Kapitel 8: Funktionen in einer Variablen

Für Funktionen in einer Variablen werden folgende elementaren Probleme gelöst: Nullstellen von Funktionen erhält man über den **solve** - bzw. **fsolve** -Befehl, die Linearfaktorenzerlegung erfolgt mit **factor** und eine Partialbruchzerlegung von gebrochenrationalen Funktionen mit **convert**. Die Bestimmung von Extremwerten, Wendepunkte und Asymptoten ist im Abschnitt über die Kurvendiskussion zusammengefasst. Das Lösen der Einzelprobleme erfolgt hierbei im Wesentlichen durch **solve**, **diff**, **simplify** sowie **plot**. Speziell für die Entwicklung einer Funktion in eine Taylorreihe benötigt man den **taylor**-Befehl.

8.1 Bestimmung von Nullstellen

fsolve	**worksheet**
Problem	Gesucht sind Näherungen für die Nullstellen einer Funktion f(x): f(x)=0
Befehl	**fsolve**(f(x)=0, x);
Parameter	*f(x):* Funktionsausdruck *x:* Variable der Funktion
Beispiel	$\sqrt{x} - 4\,x^2 = 0$ > **f(x) := sqrt(x) - 4*x^2 :** > **fsolve(f(x)=0, x);** 0. > **fsolve(f(x)=0, x, x=0.1..2);** .3968502630
Optionale Parameter	> fsolve(f(x)=0, x, x=x0..x1); x=x0..x1 gibt das Intervall an, in dem eine Nullstelle näherungsweise berechnet wird. > fsolve(f(x)=0, x, complex); berechnet auch komplexe Lösungen.
Hinweise	Ist f(x) ein Polynom vom Grade n, dann werden mit der Option *complex* alle Nullstellen (reelle als auch komplexe) des Polynoms f(x) näherungsweise bestimmt.
Siehe auch	**solve**; → Näherungsweises Lösen einer Gleichung.

8.2 Linearfaktorzerlegung von Polynomen

factor	**worksheet**
Problem	Gesucht ist eine Zerlegung des Polynoms $f(x) = a_n x^n + a_{n-1} x^{(n-1)} + \ldots + a_1 x + a_0$ in Linearfaktoren der Form $a_n x^n + a_{n-1} x^{(n-1)} + \ldots + a_1 x + a_0 = a_n (x - x_1)(x - x_2) \quad (x - x_n)$
Befehl	**`factor`**(f(x));
Parameter	*f(x):* Polynom vom Grade n

Beispiel	$f(x) = 7x^6 - 17x^5 + 20x^4 - 20x^3 + 13x^2 - 3x$
	 `> factor(f(x));`
	 $x(7x-3)(x^2+1)(x-1)^2$
	 `> factor(f(x), complex);`
	 $7.\,(x + 1.\,I)\,x\,(x - 1.\,I)\,(x - .4285714286)\,(x - 1.)^2$

Hinweise	Der **factor**-Befehl liefert falls möglich alle reellen Nullstellen und stellt das Polynom in diesen Nullstellen dar. Mit der Option *complex* werden auch die komplexen Nullstellen näherungsweise bestimmt und man erhält eine vollständige Zerlegung in Linearfaktoren.
Siehe auch	**`fsolve`**.

8.3 Partialbruchzerlegung gebrochenrationaler Funktionen

convert **parfrac**	**worksheet**
Problem	Gesucht ist eine Partialbruchzerlegung der gebrochenrationalen Funktion $(a_n x^n + a_{n-1} x^{(n-1)} + \ldots + a_1 x + a_0) \,/\, (b_m x^m + b_{m-1} x^{(m-1)} + \ldots + b_1 x + b_0)$
Befehl	**`convert`**(f(x), **parfrac**, x);
Parameter	*f(x):* Gebrochenrationale Funktion *x:* Unabhängige Variable der Funktion
Beispiel	$$\mathrm{f}(x) = \frac{x^6 - 2x^5 + x^4 + 4x + 1}{x^4 - 2x^3 + 2x - 1}$$ `> f(x):=(x^6-2*x^5+x^4+4*x+1) / (x^4-2*x^3+2*x-1):` `> convert(f(x), parfrac,x);` $$x^2 + 1 - \frac{1}{8}\,\frac{1}{x+1} + \frac{\frac{5}{2}}{(x-1)^3} + \frac{\frac{3}{4}}{(x-1)^2} + \frac{\frac{1}{8}}{x-1}$$
Hinweise	-
Siehe auch	**`fsolve`**, **`factor`**.

