

# Computeralgebra mit Sage

Christoph Fuest

September 1, 2009

# Was ist ein CAS?

- ▶ CAS=Computer-Algebra-System
- ▶ Ein Computerprogramm, das symbolisch rechnet
- ▶ Dialogbasiert oder wie eine Programmiersprache
- ▶ Aus Eingabedaten werden Ausgabedaten erzeugt:
  - ▶ Grafiken
  - ▶ Zahlen
  - ▶ oder sogar Töne und Musik

# Was ist Sage?

- ▶ Sage ist kostenlos (sogar OpenSource)
- ▶ Sage läuft im Wesentlichen unter Linux
- ▶ ...geht aber auch unter Windows, Mac OS
- ▶ ...läuft dann aber in einer VM

# Schnellstart

- ▶ eine Konsole öffnen (z.B. Alt+F2 → `konsole`)
- ▶ Sage starten (in der Konsole `sage` eingeben)
- ▶ es öffnet sich eine Eingabezeile
- ▶ man führt nacheinander Befehle aus
- ▶ Es gibt Variablen, Funktionen, Werte

# Aber was eingeben?

- ▶ Wertzuweisungen
  - ▶ `a=1`
  - ▶ `alpha=23.44`
- ▶ Grundrechenarten
  - ▶ `a+2`
  - ▶ `a-15.0`
  - ▶ `a-alpha`
  - ▶ `1/3-12*alpha`

## ...und warum ist das nicht exakt?

Ganz einfach: Weil nicht exakte Zahlen in die Berechnung einfließen. Man probiere:

- ▶  $a=1$
- ▶  $\text{alpha}=2344/100$
- ▶  $a+2$
- ▶  $a-15$
- ▶  $a-\text{alpha}$
- ▶  $1/3-12*\text{alpha}$

# Was kann man sonst noch eingeben?

- ▶ Exponenten:  $2^3$  oder  $2**3$
- ▶ Mathematische (vordefinierte) Funktionen:
  - ▶ `sqrt(4)`
  - ▶ `ln(12)` oder `ln(12.0)` für die numerische Lösung
  - ▶ `sin(0)`, auch mit z.B. `sin(pi)`
  - ▶ `n(pi)` ist die numerische Approximation zu  $\pi$
- ▶ Kombinationen des oben angegebenen, z.B.:
  - ▶ `b=sqrt(a)-sin(pi)`

# Syntax der Sprache

- ▶ Namen beginnen mit einem Buchstaben oder `_`
- ▶ Leerzeichen am Zeilenanfang haben eine besondere Bedeutung und müssen falls nicht gewollt vermieden werden!
- ▶ Mehrere Befehle können in einer Zeile stehen und werden dabei durch `;` getrennt
- ▶ Was kommt hier heraus: `100-012`
- ▶ warum?
- ▶ Funktionen werden mit dem Funktionsnamen, gefolgt von einem Paar runder Klammern aufgerufen. Zwischen diesen stehen durch `,` getrennt eventuell die Argumente der Funktion.
- ▶ Kennt jemand die Programmiersprache Python?

# Syntax der Sprache

- ▶ Oft gewollt: Die Lösung(en) einer quadratischen Gleichung. Das kann sage mithilfe von
- ▶ `x=var('x')` kennzeichnet `x` als Variable
- ▶ `solve(x^2+3*x-5,x)`
- ▶ Das erste Argument ist ein Term, der “=0” gesetzt wird. Das zweite Argument ist die Variable, nach der aufgelöst wird.
- ▶ Das Ergebnis ist eine Liste, dazu gleich mehr

# Listen

- ▶ Listen werden in Python (und damit in sage) in eckigen Klammern notiert:
  - ▶ `meine_liste=[1,2,3,1]` erzeugt eine Liste mit den Einträgen 1, 2 und 3, gefolgt von einer weiteren 1
  - ▶ Anschließend bezeichnet `meine_liste[3]` das 3. Element der Liste. Welches ist das?
- ▶ Listen können Zahlen, Terme, Texte,... enthalten

