Computer-Algebra-System MATHEMATICA

Sigrun Reimitz

Euch erwartet:

- Stephen Wolfram
- Einführung in *Mathematica*
- Literatur
- Resümee

Stephen Wolfram

- geboren 20. August 1959 in London
- mit 15 Jahren erste wissenschaftliche Publikation
- mit 20 Jahren Doktor der theoretischen Physik
- Ende 1986 Beginn mit Entwicklung von Mathematica
- 1. Version kam 1988 auf den Markt
- erfolgreiche Vermarktung durch „Wolfram Research“ (1987)
- noch heute ist S. Wolfram verantwortlich für das Grunddesign

Mathematica 4.1.
Präsentation
Kamele und Dromedare

Mathematica

■ Kurze Einführung

In[1]:= (5^2 - 1 * 30) / -5

Out[1]= 1

- zweidimensional mithilfe Tastaturkürzel oder Palletten (Formeleditor)

In[2]:= $\frac{x^2 + a * b}{c}$

Out[2]= $\frac{x^2 + a\, b}{c}$

- FullForm: Schreibweise der M-Funktionen beachten

In[3]:= Divide[Plus[Power[x, 2], Times[a, b]], c]

Out[3]= $\frac{x^2 + a\, b}{c}$

In[4]:= FullForm[%]

Out[4]//FullForm=

Times[Power[c, -1], Plus[Times[a, b], Power[x, 2]]]

In[5]:= ?Times

x*y*z or x y z represents a product of terms. More...

■ Exakte Arithmetik

Mathematica rechnet mit beliebig großen Zahlen:

In[6]:= 2^{150}

Out[6]= 1427247692705959881058285969449495136382746624

Vermeidung von Rundungsfehlern: Ersetzung erfolgt, wenn Muster gefunden wurde. (siehe Mustererkennung)

In[7]:= 2 * Pi

Out[7]= $2\,\pi$

Einen Näherungswert berechnet die Funktion N. Per default ist das Ergebnis auf 6 Stellen genau. Die Anzahl der beeinflusst.

```
In[8]:= N[Pi, 30]
```

Out[8]= 3.14159265358979323846264338328

Teiler bestimmen, Primzahlenerkennung (heute: 9.808.358 Stellen, Euler 1772)

```
In[9]:= FactorInteger[12]
```

Out[9]= $\begin{pmatrix} 2 & 2 \\ 3 & 1 \end{pmatrix}$

In[10]:= `PrimeQ[`2^{31}` - 1]`

Out[10]= True

■ Symbolische Algebra

Mathematica rechnet mit symbolischen Ausdrücken:

```
In[11]:= Sqrt[z]^2 (3*x - y - 2*x) (y + x) z^-1
```

Out[11]= $(x - y)(x + y)$

■ Vereinfachung

Umformungen lassen sich steuern z. B. durch
- Simplify sucht einfache Form des Ausdruck

In[12]:= $\frac{a}{(a-b)(a-c)} + \frac{b}{(b-a)(b-c)} + \frac{c}{(c-a)(c-b)}$ `// Simplify`

Out[12]= 0

■ Lösungsansatz

```
In[13]:= PolynomialLCM[(a - b) (a - c), (b - a) (b - c), (c - a) (c - b)]
```

Out[13]= $(a - b)(a - c)(b - c)$

■ Kurvendiskussion

In[14]:= `f[x_] =` $\frac{x^2}{x^2 - 4}$`;`

Nullstellen, Polstellen und Grenzwert:

```
In[15]:= Solve[f[x] == 0, x]
```

Out[15]= $\{\{x \to 0\}, \{x \to 0\}\}$

In[16]:=
```
Solve[Denominator[f[x]] == 0, x]
```

Out[16]= $\{\{x \to -2\}, \{x \to 2\}\}$

In[17]:=
```
Limit[f[x], x -> ∞]
```

Out[17]= 1

1. und 2. Ableitung bestimmen:

In[18]:=
```
f1[x_] = f'[x] // Simplify
```

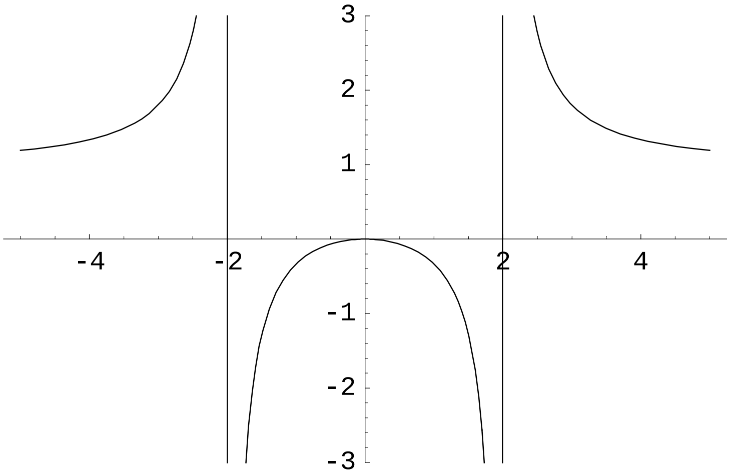
Out[18]= $-\frac{8x}{(x^2-4)^2}$

In[19]:=
```
f2[x_] = f''[x] // Simplify
```

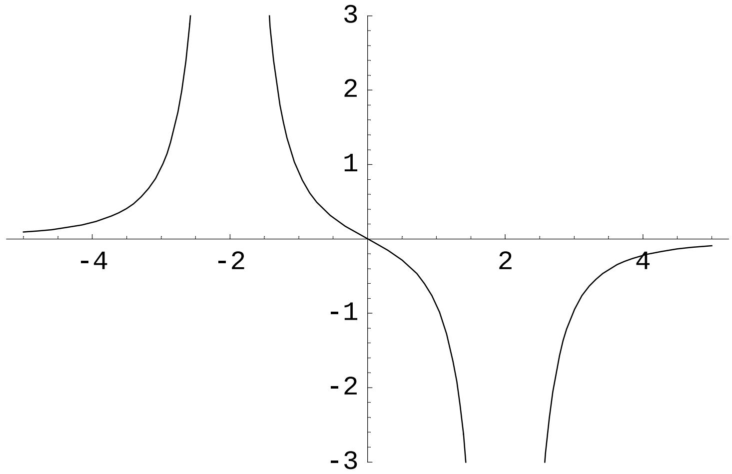
Out[19]= $\frac{8(3x^2+4)}{(x^2-4)^3}$

■ Graphische Darstellung

In[20]:=
```
Plot[f[x], {x, -5, 5},
        PlotRange -> {-3, 3}];
```

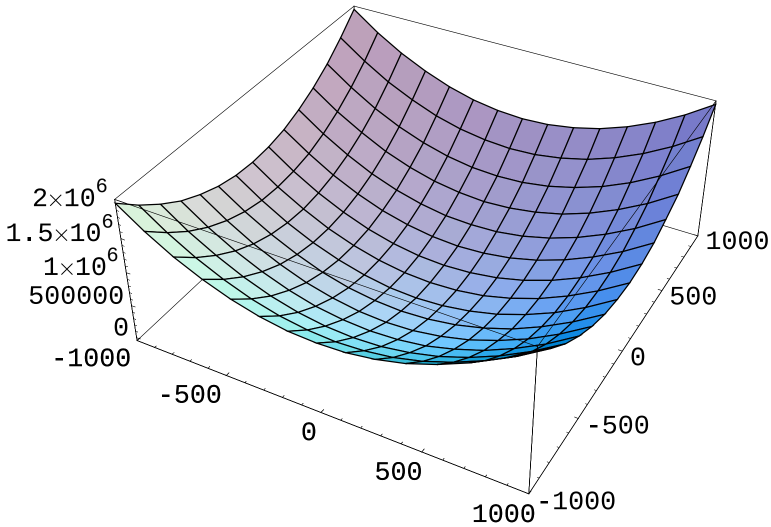


```
In[21]:= Plot[f1[x], {x, -5, 5},
           PlotRange -> {-3, 3}];
```



■ Dreidimensionale Darstellung:

In[22]:= Plot3D[$x^2 + y^2$, {x, -1000, 1000}, {y, -1000, 1000}];