8.4 Kurvendiskussion

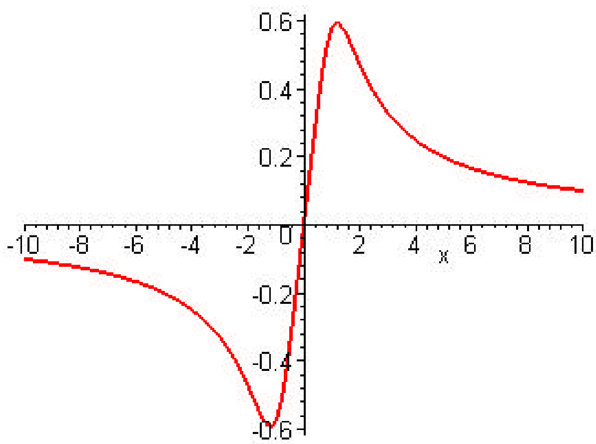
<table>
<tr><td></td><td>worksheet</td></tr>
<tr><td>Problem</td><td>Kurvendiskussion einer Funktion f(x) in einer Variablen x

(1) Graph der Funktion
(2) Symmetrie
(3) Nullstellen
(4) Lokale Extrema
(5) Wendepunkte
(6) Verhalten im Unendlichen</td></tr>
<tr><td>Befehl</td><td>Maple-Befehlsfolge</td></tr>
<tr><td>Parameter</td><td>f(x): Ausdruck in der Variablen x
x: Unabhängige Variable</td></tr>
<tr><td>Beispiel</td><td>

$$f(x) = \frac{x}{\sqrt{x^4+2}}$$

`> f:=x->x/sqrt(x^4+2):`

(1) Funktionsgraph: **plot**-Befehl
`> plot(f(x), x=-10..10);`



</td></tr>
</table>

(2) Symmetrie: f(-x)=f(x) oder f(-x)=-f(x): **simplify**-Befehl

```
> simplify(f(x)/f(-x), symbolic);
```

$$-1$$

Die Funktion ist punktsymmetrisch zum Ursprung.

(3) Nullstellen: **solve**-Befehl

```
> solve(f(x)=0,x);
```

$$0$$

(4) Lokale Extrema: Erste Ableitung gleich Null, zweite ungleich Null.
Bestimmung der relevanten Ableitungen mit dem **diff**-Befehl.

```
> fs:=simplify(diff(f(x), x));
> fss:=simplify(diff(f(x), x$2));
> fsss:=simplify(diff(f(x), x$3));
```

$$fs := -\frac{x^4-2}{(x^4+2)^{(3/2)}}$$

$$fss := 2\,\frac{x^3\,(x^4-10)}{(x^4+2)^{(5/2)}}$$

$$fsss := -6\,\frac{x^2\,(x^8-28\,x^4+20)}{(x^4+2)^{(7/2)}}$$

Extrema: Nullstellen der ersten Ableitung: **solve**-Befehl

```
> e:=[solve(fs=0,x)];
```

$$e := [2^{(1/4)},\, I\,2^{(1/4)},\, -2^{(1/4)},\, -I\,2^{(1/4)}]$$

```
> evalf(e);
```

$$[\,1.189207115\,,\, 1.189207115\;I,\, -1.189207115\,,\, -1.189207115\;I\,]$$

Es gibt 2 reelle Kandidaten für lokale Extremwerte e[1] und e[3]. Ob diese Kandidaten auch Extremwerte darstellen entscheidet die 2. Ableitung

```
> subs(x=e[1],fss);
> evalf(%);
```

$$-\frac{1}{4}\,2^{(3/4)}\sqrt{4}$$

$$-.8408964155$$

Da zweite Ableitung negativ, liegt hier ein lokales Maximum vor. Der Funktionswert ist

```
> evalf(f(e[1]));
```

$$.5946035575$$

```
> subs(x=e[3],fss);
> evalf(%);
```

$$\frac{1}{4}2^{(3/4)}\sqrt{4}$$

$$.8408964155$$

Da zweite Ableitung positiv, liegt hier ein lokales Minimum vor.

