

Das Handbuch zu KAlgebra

Aleix Pol

Übersetzung: Burkhard Lück



Das Handbuch zu KAlgebra

Inhaltsverzeichnis

1	Einführung	7
2	Syntax	9
3	Arbeiten mit dem Rechner	11
4	2D-Graphen	13
4.1	Syntax	13
4.2	Eigenschaften	13
5	3D-Graphen	15
6	Funktionen	17
7	Referenz der Funktionen für KAlgebra	18
7.1	plus	18
7.2	times	18
7.3	minus	18
7.4	divide	18
7.5	quotient	19
7.6	power	19
7.7	root	19
7.8	factorial	19
7.9	and	19
7.10	or	20
7.11	xor	20
7.12	not	20
7.13	gcd	20
7.14	lcm	20
7.15	rem	21
7.16	factorof	21
7.17	max	21
7.18	min	21
7.19	lt	21
7.20	gt	21

Das Handbuch zu KAlgebra

7.21 eq	22
7.22 neq	22
7.23 leq	22
7.24 geq	22
7.25 implies	22
7.26 approx	23
7.27 abs	23
7.28 floor	23
7.29 ceiling	23
7.30 sin	23
7.31 cos	24
7.32 tan	24
7.33 sec	24
7.34 csc	24
7.35 cot	24
7.36 sinh	24
7.37 cosh	25
7.38 tanh	25
7.39 sech	25
7.40 csch	25
7.41 coth	25
7.42 arcsin	25
7.43 arccos	26
7.44 arctan	26
7.45 arccot	26
7.46 arccosh	26
7.47 arccsc	26
7.48 arccsch	26
7.49 arcsec	27
7.50 arcsech	27
7.51 arcsinh	27
7.52 arctanh	27
7.53 exp	27
7.54 ln	28
7.55 log	28
7.56 conjugate	28
7.57 arg	28
7.58 real	28
7.59 imaginary	29
7.60 sum	29
7.61 product	29
7.62 diff	29
7.63 card	29
7.64 scalarproduct	30

Das Handbuch zu KAlgebra

7.65 selector	30
7.66 union	30
7.67 forall	30
7.68 exists	30
7.69 map	31
7.70 filter	31
7.71 transpose	31
8 Danksagungen und Lizenz	32

Zusammenfassung

KAlgebra ist eine Anwendung, die Ihren grafischen Taschenrechner ersetzt. KAlgebra hat numerische, logische, symbolische und analytische Fähigkeiten mit denen Sie mathematische Ausdrücke im Rechnerauswerten und die Ergebnisse zwei- oder dreidimensional darstellen können. KAlgebra basiert auf der Sprache „Mathematical Markup Language“ (MathML), es sind jedoch keine Kenntnisse von MathML erforderlich, um KAlgebra erfolgreich einsetzen zu können.

Kapitel 1

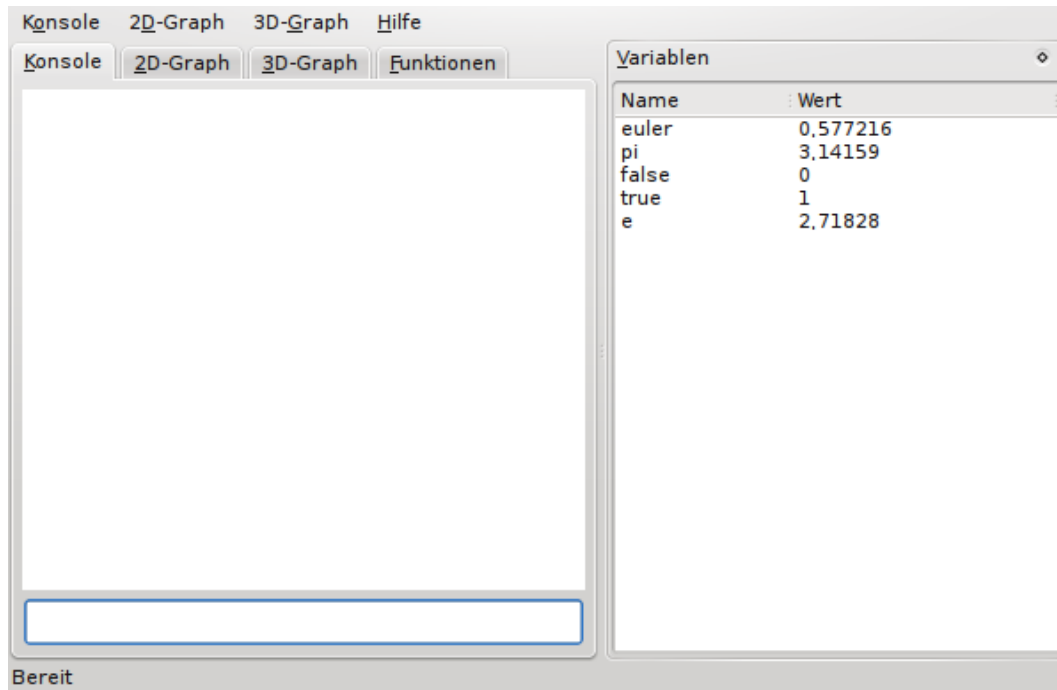
Einführung

KAlgebra bietet zahlreiche Möglichkeiten, um alle Arten von mathematischen Operationen auszuführen und deren Ausgaben graphisch darzustellen. Früher war dieses Programm stark an MathML ausgerichtet, heute kann es von Personen mit ein wenig mathematischem Verständnis benutzt werden, um sowohl einfache als auch schwierige mathematische Aufgaben zu lösen.

