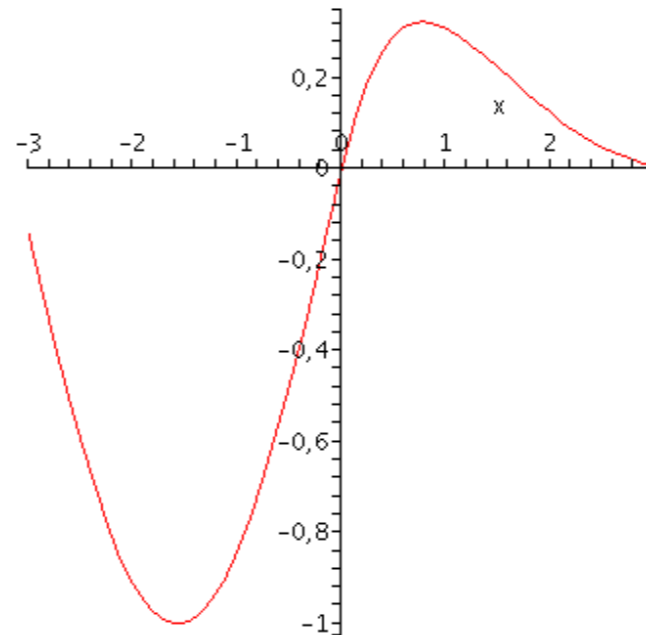
```
> testFunc2d := (x,y) -> sin(x) * cos(y);
      testFunc2d := (x, y) → sin(x) cos(y)
> testFunc2d(Pi/3, Pi/4);
       $\frac{1}{4} \sqrt{3} \sqrt{2}$ 
```

- Funktionen können auch stückweise definiert werden mit der **piecewise** Funktion.

```
> pw := x -> piecewise(x < 0, sin(x), x >= 0,
    sin(x) * exp(-x));
```

```
    pw := x → piecewise(x < 0, sin(x), 0 ≤ x, sin(x) e(-x))
```

```
> plot(pw(x), x=-3..3);
```

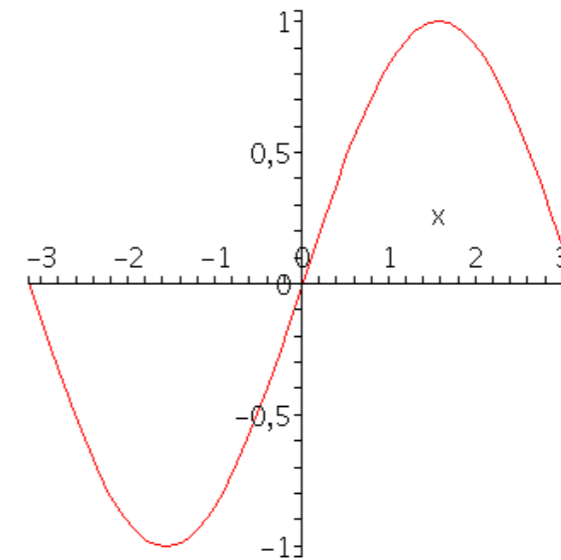


## Zweidimensionale Graphiken

- Funktionen mit einer Variable können mit Hilfe der **plot**-Funktion dargestellt werden:

```
> plot(sin(x), x=-Pi..Pi);
```

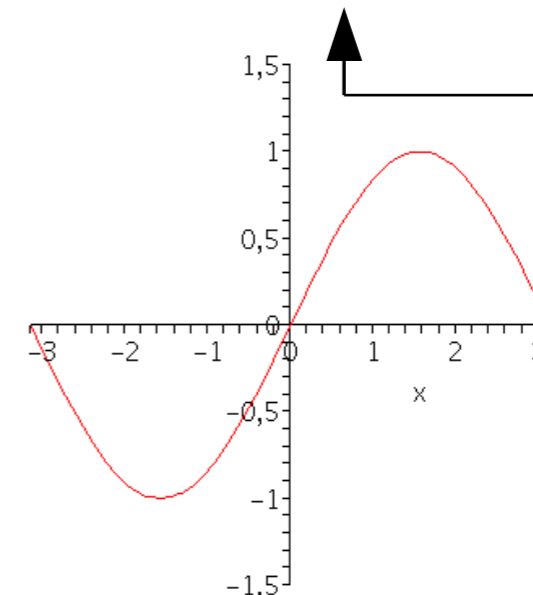
Ausdruck zu plotten



Bereich der freien Variable im Plot

- y-Bereich wird automatisch eingestellt, kann jedoch auch **manuell** eingestellt werden:

```
> plot(sin(x), x=-Pi..Pi, -1.5..1.5);
```



y-Bereich

- Es können verschiedene Optionen angegeben werden, um die Ausgabe zu beeinflussen:

