Das Freie Computeralgebrasystem Maxima

Eine allerkürzeste Einführung

Dietmar Thaler

dietmar.thaler@posteo.at

Admont, 1. Mai 2019

Ein Computeralgebrasystem (CAS) ist ein Programm zur Bearbeitung algebraischer Ausrücke. Man kann damit nicht nur arithmetische Berechnungen durchführen wie mit einer Taschenrechner, sondern auch symbolische Berechnungen „mit Buchstaben". Maxima ist ein System, welches als Macsyma bereits in den 1960ern am MIT entwickelt wurde. Es ist in Lisp geschrieben. Seit 1998 steht es unter der Gnu Public License (GPL), einer freien Lizenz. Seit 2001 führt es den heutigen Namen.

1. *Woher bekommt man Maxima?* - Man findet es auf maxima.sourceforge.net/index.html.

2. *Für welche Plattformen gibt es Maxima?* - Es gibt es für für Linux, Windows, MacOS und Android.

3. *Wie installiert man Maxima?* - Man installiert es für die unterschiedlichen Plattformen gemäß der Anleitung auf maxima.sourceforge.net/download.html.

4. *Welche Benutzeroberflächen existieren für Maxima?* - Es gibt (ausgenommen für Android) drei verschieden Userinterfaces bzw. Frontends:

   a) Eine einfache *Terminal-Schnittstelle*, die ohne grafische Zusatzfeatures auskommt und in Linux mit dem Konsolenbefehl `"maxima"` gestartet wird, in anderen Systemen über entsprechende Startmenüeinträge. Alle Maxima-Befehle müssen getippt werden.

   b) Eine *grafische Oberfläche mit sparsamem Komfort*, die sich in der Bedienungsführung noch stark an die Terminal-Schnittstelle anlehnt: xmaxima. Die meisten Befehl müssen noch getippt werden.

   c) Eine *grafische Oberfläche mit erheblichem Zusatzkomfort* und Zusatzfunktionalität. Viele Maxima-Befehle sind über ein Menü erreichbar und müssen nicht mehr eingetippt werden: wxMaxima.

   Aus Gründen der Darstellungs-Ökonomie wird in den kommenden Beispielen von der minimalen Terminal-Schnittstelle ausgegangen. Für praktische Arbeiten wird die Komfortoberfläche Wxmaxima empfohlen. Der Hauptunterschied besteht darin, dass in der Terminaloberfläche und in xmaxima Befehle explizit mit dem Semikolon `";<Return>"` abgeschlossen werden, während in wxMaxima dieser Vorgang durch die Oberfläche mit `"<Strg><Return>"` erleichtert wird. – Unabhängig von der Benutzeroberfläche lässt sich Maxima scripten. Das heißt, man kann Maxima-Programme (auch mit Kontrollstrukturen) in Textdateien schreiben und diese in Maxima ausführen lassen.

5. *Wo findet man die Dokumentation zu* Maxima? - In den grafischen Benutzeroberflächen erhält man unmittelbaren Zugang zur aktuellen Dokumentation über ein Hilfe-System, sonst auf maxima.sourceforge.net/documentation.html. Ebenso findet man da eine Reihe von Tutorials. Sehr empfehlenswert als übersichtliche, kurze und in sich geschlossene Einführung ist jene von *Richard H. Rand* auf maxima.sourceforge.net/docs/manual/intromax.html, die auch als PDF-Datei verfügbar ist.

6. *Wie präzise rechnet Maxima?* - Grundsätzlich immer exakt, weil es sich auf natürliche und rationale Zahlen (Brüche) beschränkt. Irrationale Zahlen (algebraische und transzendente) werden nicht ausgeführt sondern als solche angezeigt (z.b. $\sqrt{2}$, π als `sqrt(2)` und `%pi` etc..). Gleitkommazahlen müssen in der Ausgabe erzwungen werden, in der Eingabe sind sie jederzeit möglich.