Das Programm bietet folgende Funktionsmerkmale:

- Ein Rechner für die schnelle und leichte Auswertung von mathematischen Funktionen.
- Skript-Fähigkeiten für erweiterte Folgen von Berechnungen.
- Funktionsdefinitionen und automatische Syntax-Vervollständigung
- Kalkülfunktionen einschließlich symbolischer Differentiation, Vektorkalkül und Listenbearbeitung.
- Grafische Darstellung von Funktionen mit der Möglichkeit, mit dem Cursor die Nullstellen zu finden und weitere Arten von Analysen.
- 3D-Grafik zur Visualisierung von dreidimensionalen Funktionen.
- Ein Funktionsverzeichnis für die schnelle Referenz zu allen vorhandenen Funktionen.

Hier sehen Sie ein Bildschirmfoto vom KAlgebra-Programm:



Das Hauptfenster von KAlgebra besteht aus den Karteikarten **Rechner**, **2D-Graph**, **3D-Graph** und **Funktionen**. Auf jeder Karteikarte befindet sich ein Textfeld zur Eingabe von Funktionen oder Berechnungen und ein Bereich für die Anzeige der Ergebnisse.

Sitzungen können jederzeit mit den Aktionen im Menü **Sitzungen** verwaltet werden:

Sitzung → **Neu (Strg+N)**

Öffnet ein neues KAlgebra-Hauptfenster

Sitzung → **Vollbildmodus (Strg+Umschalt+F)**

Schaltet den Vollbildmodus für das KAlgebra-Fenster ein und aus. Dies kann auch mit dem

Knopf  rechts oben im KAlgebra-Fenster ausgeführt werden.

Sitzung → **Beenden (Strg+Q)**

Beendet das Programm.

Kapitel 2

Syntax

KAlgebra verwendet eine intuitive algebraische Syntax für die Eingabe von benutzerdefinierten Funktionen, Eine ähnliche Syntax wird für die meisten modernen grafischen Rechner verwendet. In diesem Abschnitt werden alle in KAlgebra verfügbaren eingebauten Operatoren aufgeführt. Die Syntax von KAlgebra wurde in Anlehnung an die mathematischen Anwendungen [Maxima](#) und [Maple](#) entwickelt und sollte daher allen Benutzern dieser Programme vertraut sein.

In KAlgebra werden die eingegebenen Ausdrücke in MathML für das Hintergrundprogramm umgewandelt. Ein grundsätzliches Verständnis der durch MathML unterstützten Fähigkeiten ist hilfreich, um die Möglichkeiten von KAlgebra zu entdecken.

Folgende Operatoren sind in dieser Version enthalten:

- $+$ $-$ $*$ $/$: Addition, Subtraktion, Multiplikation und Division.
- $^$, $**$: Potenzen, beide Schreibweisen und auch das Unicodezeichen ² sind erlaubt. Potenzen können auch als Wurzel benutzt werden, zum Beispiel: $a^{**}(1/b)$
- \rightarrow : Lambda-Funktion. Die Möglichkeit, eine oder mehrere freie Variablen zu definieren, die an eine Funktion gebunden werden. Im Ausdruck **length := (x, y) -> (x*x+y*y) ^ 0.5** zum Beispiel wird der Lambda-Operator zur Kennzeichnung benutzt, dass x und y als gebundene Variablen bei der Funktion „length“ verwendet werden.
- $x=a..b$: Einschränkung eines Wertebereichs (beschränkte Variable mit unterer und oberer Grenze). Die Variable x kann in diesem Beispiel nur Werte von a bis b annehmen.
- $()$: Klammern bestimmen den Vorrang einer auszuführenden Rechenoperation vor anderen in der Rechenreihenfolge.
- $abc(\text{parameter})$: Funktionen. Beim Einlesen von Funktionen wird überprüft, ob abc ein Operator ist. Wenn ja, dann wird es als Operator behandelt, ansonsten als benutzerdefinierte Funktion.
- $:=$: Definition des Wertes einer Variablen. Definitionen wie $\mathbf{x:=3}$, $\mathbf{x:=y}$ sind möglich, auch wenn y noch nicht definiert ist, oder zum Beispiel **umfang := r -> 2*pi*r**.
- $?$: Abschnittsweise definierte Bedingungen. Damit können bedingte Operationen in KAlgebra definiert werden. Anders ausgedrückt ist das eine Möglichkeit „if, elseif, else“-Bedingungen zu definieren. Das Ergebnis einer abschnittsweise definierten Funktion ist der Ausdruck nach dem $?$ der ersten wahren Bedingung. Trifft keine Bedingung zu, wird der Ausdruck nach dem letzten $?$ ohne vorangestellte Bedingung zurückgegeben. Beispiel: **piecewise { x=0 ? 0, x=1 ? x+1, ? x**2 }**
- $\{ \}$: Damit werden MathML-Container definiert, die hauptsächlich bei abschnittsweise definierten Bedingungen verwendet werden.

Das Handbuch zu KAlgebra

- = > >= < <= : Vergleichsoperatoren für gleich, größer, größer oder gleich, kleiner und kleiner oder gleich.

Mit diese MathML-ähnlichen Syntax kann mit Funktionen wie `cos()`, `sin()`, jeder beliebigen trigonometrischen Funktion, `sum()` oder `product()` gearbeitet werden. Sie können Funktionen wie `plus()`, `mal()` und jede beliebige Funktion mit Operator verwenden. Auch Boolesche Funktionen sind implementiert, Sie können zum Beispiel mit Ausdrücken wie `or(1, 0, 0, 0, 0)` arbeiten.

Kapitel 3

Arbeiten mit dem Rechner

KAlgebras Rechner hat erweiterten Fähigkeiten. Es können Ausdrücke im Modus **Berechnen** oder **Auswerten** eingegeben werden, je nach Auswahl im Menü **Rechner**.