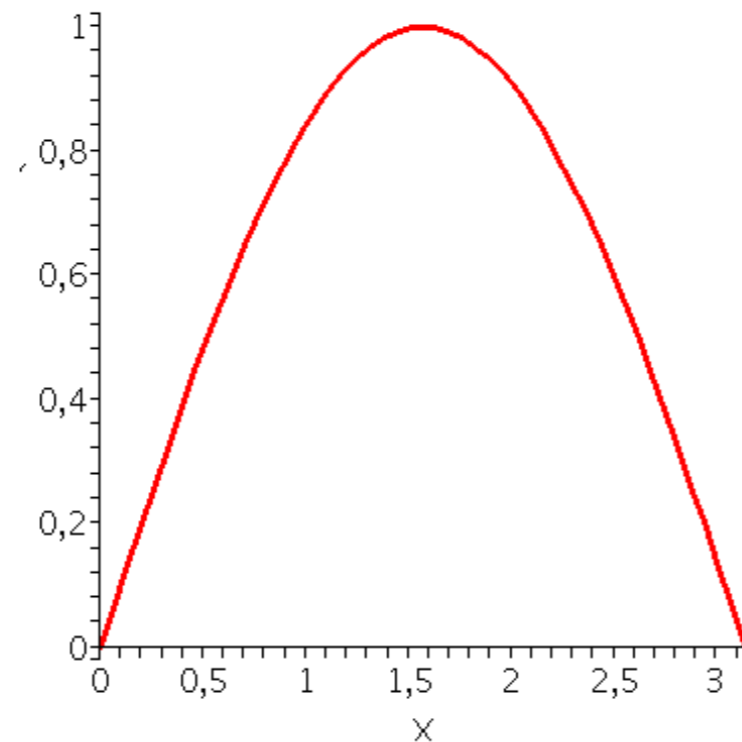
<code>coords=polar</code>	Polarkoordinaten statt kartesischen
<code>discont=true</code>	Unstetigkeit beachten
<code>scaling=constrained</code>	Gleiche Skalierung für x-, und y-Achsen
<code>style=point</code>	Punkte statt Kurve
<code>symbol=diamond</code>	Form der Punkte (auch circle, box, cross, ...)
<code>color=black</code>	Farbe der Kurve/Punkte
<code>linestyle=1</code>	Linientyp (durchgezogen, gestrichelt, ...)
<code>numpoints=1000</code>	Anzahl der Stützpunkte in der Graphik
<code>thickness=2</code>	Liniendicke
<code>title=Name</code>	Titel der Abbildung
<code>labels=[NameX, NameY]</code>	Achsenbeschriftung

- Für weitere Optionen, siehe [Maple-Hilfe](#)



- Die Optionen werden *nach der Spezifikation* des X-Bereiches (und evtl. des Y-Bereiches) in der Form **Option=Wert** aufgezählt:

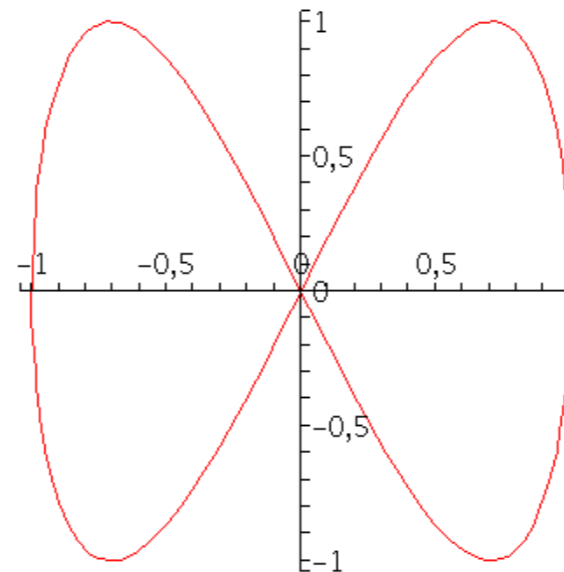
```
> plot(sin(t), t=0..Pi, thickness=2, labels=["X", "Y"]);
```



