GeoGebra & Python im Mathematikunterricht

Zentrale Befehle - mit und ohne CAS

Reimund Vehling

30. November 2025

Inhalt

Analysis

Befehlslisten für die Analysis: Orientierung

Merke

GeoGebra besitzt zwei mathematische Arbeitswelten: das **Algebra-Fenster** (numerisch) und das **CAS-Fenster** (symbolisch). Viele Befehle funktionieren in beiden, aber nicht identisch.

1. Algebra-Fenster (numerisch)

- Funktionsgraphen, numerische Ableitung, Nullstellen
- numerisches Integral, Flächen, Summen
- keine Parameterfunktionen

2. CAS-Fenster (symbolisch)

- Definition mit $:=$
- Ableitungen, Grenzwerte, Integrale symbolisch
- Funktionsschar $f(k,x)$

Bemerkung

Beide Welten ergänzen sich: Algebra zeigt Werte, CAS zeigt Strukturen.

Analysis: Befehlsliste (Algebra-Fenster) — 1/2

Befehl	Erläuterung
`f(x)=...`	Definiert eine numerische Funktion für den Graphen
`Ableitung(f)`	Numerische Ableitung; erzeugt neuen Graphen von f'
`Ableitung(f, n)`	n-te numerische Ableitung
`Extrempunkte(f)`	Berechnet lokale Extrempunkte (numerisch)
`Wendepunkte(f)`	Berechnet Wendepunkte (numerisch)
`Nullstelle(f)`	Berechnet Nullstellen des Graphen (numerisch)
`Nullstelle(f, a, b)`	Nullstelle im Intervall $[a, b]$
`Schnittpunkt(f,g)`	Berechnet Schnittpunkte zweier Graphen

Befehl	Erläuterung
`Integral(f, a, b)`	Numerisches bestimmtes Integral
`FlächeZwischen(f,g,a,b)`	Fläche zwischen zwei Graphen im Intervall $[a, b]$
`Obersumme(f,a,b,n)`	Obersumme mit n Rechtecken
`Untersumme(f,a,b,n)`	Untersumme mit n Rechtecken
`Kurve((x,f(x)),x,a,b)`	Erzeugt parametrische Kurve
`Tangente(f,x0)`	Tangente an f in x_0
`Asymptote(f)`	Versucht, Asymptoten numerisch zu bestimmen
`PolynomInterpolate(L)`	Interpolationspolynom aus Punktliste L

Analysis: Befehlsliste (CAS-Fenster) — 1/3

Befehl	Erläuterung
`f(x):=...`	Symbolische Definition einer Funktion
`f(k,x):=...`	Parameterfunktion, z. B. Kurvenschar
`Ersetze(expr, k=...)`	Substitution in Ausdrücken oder Funktionen
`Vereinfachen(expr)`	Algebraische Vereinfachung
`Faktorisieren(expr)`	Faktorisierte Darstellung eines Polynoms
`Erweitere(expr)`	Ausmultiplizieren.
`TrigErweitern(expr)`	trigonometrische Erweiterung
`TrigReduzieren(expr)`	Reduktion trigonometrischer Ausdrücke.

Analysis: Befehlsliste (CAS-Fenster) — 2/3

Befehl	Erläuterung
`Ableitung(expr,x)`	Symbolische Ableitung nach x
`Ableitung(expr,x,n)`	n-te Ableitung.
`Löse(expr=0,x)`	Symbolisches Lösen
`Löse(eq1,eq2,x,y)`	Lösen von Gleichungssystemen
`Eliminate(eq1,eq2,x)`	Eliminiert Variable aus einem Gleichungssystem
`Summe(expr,i,a,b)`	Symbolische Summe
`Produkt(expr,i,a,b)`	Symbolisches Produkt.
`Reihe(expr, x, a)`	Taylorentwicklung von expr um $x = a$

Analysis: Befehlsliste (CAS-Fenster) — 3/3

Befehl	Erläuterung
`Grenzwert(expr, x, a)`	Grenzwert von expr für $x \to a$
`Grenzwert(expr, x, `∞`)`	Grenzwert gegen unendlich
`Integral(expr, x)`	Symbolisches Integral
`Integral(expr, x,a,b)`	Symbolisches bestimmtes Integral
`Stammfunktion(expr)`	Stammfunktion von expr.
`Zerlege(expr)`	Partialbruchzerlegung
`Numerisch(expr)`	Erzwingt numerische Ausgabe
`Dezimal(expr)`	Numerische Dezimalschreibweise

Analysis: Befehle in beiden Fenstern (unterschiedlich)

Befehl	Algebra-Fenster	CAS-Fenster
`Ableitung(f)`	numerische Ableitung	symbolische Ableitung
`Integral(f,a,b)`	numerischer Wert	exaktes Integral
`Löse(expr=0,x)`	numerische Lösung	symbolische Lösung(en)
`f(x)=...`	Graphendefinition	logischer Ausdruck
`{...}`	numerische Liste	symbolische Liste
`Folge(...)`	numerische Liste	symbolische Liste/Ausdrücke

Bemerkung

Gleicher Befehl – unterschiedliche Mathematik: Das CAS arbeitet symbolisch, das Algebra-Fenster numerisch.