Im Auswertungsmodus vereinfacht KAlgebra Ausdrücke, auch wenn sie eine nicht definierte Variable enthalten. Im Berechnungsmodus wird alles berechnet und bei einer nicht definierte Variablen eine Fehlermeldung angezeigt.

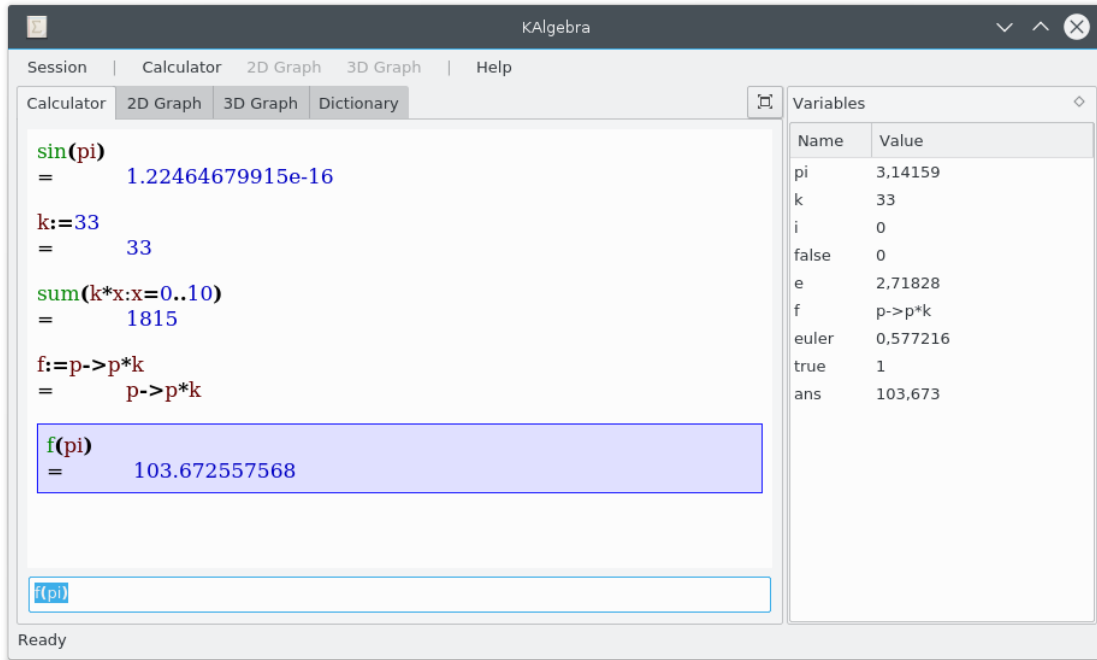
Zusätzlich zur Anzeige der vom Benutzer eingegebenen Ausdrücke und deren Ergebnissen in der Karteikarte Rechner werden alle deklarierten Variablen in einer Liste rechts angezeigt. Durch Doppelklicken auf eine Variable in der Liste öffnen Sie einen Dialog, in dem Sie den Wert der Variablen ändern oder Sie entfernen können.

„ans“ ist eine besondere Variable, sie enthält immer das Ergebnis der Auswertung des zuletzt eingegeben Ausdrucks.

Als Beispiel folgen einige Funktionen, die in das Eingabefeld der Karteikarte **Rechner** eingetragen werden können:

- `sin(pi)`
- `k:=33`
- `sum(k*x : x=0..10)`
- `f:=p->p*k`
- `f(pi)`

Das folgende Bildschirmfoto zeigt die Karteikarte Rechner nach der Eingabe der oben genannten Beispiel-Ausdrücke:



Mit den Aktionen für Skripte im Menü **Rechner** kann die Ausführung einer Folge von Berechnungen gesteuert werden:

Rechner → Skript laden (Strg+L)

Führt alle Anweisungen in der Skriptdatei nacheinander aus. Damit können Sie zum Beispiel mathematische Funktionen aus Bibliotheken oder Ihre vorher erstellte Arbeit wieder laden.

Rechner → Zuletzt geöffnete Skripte

Zeigt eine Untermenü zur Auswahl der zuletzt geöffneten Skripte an.

Rechner → Skript speichern (Strg+G)

Speichert alle Anweisungen, die seit Beginn der Sitzung eingegeben wurden, in eine Datei. Diese Textdatei kann dann zum Beispiel mit einem Editor wie Kate korrigiert oder bearbeitet werden.

Rechner → Protokoll speichern (Strg+S)

Speichert das Protokoll mit allen Ergebnissen in einer HTML-Datei, um sie ausdrucken oder weiterzugeben.

Rechner → ANS einfügen ... (F3)

Fügt die Variable „ANS“ ein, damit können vorher berechnete Werte wiederverwendet werden.

Rechner → Berechnen

Setzt den **Ausführungsmodus** auf Berechnen.

Rechner → Auswerten

Setzt den **Ausführungsmodus** auf Auswerten.

Kapitel 4

2D-Graphen

Um einen neuen 2D-Graphen in KAlgebra einzugeben, wechseln Sie zur Karteikarte **2D-Graph**. Klicken Sie dann auf den Karteireiter **Hinzufügen** und geben Sie im Textfeld oben die neue Funktion ein.

4.1 Syntax

Eine Funktion des Typs $f(x)$ können Sie direkt so eingeben. Für Funktionen des Typs $f(y)$ oder für polare Funktionen müssen Sie $y \rightarrow$ und $q \rightarrow$ als die gebundenen Variablen hinzufügen.

