

**Mathematische Methoden Physik I, WS06/07**  
**Kompaktkurs Mathematica, 21. 12. 2006**

Astronomisches Rechen-Institut,  
Zentrum für Astronomie, Universität Heidelberg  
Mönchhofstr. 12-14, 69120 Heidelberg  
E-Mail: [spurzem@ari.uni-heidelberg.de](mailto:spurzem@ari.uni-heidelberg.de)  
Tel.: 06221 – 541830  
<http://www.ari.uni-heidelberg.de/mitarbeiter/spurzem/>

## Contents

<b>1</b>	<b>Symbolische Algebra Programme</b>	<b>1</b>
<b>2</b>	<b>Mathematica</b>	<b>1</b>
<b>3</b>	<b>Erste Versuche</b>	<b>2</b>
<b>4</b>	<b>Listen, Vektoren und Matrizen</b>	<b>4</b>

## 1 Symbolische Algebra Programme

- MAPLE
- MATHEMATICA
- MATLAB
- ...

## 2 Mathematica

Mathematica ist eines von mehreren symbolischen Algebra-Software Paketen (Computeralgebra). Es wird an den Rechnern des Uni-Rechenzentrums Heidelberg (URZ) bereitgestellt. Die Software kann auch numerische Berechnungen durchführen, mit beliebig einstellbarer Genauigkeit, aber in der Regel nicht schnell. Außerdem kommt Mathematica ebenso wie Konkurrenzprodukte mit einem graphischen Interface. Mathematica wird unter Unix durch **math** aufgerufen, um im Zeilenmodus im Terminal Kommandos einzugeben (wie in gnuplot zum Beispiel, veraltete Benutzungsweise...) Der Aufruf von **mathematica** dagegen läßt eine Anzahl von X-Windows (Fenstern) erscheinen (graphisches User-Interface, GUI). In diesen Fenstern gibt es Kommando-Fenster, Hilfe-Fenster und anderes.

**Wichtig:** Zur Benutzung von mathematica sollte die **NumLock** Taste abgeschaltet sein. Eingaben in einem mathematica Kommandofenster werden mit **SHIFT+ENTER** abgeschickt zur Auswertung.

<http://mathworld.wolfram.com/> ist ein Portal für Informationen aller Art um und über mathematica. Wollen Sie Informationen über eine bestimmte Funktion o.ä. (z.B. Bessel Funktionen),

so führt es meist schneller zum Ziel, den Begriff zu googeln, und dann den Link von wolfram.com herauszusuchen.

mathematica ist so umfangreich, daß ein systematisches Lernen nur am Anfang sinnvoll ist. Später sollten Sie durch immer komplizierter werdende Beispiele lernen, die Sie dann schrittweise abwandeln.

Sie können Programmcodes zu mathematica als Datei abspeichern, laut Konventionen haben solche Dateien die Endung **.nb** (Mathematica Notebook). Mathematica Notebooks können Sie im mathematica Programm einlesen (öffnen, Pull-Down Menü File bzw. Datei); dann die Auswertungsschritte durchführen. Die Ergebnisse inkl. Graphiken können Sie entweder direkt drucken oder beim Drucken als Datei abspeichern.

Für Computeralgebra Programme gilt wie bei jeder anderen Software, daß eine Plausibilitätskontrolle der Ergebnisse und die Prüfung von Spezialfällen **dringend** geboten sind. Vertrauen Sie den Ergebnissen niemals blindlings. Es können sowohl Bedienungsfehler vorkommen, als auch Probleme der Software, die nicht zu Ihren Lasten gehen...

### **3 Erste Versuche**

```
Solve[x^3-10x^2+21x-10==0,x]
```

Durch Drücken von **SHIFT+ENTER** By pressing **SHIFT+ENTER** wird der Ausdruck ausgewertet. Andere Beispiele:

```
Plot[Sin[x], {x, 0, Pi}]
```

```
Integrate[x^n, {x, 0, 1}]
```

```
Solve[x^2+x+1==0, x]//N
```

Das letzte Beispiel löst die quadratische Gleichung analytisch, und berechnet danach numerische Werte.

Mit

```
n=2
```

```
Clear[n]
```

können Werte an Variabke zugewiesen werden. Nach **n=2** wird jedes Vorkommen vonm  $n$  automatisch durch den Zahlenwert 2 ersetzt. Erst wenn durch **Clear[n]** die Variable wieder freigegeben wird, unterbleibt dies. (Vorsicht: Fehlerquelle!).

Das “=” Zeichen wird für Variablenzuweisungen verwendet, während “==” mathematische Gleichungen definiert.

Hier ist schon mal zur Einstimmung gleich ein komplizierteres Beispiel, die Lösung eines nichtlinearen gekoppelten DGL-Systems erster Ordnung (Volterra-Lotka System).

```
In[1]= NDSolve[{u'[t]==u[t](1-v[t]),v'[t]==0.5 v[t](u[t]-1),u[0]==1,v[0]==3},  
              {u,v},{t,0,40}]
```

```
Out[1]= {u -> InterpolatingFunction[{{0., 40.}}, <>],  
         v -> InterpolatingFunction[{{0., 40.}}, <>]}
```

```
In[2]= Plot[{Evaluate[u[t] /. %1], Evaluate[v[t] /. %1]}, {t,0,40}]
```

[...]

```
In[nm] = ParametricPlot[ {{Evaluate[{u[t], v[t]} /. %1]},  
                          {Evaluate[{u[t], v[t]} /. %2]}, {Evaluate[{u[t], v[t]} /.  
%3]},  
                          {Evaluate[{u[t], v[t]} /. %4]}}], {t,0,40}]
```

Erklärung: das `NDSolve` Kommando erzeugt ein Objekt, welches die numerisch approximierete Lösung der DGL enthält (durch eine interpolierende Funktion). Bevor wir dies plotten können, muß es ausgewertet werden. Während `Plot` einen Standard-Funktionsplot erzeugt, kann `Parametric Plot` zum Plotten zweier Funktionen mit dem Parameter  $t$  verwendet werden.

Jede Mathematica Eingabe wird mit `In[n]=` protokolliert, und die Ausgabe mit `Out[n]`. Durch `Out[n]` oder `%n` kann man auf ein vorhergehendes Resultat zurückgreifen, wobei  $n$  für eine ganze Zahl steht. Z.B. steht `%10` für das in `Out[10]` = protokollierte Ergebnis.

**Aufgabe:** Lösen Sie die Newton'sche Bewegungsgleichung für einen Kometen im Schwerefeld der Sonne mit mathematica. Setzen Sie dazu  $G = M = 1$  ( $G$ : Gravitationskonstante,  $M$  Masse der Sonne). Verwenden Sie als Anfangsbedingungen  $r_0 = 1$  und wahlweise Werte aus dem Bereich  $0.1 < v_0 < 2$ .

## 4 Listen, Vektoren und Matrizen

Erzeugen einer 2x2 matrix oder eines Vektors mit 6 Elementen:

```
M={{ 0,1 },{1,2}}
```

```
v={1,4,7,6,5,3}
```

Herauslesen eines Elementes des Vektors:

```
v[[3]]
```

Erzeugen einer 10x10 Matrix, deren Elemente durch die Formel gegeben sind:

```
M = Table[1/(i+j-1), {i, 10}, {j, 10}]
```

Bestimmen der Eigenwerte der Matrix M:

```
Eigenvalues[M]
```

Bestimmung der Eigenvektoren von M:

```
Eigenvectors[M] // N
```

```
Inverse[M]
```

```
M.Inverse[M]
```

```
M.v
```

```
v.v
```