■ Polynome und Polynomfaktorisierung

Expand[Ausdruck]: Produkte und Potenzen werden ausmultiplizieren
Factor[Ausdruck]: versucht, den Ausdruck in Faktoren zu zerlegen

In[23]:= Expand[$(x + y + z)^3$]

Out[23]= $x^3 + 3\, y\, x^2 + 3\, z\, x^2 + 3\, y^2\, x + 3\, z^2\, x + 6\, y\, z\, x + y^3 + z^3 + 3\, y\, z^2 + 3\, y^2\, z$

```
In[24]:= Factor[%]
```

Out[24]= $(x+y+z)^3$

Faktorisierung hängt vom betrachteten Ring ab:

```
In[25]:= Factor[x^6 + x^2 + 1]
```

Out[25]= x^6+x^2+1

... ist aber zerlegbar im Polynomring modulo 13

```
In[26]:= Factor[x^6 + x^2 + 1, Modulus -> 13]
```

Out[26]= $(x^2+6)(x^4+7x^2+11)$

Gleichungen und Gleichungssysteme

Solve: Löse das Gleichungssystem nach x, y, ... duch Eliminieren von a,b, ...

```
In[27]:= Solve[x^2 - 3*x - 1 == 0, x]
```

Out[27]= $\left\{\left\{x\to\frac{1}{2}\left(3-\sqrt{13}\right)\right\},\left\{x\to\frac{1}{2}\left(3+\sqrt{13}\right)\right\}\right\}$

Gleichungssystem mit zwei Gleichungen und zwei Variablen: Hier möchten wir nach x und y auflösen:

```
In[28]:= Solve[{2 x + 3 y == 5, 3 x + 4 y == 11}, {x, y}]
```

Out[28]= $\{\{x\to 13,\ y\to -7\}\}$

Zwei Gleichungen mit 3 Variablen:

```
In[29]:= Solve[{y - 3 a == x, 2 x - a == y}, {x, y}]
```

Out[29]= $\{\{x\to 4a,\ y\to 7a\}\}$

Eliminierung von a: (Folie)

```
In[30]:= Solve[{y - 3 a == x, 2 x - a == y}, {x}, {a}]
```

Out[30]= $\left\{\left\{x\to\frac{4y}{7}\right\}\right\}$

Lineare Optimierung mit Ungleichsystemen

Prof. Gerhardt

In[31]:= **ConstrainedMin[5 x + 11 y, {50 x + 100 y ≥ 1000, 4 x + 3 y ≤ 60, 80 x + 100 y ≤ 1600}, {x, y}]**

Out[31]= $\{104,\ \{x\to 12,\ y\to 4\}\}$

Syntax ab Version 5.0
NMinimize[{5x+11y, {50x+100y≥1000, 4x+3y≤60, 80x+100y≤1600}},{x,y}]

■ Listen, Vectoren und Matrizen

Mengen, Vectoren und Matrizen werden in *Mathematica* als Listen dargestelt. In einer Liste stehen die Elemente Klammern. Die Elemente können durch einen Index angesprochen werden:

```
In[32]:= v1 = {a, b, c} // MatrixForm
```

Out[32]//MatrixForm=

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

```
In[33]:= m1 = {{a, b}, {c, d}, {e, f}}
```

Out[33]= $\begin{pmatrix} a & b \\ c & d \\ e & f \end{pmatrix}$

```
In[34]:= m2 = {{1, 2, a}, {4, 5, 6}}
```

Out[34]= $\begin{pmatrix} 1 & 2 & a \\ 4 & 5 & 6 \end{pmatrix}$

Operationen auf Matrizen:

```
In[35]:= m3 = m1.m2
```

Out[35]= $\begin{pmatrix} a+4b & 2a+5b & a^2+6b \\ c+4d & 2c+5d & ac+6d \\ e+4f & 2e+5f & ae+6f \end{pmatrix}$

```
In[36]:= m1[[3, 2]]
```

Out[36]= f

```
In[37]:= Transpose[m3]
```

Out[37]= $\begin{pmatrix} a+4b & c+4d & e+4f \\ 2a+5b & 2c+5d & 2e+5f \\ a^2+6b & ac+6d & ae+6f \end{pmatrix}$