## Parametrischer Plot

- Parametrische Plots: Statt einer Funktion wird eine Liste [  $x(t)$ ,  $y(t)$ ,  $t=a..b$  ] mit jeweils getrennten Funktionen für X- und Y-Koordinate übergeben.
- Beide Funktionen müssen von einer Variable abhängen.
- Laufbereich der Variable muss auch in der Liste übergeben werden.

```
> plot([sin(t), cos(2*t+Pi/2), t=-Pi..Pi]);
```

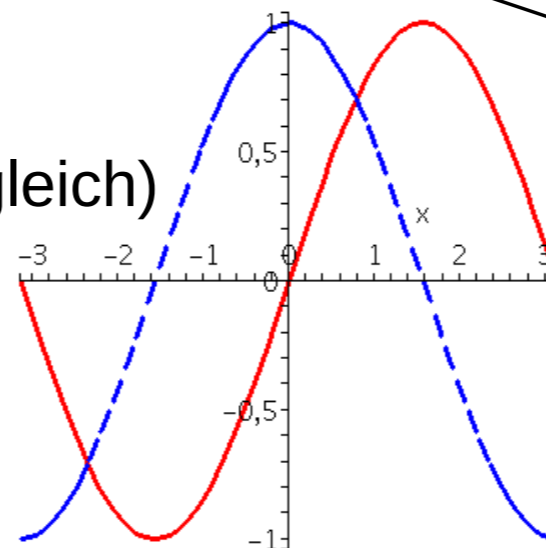


## Darstellung mehrerer Funktionen

- Sollten mehrere Funktionen geplottet werden, müssen sie in einer Liste übergeben werden
- Bei Plot-Optionen, die sich auf die Darstellung nur einer Kurve beziehen, kann eine entsprechende Anzahl von Werten in Form einer Liste übergeben werden

```
> plot([sin(x), cos(x) ], x=-Pi..Pi, color=[red, blue],  
      thickness=2, linestyle=[1,3]);
```

Global (für alle Kurven gleich)



Kurvenspezifisch

## Weitere Darstellungsarten

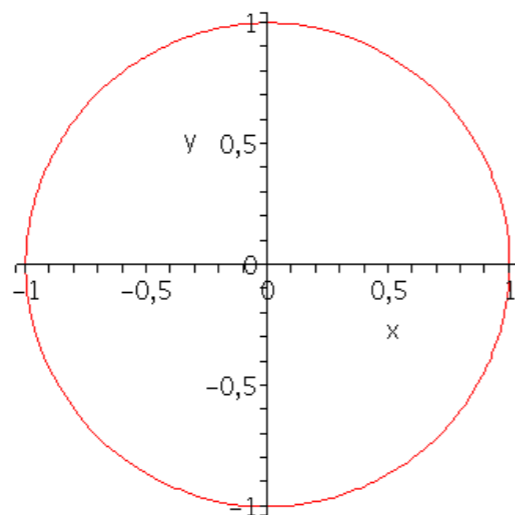
- Benötigen **erweiterte Plotbibliothek:**

```
> with(plots):
```

```
Warning, the name changecoords has been redefined
```

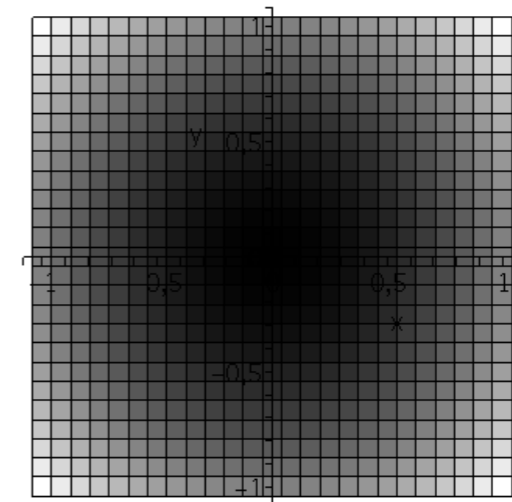
Implizite Funktion mit zwei Variablen  
**implicitplot**

```
> implicitplot(x^2+y^2=1,  
x=-1..1, y=-1..1);
```