(5) Wendepunkte: Zweite Ableitung gleich Null, dritte ungleich Null

```
> w:=[solve(fss=0,x)];
```

$$w := [0, 0, 0, 10^{(1/4)}, I\,10^{(1/4)}, -10^{(1/4)}, -I\,10^{(1/4)}]$$

```
> evalf(w);
```

$[0., 0., 0., 1.778279410, 1.778279410\ I, -1.778279410, -1.778279410$

Es gibt 3 reelle Kandidaten für Wendepunkte w[1], w[4] und w[6]. Ob diese Kandidaten auch Wendepunkte darstellen entscheidet die 3. Ableitung

```
> subs(x=w[1],fsss);evalf(%);
```

$$0$$

$$0.$$

Da die dritte Ableitung Null, liegt für den Wert x=0 kein Wendepunkt vor. In Frage kommen nun noch die Werte 1.778279410 bzw. -1.778279410:

```
> subs(x=w[4],-fsss);evalf(%);
```

$$-\frac{5}{108}\sqrt{10}\sqrt{12}$$

$$-.5071505162$$

```
> subs(x=w[6],fsss);evalf(%);
```

$$\frac{5}{108}\sqrt{10}\sqrt{12}$$

$$.5071505162$$

(6) Asymptotisches Verhalten: Das asymptotische Verhalten bestimmt man mit dem **asympt**-Befehl

```
> asympt(f(x),x);
```

$$\frac{1}{x} + \mathrm{O}\left(\frac{1}{x^5}\right)$$

Hinweise	Falls der **solve**-Befehl keine befriedigenden Ergebnisse liefert, sollte der **fsolve**-Befehl verwendet werden, der eine Näherungslösung der Nullstellen bestimmt. Mit **simplify** werden die Ausdrücke vereinfacht.
Siehe auch	**subs**, **fsolve**, **simplify**.

8.5 Taylorentwicklung einer Funktion

taylor	**worksheet**
Problem	Gesucht ist die Taylorentwicklung der Ordnung N für eine Funktion f(x) mit einer Variablen x $$f(x) = f(x_0) + \left(\frac{\partial}{\partial x} f(x_0)\right)(x - x_0) + \ldots + \frac{1}{N!}\left(\frac{\partial}{\partial x}\right)^N f(x_0)\,(x - x_0)^N$$
Befehl	**taylor**(f(x), x=x0, N+1);
Parameter	*f(x):* Funktionsausdruck *x=x0:* Entwicklungspunkt *N:* Ordnung der Taylorreihe
Beispiel	$$f(x) = e^x$$ an der Stelle $x_0 = 0$ bis zur Ordnung 5. `> f:=x->exp(x):` `> taylor(f(x), x=0, 6);` $$1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{24}x^4 + \frac{1}{120}x^5 + O(x^6)$$ `> p:= convert(%,polynom);` $$p := 1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{24}x^4 + \frac{1}{120}x^5$$ `> plot([f(x), p], x=-2..4,color=[red,blue]);`[1]
Hinweise	$O(x^6)$ bedeutet, dass Terme ab der Ordnung 6 abgeschnitten werden. Mit **convert** wird die Partialsumme in ein Polynom umgewandelt, welches dann z.B. mit dem **plot**-Befehl gezeichnet werden kann. Die allgemeine Taylorreihe mit einem allgemeinen Glied kann nicht durch den elementaren Befehlssatz von Maple bestimmt werden.
Siehe auch	**convert**, **mtaylor**; → Konvergenz von Potenzreihen: Konvergenzradius → Fehlerrechnung.

[1] Aus Platzgründen wird auf die Ausgabe der Graphik verzichtet.