■ Programmieren mit Mathematica

Variablen sind global vereinbart,
eine lokale Verwendung ist über folgende Syntax möglich: Module[{a, b, ...}, proc]:

```
In[38]:= ?Global`*
```

Global`

a b c d e f f1 f2 m1 m2 m3 v1 x y z

```
In[39]:= Remove["Global`*"]
```

■ Funktionen programmieren

Parameter für Funktionen werden mit einem Unterstrich vereinbart. Beim Aufruf der Funktion fällt dieser weg. Zuweisu
Aufruf der Funktion. Zusammenfassung von mehreren Anweisungen mit Semikolon.

In[40]:= $f[x\_] := x^2;$
$g[x\_, y\_] := \frac{x}{1 + y^2};$
$f[t] - g[8, v]$

Out[42]= $t^2 - \frac{8}{v^2 + 1}$

Rekursionen entwickeln:

```
In[43]:= nextPrime[n_] := n + 1 /; PrimeQ[n + 1]
         nextPrime[n_] := nextPrime[n + 1]
```

und ausführen

```
In[45]:= nextPrime[1234567]
```

Out[45]= 1234577

```
In[46]:= PrimeQ[%]
```

Out[46]= True

■ Verzweigungen, Schleifen

Schleifen selbst liefern kein Ergebnis : Zuweisung an Variable nötig

```
In[47]:= f[x_] := If[x > 0, "positiv", "nichtpositiv", "unklar"]
```

```
In[48]:= f[-6]
```

Out[48]= nichtpositiv

```
In[49]:= f[w]
```

Out[49]= unklar

Zählschleife Do[anweisung, iterator{k, k_1 , k_2 , k_3}].

```
In[50]:= y = 1; Do[y = k * y, {k, 1, 100}]; y
```

Out[50]= 93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156\
18286253697920827223758251185210916864000000000000000000000000

```
In[51]:= 100 !
```

Out[51]= 93326215443944152681699238856266700490715968264381621468592963895217599993229915608941463976156\
18286253697920827223758251185210916864000000000000000000000000

Mustererkennung

Die Arbeitsweise von *Mathematica* beruht auf Mustererkennung:
eingebaute Regel für Sin[Pi], Sin[x] bleibt unverändert

```
In[52]:= Sin[Pi]

Out[52]= 0

In[53]:= Sin[x]

Out[53]= sin(x)
```

Sinus von irgendetwas: Sin [x_]
Die Position bestimmter Muster ist bestimmbar:

```
In[54]:= ausdruck = 1 + Sin[3] - Sin[x] + Sin[Pi]; Position[ausdruck, Sin[_]]

Out[54]= {{2}, {3, 2}}

In[55]:= Plus[1, Sin[3], Times[-1, Sin[x]]]

Out[55]= -sin(x) + sin(3) + 1
```

Mustererkennung für Programmierung nutzen:
Bsp. Regeln für die Differentation
- Potenzregel

```
In[56]:= meineDiff[f_ + g_, x_] := meineDiff[f, x] + meineDiff[g, x]

In[57]:= meineDiff[c_ f_, x_] := c * meineDiff[f, x] /; FreeQ[c, x]

In[58]:= meineDiff[c_, x_] := 0 /; FreeQ[c, x]

In[59]:= meineDiff[x_^n_., x_] := n * x^(n-1) /; FreeQ[n, x]
```

c ist ein Konstante, wenn c x nicht enthält: f(x)=5
irgendetwas hoch irgendwas (Punkt schließt 1. Potenz ein)

```
In[60]:= meineDiff[2 x^3 + 7 x^2 + 1, x]

Out[60]= 6 x^2 + 14 x
```

Rememberprogrammierung

Verringerung der Komplexität rekursiver Programme am Beispiel der Fibonaccizahlen
$F_n = F_{n-1} + F_{n-2}$
mit $F_0 = 0$ und $F_1 = 1$

```
In[61]:= fib[0] = 0;
         fib[1] = 1;
         fib[n_] := fib[n - 1] + fib[n - 2]
```

```
Timing[fib[30]]
```