Beispiele:

- `sin(x)`
- `x2`
- `y->sin(y)`
- `q->3*sin(7*q)`
- `t->vector{sin t, t**2}`

Wenn Sie eine Funktion eingegeben haben, klicken Sie auf den Knopf **OK**, um den Graphen der Funktion auf der Karteikarte anzuzeigen.

4.2 Eigenschaften

Mehrere Graphen können in derselben Ansicht angezeigt werden. Wechseln Sie dazu auf die Karteikarte **Hinzufügen**. Jeder Graph kann in einer eigenen Farbe dargestellt werden.

Die Ansicht kann mit dem Mausrad vergrößert oder verkleinert werden. Mit der linken Maustaste können Sie Bereiche in der Ansicht zur Vergrößerung auswählen. Mit den Pfeiltasten kann die Ansicht verschoben werden.

ANMERKUNG

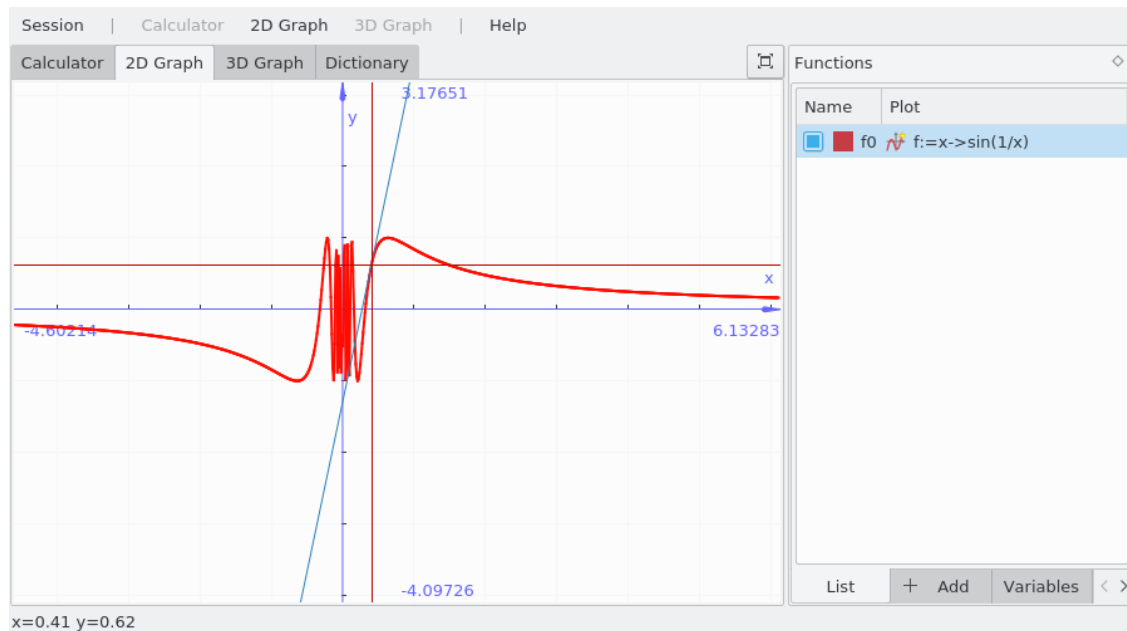
Das Darstellungsfeld für einen **2D-Graph** kann auf der Karteikarte **Darstellungsfeld** definiert werden.

Auf der Karteikarte **Liste** können Sie unten rechts mit Doppelklick auf eine Funktion die Karteikarte **Bearbeiten** öffnen. Dort kann die Funktion geändert werden. Verwenden Sie das Ankreuzfeld links neben dem Funktionsnamen, um die Anzeige der Funktion ein- und auszuschalten.

Im Menü **2D-Graph** sind diese Einträge vorhanden:

- **Gitter:** Zeigt die Gitternetzlinien an oder blendet sie aus
- **Seitenverhältnis beibehalten:** Seitenverhältnis bei Größenänderung beibehalten
- **Speichern:** Den Graphen als Bilddatei speichern (**Strg+S**)
- **Vergrößern/Verkleinern:** Vergrößern (**Strg++**) und Verkleinern (**Strg+-**) der Ansicht
- **Originalgröße:** Ansicht wieder auf die Originalgröße zurücksetzen
- **Auflösung:** Auflösung des Graphen auswählen

Das folgende Bildschirmfoto zeigt die Funktion $\sin(1/x)$ mit dem Cursor an der rechten Nullstelle der Funktion. Es wird eine sehr hohe Auflösung zur Anzeige verwendet, da die Funktion mit immer höherer Frequenz in der Nähe des Koordinatenursprungs schwingt. Folgen Sie mit dem Cursor dem Verlauf der Funktion, dann wird für jeden Punkt der x - und y -Wert der Funktion links unten angezeigt und diese Anzeige laufend aktualisiert. Gleichzeitig wird die „Tangente“ der Funktion am aktuellen Schnittpunkt mit dem Cursor gezeichnet.



Kapitel 5

3D-Graphen

Wechseln Sie zur Karteikarte **3D-Graph** und geben Sie im Textfeld unten eine Funktion ein. Zurzeit kann noch kein Funktionswert für die z-Achse eingegeben werden. KAlgebra erlaubt nur 3D-Funktionen abhängig von x und y, wie $(x, y) \rightarrow x * y$, dabei ist $z = x * y$.

Beispiele:

- $(x, y) \rightarrow \sin(x) * \sin(y)$
- $(x, y) \rightarrow x/y$

Die Ansicht kann mit dem Mauszeiger verschoben und mit dem Mauseisrad vergrößert oder verkleinert werden. Mit der linken Maustaste können Bereiche in der Ansicht zur Vergrößerung ausgewählt werden.