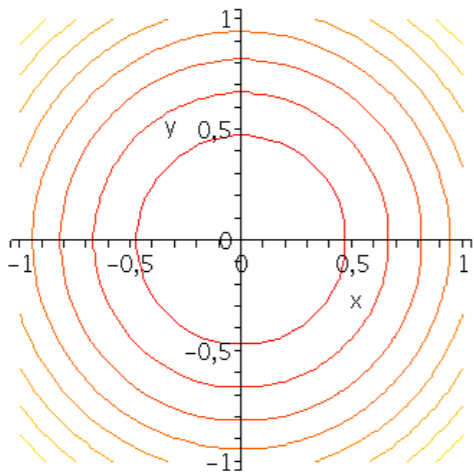
Dichtediagramm  
**densityplot**

```
> densityplot(x^2+y^2, x=-1..1,  
y=-1..1);
```



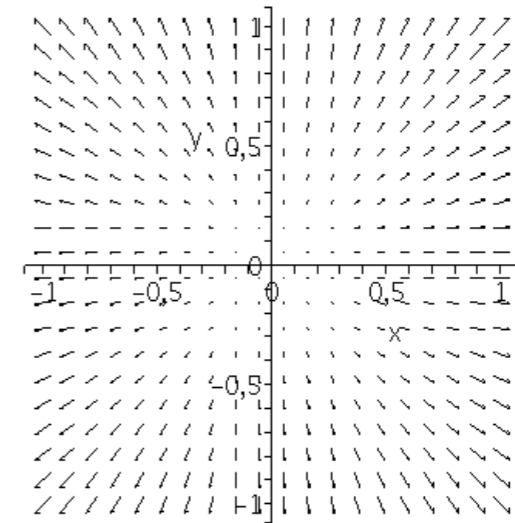
## Höhenlinien contourplot

```
> contourplot(x^2+y^2, x=-1..1,  
y=-1..1);
```



## Feldlinien fieldplot

```
> fieldplot([x, y], x=-1..1,  
y=-1..1);
```

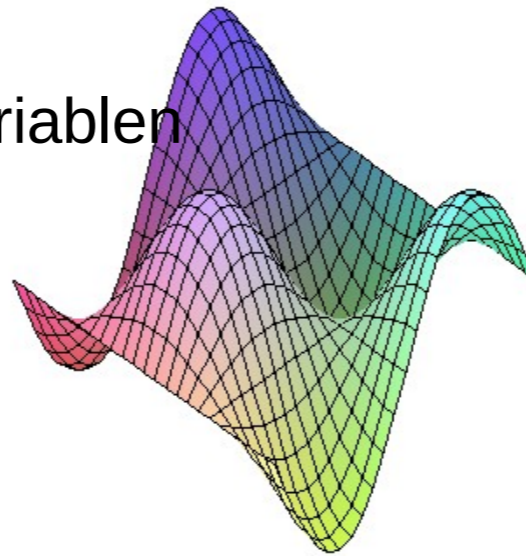


## 3D-Graphiken

- Befehl **plot3d**:

```
> plot3d(sin(x)*cos(y), x=0..2*Pi, y=0..2*Pi);
```

Ausdruck mit zwei Variablen



Bereich 1. Variable

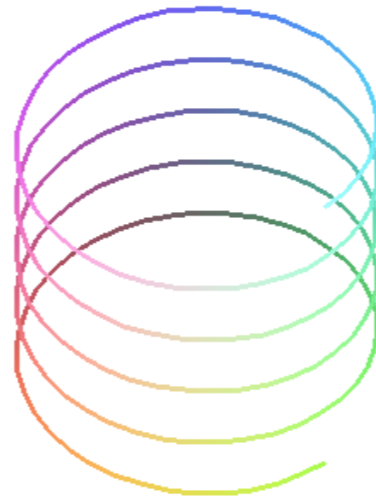
Bereich 2. Variable

- Analog zu plot, viele zusätzliche Optionen (siehe **?plot3d**)