{7.75 Second, 832040}

Jedes berechnete Ergebnis wird hinterlegt durch fib[n_] := **fib[n] =** fib[n-1]+fib[n-2]

```
In[64]:= fib1[0] = 0;
         fib1[1] = 1;
         fib1[n_] := fib1[n] = fib1[n - 1] + fib1[n - 2];
         fib1[30];
         ? fib1
```

```
Global`fib1
```

```
fib1[0] = 0

fib1[1] = 1

fib1[2] = 1

fib1[3] = 2

fib1[4] = 3

fib1[5] = 5

fib1[6] = 8

fib1[7] = 13

fib1[8] = 21

fib1[9] = 34

fib1[10] = 55

fib1[11] = 89

fib1[12] = 144

fib1[13] = 233

fib1[14] = 377

fib1[15] = 610

fib1[16] = 987

fib1[17] = 1597

fib1[18] = 2584

fib1[19] = 4181

fib1[20] = 6765

fib1[21] = 10946

fib1[22] = 17711

fib1[23] = 28657

fib1[24] = 46368

fib1[25] = 75025

fib1[26] = 121393

fib1[27] = 196418

fib1[28] = 317811

fib1[29] = 514229

fib1[30] = 832040

fib1[n_] := fib1[n] = fib1[n - 1] + fib1[n - 2]
```

2.3.2 Ein LOP - Beispiel

Ein Händler muss 1000 Steine von einer Stadt in eine andere Stadt transportieren.
Dazu kann er Kamele und Dromedare mieten.
Ein Dromedar kostet am Tag 5$, ein Kamel am Tag 11$ Leihgebühr.
Ein Dromedar trägt 50 Steine, ein Kamel 100 Steine.
Ein Dromedar braucht für den Transport 4 Ballen Heu, ein Kamel 3 Ballen.
Ein Dromedar braucht 80 Liter, ein Kamel 100 Liter Wasser.
Verfügbar sind für die Reise 60 Ballen Heu und 1600 Liter Wasser.
Wie viele Dromedare und Kamele muss der Händler für den Transport leihen, wenn er die geringsten Leihgebühren zahlen will?

Bezeichnung: x_1 Anzahl der Dromedare,
x_2 Anzahl der Kamele.

Modell:

$$\min\{5x_1 + 11\,x_2 = z\}, \text{ wobei } S = \{\, x \in R^2 \mid \begin{array}{rcl} 50x_1 + 100\,x_2 & \geq & 1000, \\ 4x_1 + 3\,x_2 & \leq & 60, \\ 80x_1 + 100\,x_2 & \leq & 1600, \\ x_1,\ x_2 & \geq & 0 \,\} \end{array}$$

Restriktionen: Steine,
Heu,
Wasser,
Nichtnegativitätsbedingung.

Teilaufgaben:

1. Stellen Sie das mathematische Modell dieses LOP's auf.
2. Lösen Sie die Aufgabe grafisch.
3. Berechnen Sie die extremalen Punkte und berechnen Sie für jeden extremalen Punkt den Wert der Zielfunktion.
   Geben Sie S, Θ, Θ_x, Θ_μ an.

4. Wie lautet die optimale Lösung?

Ersetzung einer Variablen

$$\begin{array}{rcrcrcl}
x & - & y & + & 3a & = 0 & \text{(I)} \\
2x & - & y & - & a & = 0 & +3\text{(II)} \\
\hline
7x & - & 4y & & & = 0 & \text{(III)}
\end{array}$$

Literatur

– **Bücher**
  - **„Das Mathematica Buch- Die offizielle Dokumentation“**
    – Stephen Wolfram 1499 S.
  - **„Mathematica – Einführung, Anwendung, Referenz“**
    – Michael Kofler, Hans-Gert Gräbe
  - **„Erste Schritte mit Mathematica“**
    – Werner Burkhardt

– **Web-Links**
  - http://documents.wolfram.com/v4-de/index.html
  - www.mathematik.uni-kassel.de/~rascha/Mathematica_Kurs.html

Resümee

– einfache, intuitive Bedienung
– mächtiges Hilfsmittel
– ersetzt keine Mathematikveranstaltung
– ...und jetzt Eure Meinung

Vielen Dank für Eure Mitarbeit!