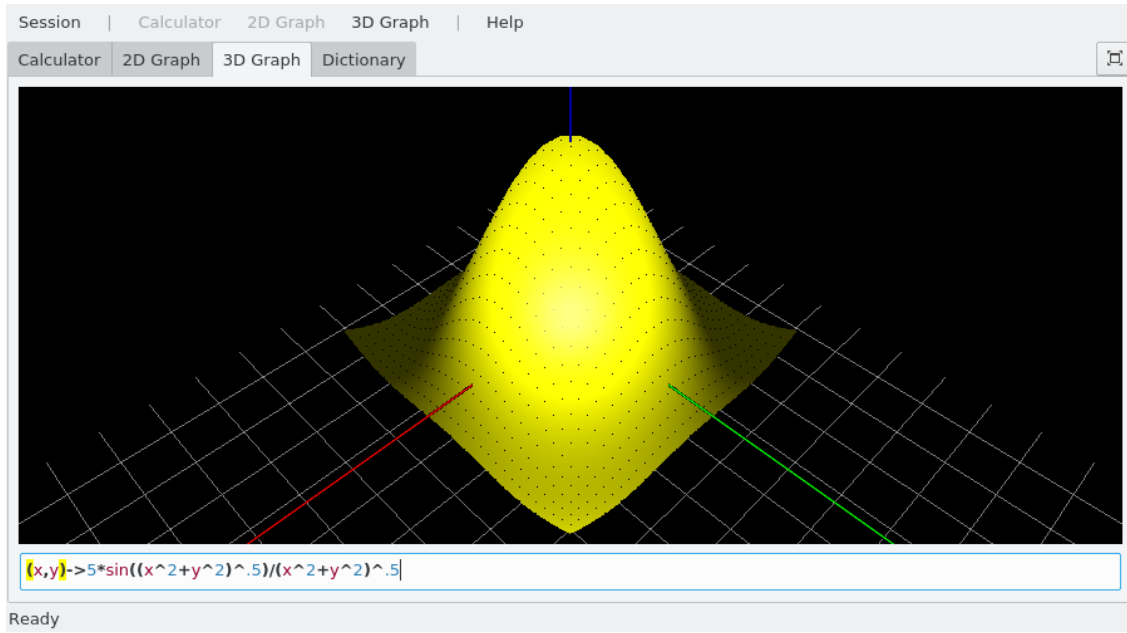
Die Tasten **Pfeil links** und **Pfeil rechts** drehen die Grafik um die senkrechte z-Achse, die Tasten **Pfeil hoch** und **Pfeil runter** drehen die Grafik um die horizontale Achse der Ansicht. Drücken Sie die Taste **W** zum Vergrößern und die Taste **S** zum Verkleinern der Ansicht.

Im Menü **3D-Graph** sind diese Einträge vorhanden:

- **Speichern**: Den Graphen als Bilddatei oder unterstütztes Dokument speichern (**Strg+S**)
- **Ansicht zurücksetzen**: Ansicht wieder auf die Originalgröße in der Karteikarte **3D-Graph** zurücksetzen
- Anzeige der Grafik als Gitternetz aus **Punkten**, **Linien** oder als **gefüllte Fläche** mit einem Liniengitter in der Karteikarte **3D-Graph**.

Hier sehen Sie ein Bildschirmfoto der sogenannten „Sombbrero“-Funktion, die 3D-Grafik wird im Liniengitter-Stil angezeigt.

Das Handbuch zu KAlgebra

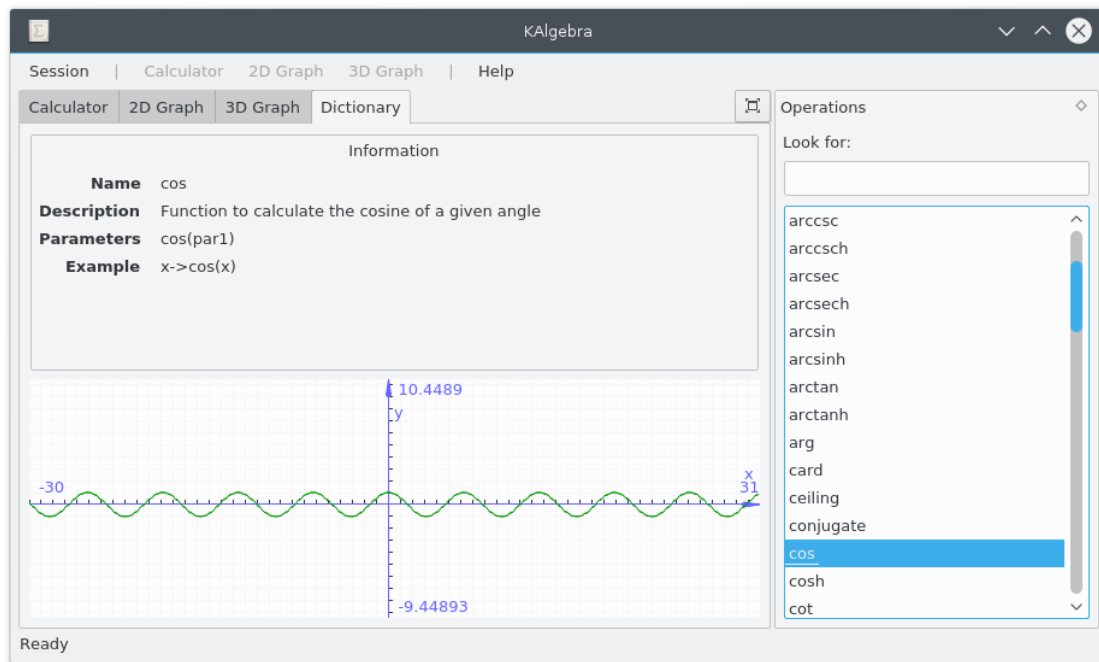


Kapitel 6

Funktionen

Auf der Karteikarte **Funktionen** finden Sie eine Liste aller in KAlgebra verfügbaren Funktionen. Wählen Sie eine Funktion aus der Liste oder im Suchfeld, dann wird die Beschreibung, die Parameter und ein Beispiel dieser Funktion mit dem Graphen links angezeigt.