# Lösen von GLS

- ▶ Zum Lösen von Gleichungssystemen kommt ebenfalls die Funktion `solve` zum Einsatz:
  - ▶ `x, y = var('x, y')`
  - ▶ `solve([x+y==6, x-y==4], x, y)`
  - ▶ hier kommt `==` zum Einsatz. Dieses ist das vergleichende Gleichheitszeichen
    - ▶ `1==2` hat den Wert falsch, `1==1` den Wert wahr

# Aufgaben

- ▶ Berechne:
  - ▶  $\sin(\pi) - \sin\left(\frac{\pi}{2}\right)$
  - ▶ die numerische Repräsentation von  $\sin(3)$
- ▶ Vereinfache:
  - ▶  $\sqrt{8}\sqrt{27}$
- ▶ In welchem Punkt/welchen Punkten schneidet die Gerade  $x = 4 - 2y$  die Parabel  $x = y^2 - 1$

# Differentiation

- ▶ Sehr hilfreich: Sage kann ableiten!
- ▶ Dazu dient der Befehl `diff`, der ein Funktional und die Differentiationsvariable als Argumente erhält
- ▶ Beispiel: Leite  $f(x) = x^2$  nach  $x$  ab: `diff(x^2,x)`
- ▶ Mehrfachableitungen gehen auch: `diff(x^2,x,2)` leitet 2 mal nach  $x$  ab

# Aufgaben

► Berechne:

- $\frac{d}{dx}x^4 - 20x^2 + 21x$
- $f'(x)$  von  $f(x) = 7x - 3y$
- $\left(\frac{d}{dx}\right)^2 \frac{d}{dy}2x - 4x^2y$

# Integration

- ▶ Sage kann auch integrieren
- ▶ Die Syntax funktioniert wie bei diff, bloß mit dem Funktionsnamen integral, statt diff
- ▶ z.B.: `integral(x^2, x)` integriert  $f(x) = x^2$  nach  $x$
- ▶ Für eigentliche Integrale können als weitere Parameter die untere und obere Grenze angegeben werden
- ▶ z.B.: `integral(x^2, x, 0, 2)`

# Aufgaben

► Berechne:

►  $\int x^2 dx$

►  $\int x^3 - 2x + 4 dx$

►  $\int_0^1 x^2 dx$

►  $\int_0^1 \int_0^2 xy + 2x^2 dx dy$

► “Beweise”:

►  $\int -x dx = - \int x dx$

# Funktionsgraphen plotten

- ▶ Sehr vielfältige Möglichkeiten zur Darstellung von Funktionsgraphen
- ▶ Schnellste: `plot(x^2-x,x,0,4)` plottet die Funktion  $f(x) = x^2 - x$  von 0 bis 4
- ▶ Plotten geht natürlich nur, falls keine Unbestimmten mehr in der Funktionsgleichung vorkommen.

# Funktionsgraphen plotten

- ▶ Plots können zur späteren Verwendung auch gespeichert werden, es findet dann keine automatische Ausgabe statt.
- ▶ `plot1=plot(x^2,x,0,3)`
- ▶ Das macht Sinn, wenn man mehrere Plots in eine Grafik zeichnen möchte. Wenn `plot1` und `plot2` zwei gespeicherte Plots sind, kann man diese einfach “addieren”: `plot1+plot2`

# Funktionsgraphen plotten

- ▶ Sehr sinnvoll bei der Verwendung mehrere Graphen in einem Plot: Verschiedene Farben

- ▶ Dies geht über benannte Argumente:

```
plot1=plot(x^2,x,0,4,color='red')
```

```
plot2=plot(x-2,x,0,3,color='blue')
```

```
plot1+plot2
```

# Aufgaben

- ▶ Man zeichne die Funktion  $f(x) = x^3 - 2x^2 - 2x + 1$  in dem Bereich von  $-4$  bis  $4$
- ▶ Man zeichne die Tangenten in  $x = 1$  in dasselbe Diagramm
- ▶ Man verwende für die Funktion rot und für die Tangente blau

- ▶ Da sage auf Python basiert, kann man alle Elemente dieser Programmiersprache verwenden. So auch eine for-Schleife:
- ▶ `for b in xrange(0,10):`
- ▶     `if (b==0):`
- ▶         `p=plot(x^2+b,x,-2,2)`
- ▶     `else:`
- ▶         `p=p+plot(x^2+b,x,-2,2)`