## Parameterischer Plot in 3D

- Ähnlich zu 2D, kann man auch in 3D Kurven mit dem Befehl **spacecurve** als parametrische Abbildungen zeichnen:

```
> spacecurve([sin(t), cos(t), t], t=0..10*Pi,  
numpoints=200, thickness=2);
```

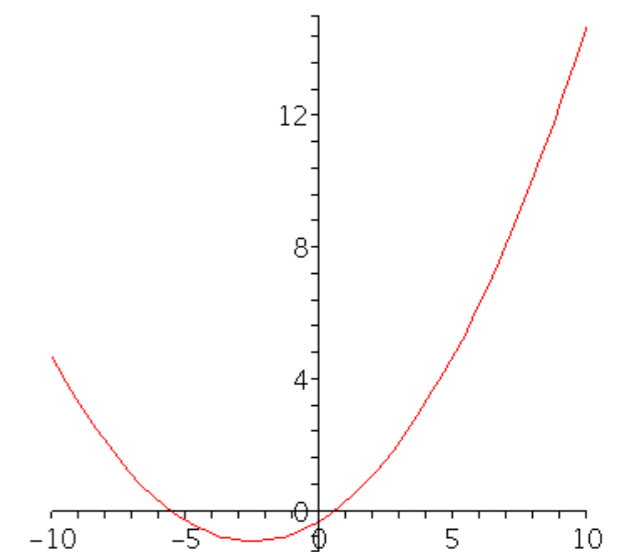


## Plotten von Funktionen mit Parametern

- Plot erwartet einen Ausdruck mit **einem einzigen Unbekannten**
- Wenn die Funktion mit unbekanntem Parametern definiert wurde, müssen diese vor dem Plotten gesetzt werden:

```
> restart;
=
> f := x -> a*x^2 + b*x + c;
      f := x -> a x2 + b x + c
=
> f(10);
      100 a + 10 b + c
=
> f(x);
      a x2 + b x + c
=
```

```
=
> a := 0.1: b := 0.5: c := -0.3:
=
> f(10);
      14.7
=
> f(x);
      0.1 x2 + 0.5 x - 0.3
=
> plot(f(x), x=-10..10);
```



**Nachteil:** Parameter behalten ihren Wert auch nach dem Plotten



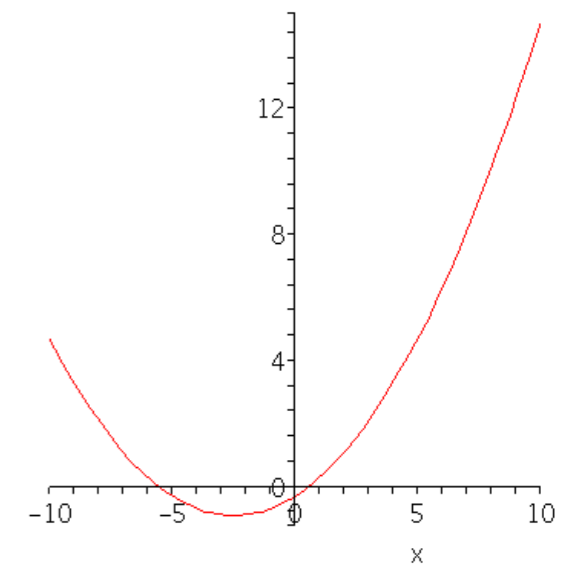
## Substitution via subs

- Mit der **subs**-Funktion können Parameter temporär substituiert werden.
- Es werden die angegebenen Werte nur in den angegebenen Ausdruck eingesetzt, aber keine Variablen mit entsprechenden Werten angelegt:

```
> restart;
> f := x -> a*x^2 + b*x + c;
      f := x -> a x2 + b x + c
> f(10);
      100 a + 10 b + c
> f(x);
      a x2 + b x + c
> tmp := subs(a=0.1, b=0.5, c=-0.3, f(x));
      tmp := 0.1 x2 + 0.5 x - 0.3
```

↑  
Variablensubstitution

```
> plot(tmp, x=-10..10);
```



```
> f(x);
```

$a x^2 + b x + c$

Vielen Dank für die Aufmerksamkeit!

Machen Sie bitte mit den Übungsaufgaben weiter!