Hier sehen Sie ein Bildschirmfoto der KAlgebra-Karteikarte „Funktionen“ mit der Anzeige der Cosinus-Funktion.



Kapitel 7

Referenz der Funktionen für KAlgebra

7.1 plus

- Name: plus
- Beschreibung: Addition
- Parameter: plus(... Parameter, ...)
- Beispiel: $x \rightarrow x+2$

7.2 times

- Name: times
- Beschreibung: Multiplikation
- Parameter: times(... Parameter, ...)
- Beispiel: $x \rightarrow x*2$

7.3 minus

- Name: minus
- Beschreibung: Subtraktion. Subtrahiert alle Werte vom ersten Wert.
- Parameter: minus(... Parameter, ...)
- Beispiel: $x \rightarrow x-2$

7.4 divide

- Name: divide
- Beschreibung: Division
- Parameter: divide(par1, par2)
- Beispiel: $x \rightarrow x/2$

7.5 quotient

- Name: quotient
- Beschreibung: Quotient
- Parameter: quotient(par1, par2)
- Beispiel: $x \rightarrow \text{quotient}(x, 2)$

7.6 power

- Name: power
- Beschreibung: Potenz
- Parameter: power(par1, par2)
- Beispiel: $x \rightarrow x^2$

7.7 root

- Name: root
- Beschreibung: Wurzel
- Parameter: root(par1, par2)
- Beispiel: $x \rightarrow \text{root}(x, 2)$

7.8 factorial

- Name: factorial
- Beschreibung: Fakultät. $\text{factorial}(n) = n!$
- Parameter: factorial(par1)
- Beispiel: $x \rightarrow \text{factorial}(x)$

7.9 and

- Name: and
- Beschreibung: Boolesches Und
- Parameter: and(... Parameter, ...)
- Beispiel: $x \rightarrow \text{piecewise} \{ \text{and}(x > -2, x < 2) ? 1, ? 0 \}$

7.10 or

- Name: or
- Beschreibung: Boolesches Oder
- Parameter: or(... Parameter, ...)
- Beispiel: $x \rightarrow \text{piecewise } \{ \text{or}(x > 2, x > -2) ? 1, ? 0 \}$

7.11 xor

- Name: xor
- Beschreibung: Boolesches exclusives Oder
- Parameter: xor(... Parameter, ...)
- Beispiel: $x \rightarrow \text{piecewise } \{ \text{xor}(x > 0, x < 3) ? 1, ? 0 \}$

7.12 not

- Name: not
- Beschreibung: Boolesches Nicht
- Parameter: not(par1)
- Beispiel: $x \rightarrow \text{piecewise } \{ \text{not}(x > 0) ? 1, ? 0 \}$

7.13 gcd

- Name: gcd
- Beschreibung: Größter gemeinsamer Teiler
- Parameter: gcd(... Parameter, ...)
- Beispiel: $x \rightarrow \text{gcd}(x, 3)$

7.14 lcm

- Name: lcm
- Beschreibung: Kleinstes gemeinsames Vielfaches
- Parameter: lcm(... Parameter, ...)
- Beispiel: $x \rightarrow \text{lcm}(x, 4)$

7.15 rem

- Name: rem
- Beschreibung: Rest
- Parameter: rem(par1, par2)
- Beispiel: $x \rightarrow \text{rem}(x, 5)$

7.16 factorof

- Name: factorof
- Beschreibung: Faktor von
- Parameter: factorof(par1, par2)
- Beispiel: $x \rightarrow \text{factorof}(x, 3)$

7.17 max

- Name: max
- Beschreibung: Maximum
- Parameter: max(... Parameter, ...)
- Beispiel: $x \rightarrow \text{max}(x, 4)$

7.18 min

- Name: min
- Beschreibung: Minimum
- Parameter: min(... Parameter, ...)
- Beispiel: $x \rightarrow \text{min}(x, 4)$

7.19 lt

- Name: lt
- Beschreibung: Kleiner als. $\text{lt}(a,b)=a<b$
- Parameter: lt(par1, par2)
- Beispiel: $x \rightarrow \text{piecewise} \{ x < 4 ? 1, ? 0 \}$

7.20 gt

- Name: gt
- Beschreibung: Größer als. $\text{gt}(a,b)=a>b$
- Parameter: gt(par1, par2)
- Beispiel: $x \rightarrow \text{piecewise} \{ x > 4 ? 1, ? 0 \}$

7.21 eq

- Name: eq
- Beschreibung: Gleich. $\text{eq}(a,b) = a=b$
- Parameter: $\text{eq}(\text{par1}, \text{par2})$
- Beispiel: $x \rightarrow \text{piecewise} \{ x=4 ? 1, ? 0 \}$

7.22 neq

- Name: neq
- Beschreibung: Ungleich. $\text{neq}(a,b) = a \neq b$
- Parameter: $\text{neq}(\text{par1}, \text{par2})$
- Beispiel: $x \rightarrow \text{piecewise} \{ x \neq 4 ? 1, ? 0 \}$

7.23 leq

- Name: leq
- Beschreibung: Kleiner-Gleich. $\text{leq}(a,b) = a \leq b$
- Parameter: $\text{leq}(\text{par1}, \text{par2})$
- Beispiel: $x \rightarrow \text{piecewise} \{ x \leq 4 ? 1, ? 0 \}$

