

Computer Algebra

Sebastian Iwanowski
FH Wedel

1. Einführung in das Computer-Algebra-System Maxima

Einführung in das CAS Maxima

On-line Literatur zu Maxima

- On-line Hilfe zu Maxima (integriert in Werkzeug)
- Reference Manual

- Maxima-Beispiele Jürgen Sprau (BBS Rodalben/Pfalz)
- Minimal Maxima Robert Dodier (University of Colorado)
- Introduction Richard Rand (Cornell University)
- Einführung Robert Glöckner (EDV-Experte)

- Vergleich zu anderen CAS: Vorlesungsskript Gräbe (Uni Leipzig)

Einführung in das CAS Maxima

Überblick über die Sprachelemente

- Funktional: Lisp-basiert
 - Listenorientierung, Auswertung und Nichtauswertung von Ausdrücken
 - Kap.3 Dodier: Unterschiede zwischen Ausdruck a , $'a$, $''a$ und $ev(a)$
- „Objekte“
 - Symbole und symbolische Ausdrücke
 - Listen und Matrizen ($[...]$ und $[[...],..., [...]]$)
 - Funktionen ($\langle fname \rangle(\langle Parameter \rangle)$)
 - Mengen

Einführung in das CAS Maxima

Überblick über die Sprachelemente

- Wichtige automatisch zur Verfügung gestellte Funktionen
 - Termmanipulationen (solve, subst, ratsimp, factor, etc.)
 - Funktionsanwendungen auf Listen (apply, map)
 - Analysisoperatoren (diff, integrate, etc.)
 - Standardfunktionen (sin, cos, log, etc.)
 - Symbolische Konstante mit fester Wertbedeutung (%pi, %e, %i)
- Plotten
 - Definitions- und Wertebereiche, Plotmodi, Plotziele

Einführung in das CAS Maxima

Überblick über die Sprachelemente

- Programmieren von Kontrollflüssen

Blockbildung mit lokalen Variablen, Rückgabewert: letztes Element

Verzweigungen: `if ... then ... else ...`

Schleifen: `for ... thru <limit> do ...`
`for ... while <condition> do ...`
`for ... unless <condition> do ...`

Viele Möglichkeiten, die nicht alle in on-line-Hilfe oder Referenzmanual dokumentiert sind:

```
for variable: initial_value step increment thru limit do (...)  
for variable in list end_tests do (...)  
while condition do (...)  
unless condition do (...)  
do (...) /* Abbruch durch return(<Rückgabewert>) */
```

Funktionen können in .mac-Dateien ausgelagert werden und mit Hilfe von `load` in die Arbeitsumgebung geladen werden.