A. Filler, Humboldt-Universität zu Berlin

Einführung in das CAS Maxima

1 Arithmetik - Maxima als "besserer Taschenrechner"

Man kann Maxima wie einen Taschenrechner für arithmetische Aufgaben verwenden. Eine Eingabe ist mit Shift+Enter abzuschließen, dann gibt Maxima das Ergebnis zurück.

```
(%i1) 2+5;
```
(%o1) 7

```
(%i2) 2/3-5/7;
```
(%o2) $-\frac{1}{21}$

```
(%i3) 7^2;
```
(%o3) 49

```
(%i4) sqrt(33);
```
(%o4) $\sqrt{33}$

Maxima gibt im Normalfall exakte Werte aus, deshalb erhält man als Wurzel aus 33 eben $\sqrt{33}$. Möchte man numerische Näherungswerte, so kann man die Option `numer` verwenden oder das Ergebnis mittels `float()` in eine Gleitkommazahl umwandeln.

```
(%i5) sqrt(33), numer;
```
(%o5) 5.744562646538029

```
(%i6) float(sqrt(33));
```
(%o6) 5.744562646538029

Dezimalzahlen (Gleitkommazahlen) werden in der Regel mit 16 Stellen Genauigkeit ausgegeben. Obwohl Maxima intern mit doppelter Genauigkeit rechnet, kann es sinnvoll sein, die Genauigkeit zu erhöhen. Dazu arbeitet man mit "großen Gleitkommazahlen" (`bigfloat`). Die Anzahl der Stellen lässt sich durch die Optionsvariable `fpprec` angeben.

```
(%i7) fpprec: 50;
(%i8) bfloat(sqrt(33));
```
(%o8) $5.7445626465380286598506114682189293182202644579828b0$

Bemerkung: "`b0`" am Ende bedeutet "mal 10 hoch 0", siehe auch das folgende Beispiel.

```
(%i9) bfloat(sqrt(333));
```
(%o9) $1.8248287590894659066999052735606201186254910284359b1$

2 Rechnen mit Variablen, Termumformungen

2.1 Vereinfachen von Ausdrücken

In CAS kann man mit Variablen ebenso wie mit Zahlen rechnen, mitunter werden eingegebene Ausdrücke dadurch vereinfacht.

```
(%i10) a/3+a/2;
```
(%o10) $\frac{5\,a}{6}$

Wenn Maxima nicht von selbst vereinfacht (wie im letzten Beispiel) lassen sich Befehle wie `ratsimp` verwenden (siehe auch im Menü von wxMaxima: Vereinfachen). `ratsimp` vereinfacht rationale Ausdrücke.

```
(%i11) 1/(a+b) - 1/(a-b) + (a+b)/(a^2-b^2);
```
(%o11) $\frac{b+a}{a^2-b^2}+\frac{1}{b+a}-\frac{1}{a-b}$

```
(%i12) ratsimp(%);
```
(%o12) $\frac{1}{b+a}$

`%` bewirkt, dass `radsimp` auf die letzte Eingabe 1/(a+b) – 1/(a-b) + (a+b)/(a^2–b^2) angewendet wird. Man könnte statt dessen auch von vornherein Folgendes schreiben:

```
(%i13) ratsimp( 1/(a+b) - 1/(a-b) + (a+b)/(a^2-b^2) );
```

(%o13) $\frac{1}{b+a}$

Ein weiteres Beispiel:

```
(%i14) a^3/((a-b)*(a-c)) + b^3/((b-c)*(b-a)) + c^3/((c-a)*(c-b));
```

(%o14) $\frac{c^3}{(c-a)\,(c-b)} + \frac{b^3}{(b-a)\,(b-c)} + \frac{a^3}{(a-b)\,(a-c)}$

```
(%i15) ratsimp(%);
```

(%o15) $c+b+a$

Zum Vereinfachen von Wurzeln eignet sich `radcan`.

```
(%i16) sqrt(a*b)/sqrt(a);
```

(%o16) $\frac{\sqrt{a\,b}}{\sqrt{a}}$

```
(%i17) radcan(%);
```

(%o17) $\sqrt{b}$

2.2 Ausmultiplizieren (`expand`) und Ausklammern (`factor`)

```
(%i18) term1:(x+2)^3;
       expand(term1);
```

(%o18) $(x+2)^3$
(%o19) $x^3+6\,x^2+12\,x+8$

```
(%i20) term2: x^4 + 2*x^3 - 3*x^2 - 8*x - 4;
       factor(term2);
```

(%o20) $x^4+2\,x^3-3\,x^2-8\,x-4$
(%o21) $(x-2)\,(x+1)^2\,(x+2)$

Bei ganzen Zahlen bewirkt factor eine Zerlegung in Primfaktoren:

```
(%i22) factor(8823888346);
```

(%o22) $2\,7^3\,13\,727\,1361$

2.3 Zuweisen von Werten zu Variablen, Löschen von Zuweisungen

Häufig ist es sinnvoll, Variablennamen zu verwenden und Variablen konkrete Werte zuzuweisen, um Werte von Termen für verschiedene Ausgangswerte zu berechnen.

```
(%i23) a:11;
       (a+3)^4;
```

(%o23) 11

(%o24) 38416

VORSICHT: Sobald `a:11` ausgeführt wurde, ist für a der Wert 11 gespeichert. Der Variablenname a ist nicht mehr "frei". Man sieht das, indem man einen Term eingibt, in dem a auftritt:

```
(%i25) (a-5)^2;
```

(%o25) 36

Um alle Werte zu löschen und die entsprechenden Symbole zu "‘befreien"’, gibt man `kill(all);` ein oder wählt den Menübefehl Maxima -> Speicher löschen.

```
(%i26) kill(all);
```

```
(%i1) (a-5)^2;
```