7.24 geq

- Name: geq
- Beschreibung: Größer-Gleich. $\text{geq}(a,b) = a \geq b$
- Parameter: $\text{geq}(\text{par1}, \text{par2})$
- Beispiel: $x \rightarrow \text{piecewise} \{ x \geq 4 ? 1, ? 0 \}$

7.25 implies

- Name: implies
- Beschreibung: Boolesche Implikation
- Parameter: $\text{implies}(\text{par1}, \text{par2})$
- Beispiel: $x \rightarrow \text{piecewise} \{ \text{implies}(x < 0, x < 3) ? 1, ? 0 \}$

7.26 approx

- Name: approx
- Beschreibung: Näherungswert. $\text{approx}(a) = a \pm n$
- Parameter: $\text{approx}(\text{par1}, \text{par2})$
- Beispiel: $x \rightarrow \text{piecewise} \{ \text{approx}(x, 4) ? 1, ? 0 \}$

7.27 abs

- Name: abs
- Beschreibung: Absoluter Wert. $\text{abs}(n) = |n|$
- Parameter: $\text{abs}(\text{par1})$
- Beispiel: $x \rightarrow \text{abs}(x)$

7.28 floor

- Name: floor
- Beschreibung: Abrunden $\text{floor}(n) = \lfloor n \rfloor$
- Parameter: $\text{floor}(\text{par1})$
- Beispiel: $x \rightarrow \text{floor}(x)$

7.29 ceiling

- Name: ceiling
- Beschreibung: Aufrunden. $\text{ceil}(n) = \lceil n \rceil$
- Parameter: $\text{ceiling}(\text{par1})$
- Beispiel: $x \rightarrow \text{ceiling}(x)$

7.30 sin

- Name: sin
- Beschreibung: Funktion, um den Sinus eines gegebenen Winkels auszurechnen
- Parameter: $\text{sin}(\text{par1})$
- Beispiel: $x \rightarrow \text{sin}(x)$

7.31 **cos**

- Name: `cos`
- Beschreibung: Funktion, um den Kosinus eines gegebenen Winkels auszurechnen
- Parameter: `cos(par1)`
- Beispiel: `x->cos(x)`

7.32 **tan**

- Name: `tan`
- Beschreibung: Funktion, um den Tangens eines gegebenen Winkels auszurechnen
- Parameter: `tan(par1)`
- Beispiel: `x->tan(x)`

7.33 **sec**

- Name: `sec`
- Beschreibung: Sekans
- Parameter: `sec(par1)`
- Beispiel: `x->sec(x)`

7.34 **csc**

- Name: `csc`
- Beschreibung: Kosekans
- Parameter: `csc(par1)`
- Beispiel: `x->csc(x)`

7.35 **cot**

- Name: `cot`
- Beschreibung: Kotangens
- Parameter: `cot(par1)`
- Beispiel: `x->cot(x)`

7.36 **sinh**

- Name: `sinh`
- Beschreibung: Sinus Hyperbolicus
- Parameter: `sinh(par1)`
- Beispiel: `x->sinh(x)`

7.37 cosh

- Name: cosh
- Beschreibung: Kosinus Hyperbolicus
- Parameter: cosh(par1)
- Beispiel: $x \rightarrow \cosh(x)$

7.38 tanh

- Name: tanh
- Beschreibung: Tangens Hyperbolicus
- Parameter: tanh(par1)
- Beispiel: $x \rightarrow \tanh(x)$

7.39 sech

- Name: sech
- Beschreibung: Sekans Hyperbolicus
- Parameter: sech(par1)
- Beispiel: $x \rightarrow \operatorname{sech}(x)$

7.40 csch

- Name: csch
- Beschreibung: Kosekans Hyperbolicus
- Parameter: csch(par1)
- Beispiel: $x \rightarrow \operatorname{csch}(x)$

7.41 coth

- Name: coth
- Beschreibung: Kotangens Hyperbolicus
- Parameter: coth(par1)
- Beispiel: $x \rightarrow \operatorname{coth}(x)$

7.42 arcsin

- Name: arcsin
- Beschreibung: Arkussinus
- Parameter: arcsin(par1)
- Beispiel: $x \rightarrow \arcsin(x)$

7.43 arccos

- Name: arccos
- Beschreibung: Arkuskosinus
- Parameter: arccos(par1)
- Beispiel: $x \rightarrow \arccos(x)$

7.44 arctan

- Name: arctan
- Beschreibung: Arkustangens
- Parameter: arctan(par1)
- Beispiel: $x \rightarrow \arctan(x)$

7.45 arccot

- Name: arccot
- Beschreibung: Arkuskotangens
- Parameter: arccot(par1)
- Beispiel: $x \rightarrow \operatorname{arccot}(x)$

7.46 arccosh

- Name: arccosh
- Beschreibung: Arkuskosinus Hyperbolicus
- Parameter: arccosh(par1)
- Beispiel: $x \rightarrow \operatorname{arccosh}(x)$

7.47 arccsc

- Name: arccsc
- Beschreibung: Arkuskosekans
- Parameter: arccsc(par1)
- Beispiel: $x \rightarrow \operatorname{arccsc}(x)$

7.48 arccsch

- Name: arccsch
- Beschreibung: Arkuskosekans Hyperbolicus
- Parameter: arccsch(par1)
- Beispiel: $x \rightarrow \operatorname{arccsch}(x)$

7.49 arcsec

- Name: arcsec
- Beschreibung: Arkussekans
- Parameter: arcsec(par1)
- Beispiel: $x \rightarrow \text{arcsec}(x)$