CAS: typische Fehler

- "=" statt ":=" (Vergleich statt Definition)
- Variable vergessen ($f := x^2$ statt $f(x) := x^2$)
- Parameter nicht korrekt gesetzt ($f(2x)$ statt $f(2, x)$)
- Löse-Aufruf ohne Definition von Ableitungen
- fehlende Klammern: `(x+1)/(x-2)` vs. `x+1/x-2`
- Mischung von Algebra- und CAS-Syntax
- Listenindex vs. Funktionsauswertung: $L(2)$ vs. $f(2)$

Bemerkung

Die meisten CAS-Fehler sind syntaktisch – nicht fachlich.

CAS: drei Grundprinzipien

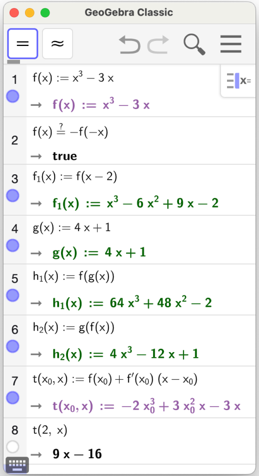
1. **Definition → Objekt** $f(x) := x^2, \quad f(k,x) := x^2 + kx$
2. **Operation → neues Objekt** $f_1(k,x) := \frac{\partial f}{\partial x}$
3. **Transformation → Struktur sichtbar** $\text{Ersetze}(f(k,x), k = 1/x)$

Bemerkung

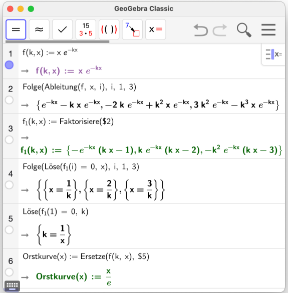
CAS = Objekte definieren, transformieren, weiterverwenden. Das zeigt Mathematik als Struktur.

CAS: Drei Beispiele auf einen Blick

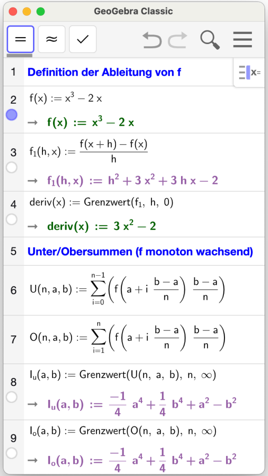
Definition: $:=$



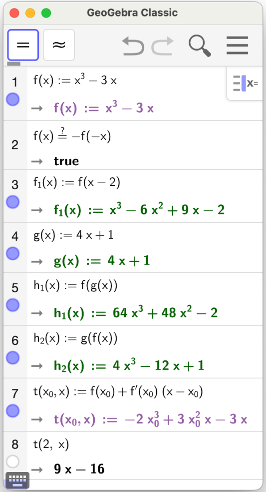
Kurvenschar: $f(k, x)$



Modulare Mathematik



CAS: Definition := statt Vergleich =

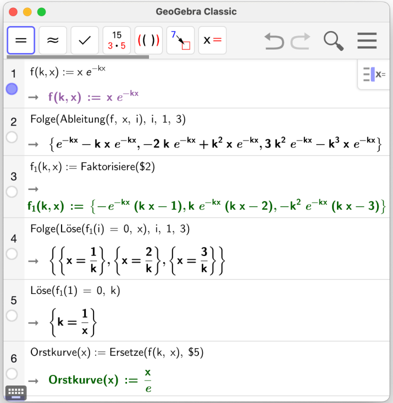


Merke

:= definiert ein Objekt.
= prüft nur eine Gleichheit (true/false).

- Im CAS ist := der wichtigste Operator.
- $f(x) := x^2$ erzeugt eine Funktion.
- $f(x) \stackrel{?}{=} -f(-x)$ liefert `true` oder `false`, hier `true`, also f punktsymmetrisch bzgl. $(0, 0)$.
- Funktionen als Objekte am Beispiel einer Verkettung
- Modulares Arbeiten: Nicht nur eine Tangente - alle auf einmal.

CAS: Parameterfunktionen und Kurvenscharen

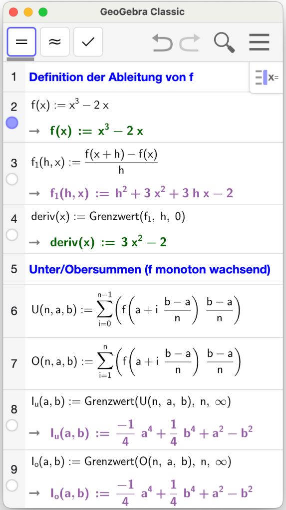


Merke

Funktionen können im CAS mehrere Variablen haben: $f(k,x)$ beschreibt eine ganze Funktionsfamilie.

- Beispiel: $f(k,x) := x\mathrm{e}^{-kx}$, statt $f_k(x)$
- Ableitungen bleiben Scharen: $f_1(k,x)$, $f_2(k,x)$
- Faktorisiere(), Vereinfache(), Multipliziere() - ohne Termkompetenz wenig Nutzen.
- Löse($f_1(k,x) = 0, x$) liefert *Kandidanten* für Extrempunkte für alle k.

CAS: Modulare Mathematik sichtbar machen



Merke

Im CAS entstehen neue Objekte Schritt für Schritt: Differenzenquotient $\rightarrow$ Grenzwert $\rightarrow$ Ableitung $\rightarrow$ Anwendungen.

- Differenzenquotient $f_1(h,x)$ definieren.
- Grenzwert $h \rightarrow 0$ ergibt die Ableitung.
- Summen und Grenzwerte führen zum Integral.