(%o1) $(a-5)^2$

3 Lösen von Gleichungen

3.1 solve und algsys

Gleichungen und auch Gleichungssysteme lassen sich in Maxima mittels `solve` lösen. Es müssen dabei die Variablen angegeben werden, nach denen eine Gleichung oder ein Gleichungssystem gelöst werden sollen.

```
(%i2) solve(3*x+5=4, x);
```

(%o2) $[x = -\frac{1}{3}]$

```
(%i3) solve(3*x^2-3*x+1=0, x);
```

(%o3) $[x = -\frac{\sqrt{3}\,i-3}{6}, x = \frac{\sqrt{3}\,i+3}{6}]$

Soll ein Gleichungssystem gelöst werden, so müssen die Gleichungen in eckigen Klammern, getrennt durch Kommas, eingegeben werden, das gilt auch für die Variablen.

```
(%i4) solve([y+3*x+5=4, -4*y+5*x+7=8], [x,y]);
```

(%o4) $[[x = -\frac{3}{17}, y = -\frac{8}{17}]]$

Maxima ermittelt prinzipiell sowohl reelle als auch komplexe Lösungen von Gleichungen. In der Schule ist das oft unerwünscht. Deshalb kann es sinnvoll sein, den Grundbereich der Variablen einzuschränken. Dies ist mittels `realonly:true` möglich. Statt des Lösungsbefehls `solve` muss dann aber `algsys` verwendet werden.

```
(%i5) algsys([3*x^2-3*x+1=0],[x],realonly:true);
```

(%o5) $[]$

```
(%i6) gl:x^3-4*x^2+6*x-4=0;
      solve(gl,x);
```

(%o6) $x^3 - 4\,x^2 + 6\,x - 4 = 0$

(%o7) $[x = 1 - i, x = i + 1, x = 2]$

```
(%i8) algsys([gl],[x],realonly:true);
```

(%o8) $[[x = 2]]$

3.2 Umstellen von Gleichungen nach einer Variablen

Gleichungen mit mehreren Variablen lassen sich nach einer Variablen umstellen, indem man sie nach dieser Variablen löst.

```
(%i9) solve([y+3*x+5=4],[y]);
```

(%o9) $[y = -3\,x - 1]$

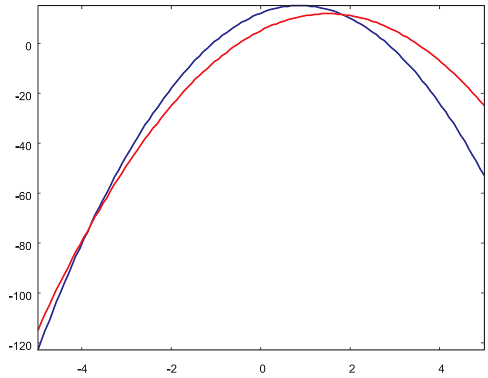
```
(%i10) solve([-4*y+5*x+7=8],[y]);
```

(%o10) $[y = \frac{5\,x-1}{4}]$

4 Darstellung von Funktionsgraphen und von durch Gleichungen gegebenen Geraden

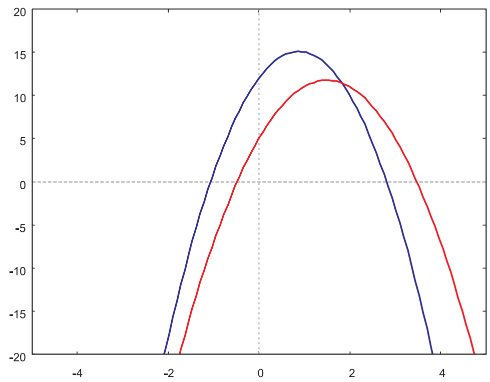
Maxima hat mehrere Möglichkeiten, Grafiken zu erstellen. Am flexibelsten und für zwei- und dreidimensionale Grafiken verwendbar sind `draw2d` und `draw3d`. Grafiken werden hierbei in separaten Fenstern dargestellt.

```
(%i11) f(x):= -4*x^2+7*x+12 $
       g(x):= -3*x^2+9*x+5  $
       xmin:-5 $
       xmax:5  $
       load("draw") $
       draw2d( color = blue, explicit(f(x), x,xmin,xmax) ,
               color = red, explicit(g(x), x,xmin,xmax))$
```



`draw2d` verfügt über eine Vielzahl von Optionen, erwähnt seien hier nur `yrange = [ymin,ymax]`, um den dargestellten Wertebereich zu wählen, sowie `user_preamble = ["set zeroaxis"]`, um Koordinatenachsen zu zeichnen.

```
(%i17) f(x):= -4*x^2+7*x+12  $
       g(x):= -3*x^2+9*x+5   $
       xmin:-5 $
       xmax:5  $
       load("draw") $
       draw2d( user_preamble = ["set zeroaxis"] ,
               yrange = [-20,20] ,
               color = blue, explicit(f(x), x,xmin,xmax) ,
               color = red, explicit(g(x), x,xmin,xmax))$
```



Während sich mittels `explicit()` Funktionsgraphen zeichnen lassen, stellt `implicit()` Lösungsmengen von Gleichungen dar. Im folgenden Beispiel bewirkt `"set size ratio 1"`, dass Höhe und Breite der erzeugten Grafik gleich sind.

```
(%i23) Gl11: 4*x - 5*y = 13  $
       Gl12: 3*x + 4*y = 3   $
       load("draw") $
       draw2d( user_preamble = ["set size ratio 1", "set zeroaxis"] ,
               color = blue, implicit(Gl11, x,-4,4, y,-4,4),
               color = red , implicit(Gl12, x,-4,4, y,-4,4) )$
```