7.50 arcsech

- Name: arcsech
- Beschreibung: Arkussekans Hyperbolicus
- Parameter: arcsech(par1)
- Beispiel: $x \rightarrow \text{arcsech}(x)$

7.51 arcsinh

- Name: arcsinh
- Beschreibung: Arkussinus Hyperbolicus
- Parameter: arcsinh(par1)
- Beispiel: $x \rightarrow \text{arcsinh}(x)$

7.52 arctanh

- Name: arctanh
- Beschreibung: Arkustangens Hyperbolicus
- Parameter: arctanh(par1)
- Beispiel: $x \rightarrow \text{arctanh}(x)$

7.53 exp

- Name: exp
- Beschreibung: Exponent (e^x)
- Parameter: exp(par1)
- Beispiel: $x \rightarrow \text{exp}(x)$

7.54 **ln**

- Name: ln
- Beschreibung: Natürlicher Logarithmus zur Basis e
- Parameter: $\ln(\text{par1})$
- Beispiel: $x \rightarrow \ln(x)$

7.55 **log**

- Name: log
- Beschreibung: Dekadischer Logarithmus
- Parameter: $\log(\text{par1})$
- Beispiel: $x \rightarrow \log(x)$

7.56 **conjugate**

- Name: conjugate
- Beschreibung: konjugiert-komplexe Zahl
- Parameter: $\text{conjugate}(\text{par1})$
- Beispiel: $x \rightarrow \text{conjugate}(x*i)$

7.57 **arg**

- Name: arg
- Beschreibung: Polarwinkel einer komplexen Zahl
- Parameter: $\text{arg}(\text{par1})$
- Beispiel: $x \rightarrow \text{arg}(x*i)$

7.58 **real**

- Name: real
- Beschreibung: Realteil einer komplexen Zahl
- Parameter: $\text{real}(\text{par1})$
- Beispiel: $x \rightarrow \text{real}(x*i)$

7.59 imaginary

- Name: imaginary
- Beschreibung: Imaginärteil einer komplexen Zahl
- Parameter: imaginary(par1)
- Beispiel: $x \rightarrow \text{imaginary}(x*i)$

7.60 sum

- Name: sum
- Beschreibung: Summe
- Parameter: sum(par1 : var=von..bis)
- Beispiel: $x \rightarrow x*\text{sum}(t*t:t=0..3)$

7.61 product

- Name: product
- Beschreibung: Produkt
- Parameter: product(par1 : var=von..bis)
- Beispiel: $x \rightarrow \text{product}(t+t:t=1..3)$

7.62 diff

- Name: diff
- Beschreibung: Differentiation
- Parameter: diff(par1 : var)
- Beispiel: $x \rightarrow (\text{diff}(x^2:x))(x)$

7.63 card

- Name: card
- Beschreibung: Kardinal
- Parameter: card(par1)
- Beispiel: $x \rightarrow \text{card}(\text{vector} \{ x, 1, 2 \})$

7.64 scalarproduct

- Name: scalarproduct
- Beschreibung: Skalarprodukt
- Parameter: scalarproduct(... Parameter, ...)
- Beispiel: `x->scalarproduct(vector { 0, x }, vector { x, 0 })[1]`

7.65 selector

- Name: selector
- Beschreibung: Wählt das par1-te Element aus der Liste oder dem Vektor par2.
- Parameter: selector(par1, par2)
- Beispiel: `x->scalarproduct(vector { 0, x }, vector { x, 0 })[1]`

7.66 union

- Name: union
- Beschreibung: Verbindet mehrere Einträge desselben Typs
- Parameter: union(... Parameter, ...)
- Beispiel: `x->union(list { 1, 2, 3 }, list { 4, 5, 6 })[rem(floor(x), 5)+3]`

7.67 forall

- Name: forall
- Beschreibung: Für alle
- Parameter: forall(par1 : var)
- Beispiel: `x->piecewise { forall(t:t@list { true, false, false }) ? 1, ? 0 }`

7.68 exists

- Name: exists
- Beschreibung: Existenz
- Parameter: exists(par1 : var)
- Beispiel: `x->piecewise { exists(t:t@list { true, false, false }) ? 1, ? 0 }`

7.69 map

- Name: map
- Beschreibung: Wendet eine Funktion auf jedes Element einer Liste an
- Parameter: map(par1, par2)
- Beispiel: $x \rightarrow \text{map}(x \rightarrow x+x, \text{list } \{ 1, 2, 3, 4, 5, 6 \})[\text{rem}(\text{floor}(x), 5)+3]$

7.70 filter

- Name: filter
- Beschreibung: Entfernt alle Elemente, die nicht einer Bedingung entsprechen
- Parameter: filter(par1, par2)
- Beispiel: $x \rightarrow \text{filter}(u \rightarrow \text{rem}(u, 2)=0, \text{list } \{ 2, 4, 3, 4, 8, 6 \})[\text{rem}(\text{floor}(x), 5)+3]$

7.71 transpose

- Name: transpose
- Beschreibung: Transpose
- Parameter: transpose(par1)
- Beispiel: $x \rightarrow \text{transpose}(\text{matrix } \{ \text{matrixrow } \{ 1, 2, 3, 4, 5, 6 \} \})[\text{rem}(\text{floor}(x), 5)+3][1]$

Kapitel 8

Danksagungen und Lizenz

- Program copyright 2005-2009 Aleix Pol

Dokumentation Copyright 2007 Aleix Pol aleixpol@gmail.com

Übersetzung Burkhard Lück lueck@hube-lueck.de

Diese Dokumentation ist unter den Bedingungen der [GNU Free Documentation License](#) veröffentlicht.

Dieses Programm ist unter den Bedingungen der [GNU General Public License](#) veröffentlicht.