7. *Was kann Maxima?* - Manipulation algebraischer Ausdrücke, Grenzwerte, Gleichungen und Gleichungssysteme symbolisch und numerisch, Differenzieren, Integrieren symbolisch und numerisch, Reihen, Matrizenmanipulationen, gewöhnliche Differenzialgleichungen symbolisch und numerisch, 2-dimensionale und 3-dimensionale Plots usw.. Über mitgelieferte und nachladbare Pakete ist darüber hinaus noch eine Menge von Zusatzfunktionalität möglich (z.B. Vektoralgebra und elementare Vektoranalysis). Auskunft gibt die Dokumentation.

8. *Ist Maxima erweiterbar?* - Ja. Über externe Scripts und die Programmierfunktionalität lassen sich eigene Pakete schreiben. Weiter gehende Funktionserweiterungen sind mit Lisp möglich.

Beispiel einer interaktiven Maxima-Sitzung (im Terminal nach Aufruf von "`maxima --quiet`")

```
(%i1) f: (x-3)*(x^2-3*x-4);
                                        2
(%o1)                          (x - 3) (x  - 3 x - 4)
(%i2) expand(f);
                                 3       2
(%o2)                           x  - 6 x  + 5 x + 12
(%i3) solve(%,x);
(%o3)                          [x = 4, x = - 1, x = 3]
(%i4) diff(%o2,x,1);
                                       2
(%o4)                              3 x  - 12 x + 5
(%i5) integrate(%o2,x);
                                 4                2
                                x         3    5 x
(%o5)                           -- - 2 x  + ---- + 12 x
                                4              2
(%i6) integrate(%o2,x,0,5);
                                          115
(%o6)                                     ---
                                           4
(%i7) dgl: 'diff(y,x,2) + omega**2 * y = 0;
                                   2
                                  d y          2
(%o7)                             --- + omega  y = 0
                                    2
                                  dx
(%i8) assume(omega**2 > 0);
                                          2
(%o8)                                [omega  > 0]
(%i9) gen: ode2(dgl,y,x);
(%o9)                  y = %k1 sin(omega x) + %k2 cos(omega x)
(%i10) ic2(gen, x=0, y=y0, 'diff(y,x)=0);
(%o10)                            y = cos(omega x) y0
(%i11)
```

Anmerkungen:

- Zuweisungen zu einer Variablen erfolgen durch den Doppelpunkt "**:**" , also z.B. "`f: x^2-4`"
- "`expand()`" expandiert Ausdrücke, andere nützliche Vereinfachungsfunktionen sind u.a. "`ratsimp(..)`", "`factor(..)`", "`radcan(..)`" und viele mehr. Vergleiche maxima.sourceforge.net/docs/manual/maxima_9.html#SEC46.
- Mit dem Label "%" wird immer der letzte Output angesprochen
- Mit dem Label "%o2" wird der Output nach der 2. Eingabe angesprochen, mit "%i2" die 2. Eingabe selbst.
- "`ode2(..)`" liefert die allgemeine Lösung der gewöhnlichen Differenzialgleichung, "`ic2(..)`" die spezielle Lösung des Anfangswertproblems der Dfgl. zweiter Ordnung durch Eingabe der Anfangsbedinungen für $y = y_0$ und $y' = 0$ an der Stelle $x = 0$ für die allgemeine Lösung. "`ic1(..)`" liefert die spezielle Lösung für ein Anfangswertproblem erster Ordnung und "`bc()`" löst ein Randwertproblem.
- Mit dem Befehl "`assume(..)`" werden vorab bekannte Annahmen (über Vorzeichen, etc..) dem CAS mitgeteilt, damit während der Ausführung keine interaktiven Abfragen erfolgen müssen.
- Bei aller Erleichterung, die Maxima als CAS bringt, ist zu beachten, dass nicht jedes Problem analytisch lösbar ist und nicht jede Lösung automatisch in praktikabler Form geliefert wird. „Handarbeit“ und mathematischer Hausverstand sind weiterhin angeraten.

Eine etwas ausführlichere Pdf-Präsentation zu Maxima findet man auch unter glt.foehnwall.at/glt19.html.