

5. Algebraische Umformungen

Bisher haben wir uns mit Arithmetik beschäftigt, und nun wollen wir zu Umformungen kommen. Die drei Grundbefehle für algebraische Umformungen sind

simplify(...) - Vereinfache den gegebenen Term
expand(...) - Ausmultiplizieren
normal(...) - Auf Hauptnenner bringen
normal(..., expanded) - Variante von normal()
bei der Zähler und Nenner
ausmultipliziert werden.

> `expand((x+y)^2);`

$$x^2 + 2xy + y^2$$

> `normal(1/(1+x)+1/(1-x));`

$$-\frac{2}{(1+x)(-1+x)}$$

> `normal(1/(1+x)+1/(1-x), expanded);`

$$-\frac{2}{-1+x^2}$$

Als letztes Beispiel wollen wir einen Ausdruck in zwei Variablen a und b vereinfachen.

> `A:=((b*a)^19*a^12)/(b^12*sqrt(a^3)^13)-sqrt(a)^10/b^7;`

$$a := a$$

$$b := b$$

$$A := \frac{b^7 a^{31}}{(a^3)^{(13/2)}} - \frac{a^5}{b^7}$$

> `simplify(%);`

$$-\frac{a^5 (-b^{14} a^8 + \sqrt{a^3})}{\sqrt{a^3} b^7}$$

Verwunderlich ist hier, dass die in Zähler und Nenner vorkommenden Potenzen von a nicht weiter ausgekürzt werden. Dies liegt daran das Maple automatisch nur sichere, unzweideutige Umformungen durchführt. Ist aber a negativ oder komplex so ist die Wurzel im Nenner problematisch. Durch den Zusatz assuming ... kann Maple mitgeteilt werden, dass wir uns a als eine positive, reelle Zahl denken.

> `simplify(A) assuming a>0;`

$$\frac{a^5 (b^{14} a^{(13/2)} - 1)}{b^7}$$

6. Mehr zu Variablen und %

Das Zeichen % steht für das Ergebnis der letzten Rechnung. Man

muss immer aufpassen, dass diese letzte Rechnung nicht unbedingt in der Zeile direkt über uns steht.

```
> 2;
```

2

```
> 3;
```

3

Geht man nun mit dem Cursor in die Zeile mit der 2 und führt diese durch Drücken der Eingabetaste erneut aus, so wird % zum Wert 2

```
> %;
```

2

und nicht etwa 3. Die Drei steht zwar direkt über dieser Zeile, war aber eben nicht die zuletzt durchgeführte Rechnung. Solange man noch am Herumprobieren ist und einige Zeilen häufig ändert muss man daher beim Einsatz von %,%%,%%% etwas vorsichtig sein.

Wenn man sich nicht mehr sicher ist ob und mit was eine Variable gerade belegt ist kann man den Maple Befehl `whattype()` verwenden. Beispielsweise

```
> x:=7;
```

$x := 7$

```
> y:=(a^2+a+1)*(b^2+1)*(b-1);
```

$y := (a^2 + a + 1)(b^2 + 1)(b - 1)$

```
> whattype(x);
```

integer

```
> whattype(a);
```

symbol

```
> whattype(y);
```

*

Der Typ '*' steht für ein Produkt. Mit dem Maple Befehl `op` kann man eine Aufzählung der beteiligten Faktoren erhalten

```
> op(y);
```

$a^2 + a + 1, b^2 + 1, b - 1$

Auf einzelne Einträge einer solchen Aufzählung kann man durch eine Indexnotation `A[i]` zugreifen, dies meint den *i*-ten Eintrag der Aufzählung *A*, etwa im Beispiel

```
> %[2];whattype(%[1]);
```

$b^2 + 1$

+

Alternativ kann man auch gleich einen spezifischen Operanden auswählen, durch

`op(index, Formel)`

wird Operand Nummer 'index' der Formel ausgewählt.

```
> op(2,y);
```

$b^2 + 1$

Das kann man auch iterieren und sich zu tiefer verschachtelten Teilausdrücken durchkämpfen

```
> op(op(y)[1])[2];
```

a

Dies kann man aber auch durch eine weitere Variante des indizierten op-Befehls erreichen.

```
> op([1,2],y);
```

a

Wenn man Teilausdrücke einer Formel für weitere Rechnungen braucht, kann man diese anstelle der op-Konstruktionen natürlich auch per Copy/Paste einsetzen. Was die angebrachte Methode ist, ist eine Geschmacksfrage. Wenn man dieselbe Rechnung mit kleinen Änderungen nochmal machen möchte empfehlen sich die op(s) da man andernfalls das Copy/Paste der Zwischenergebnisse wiederholen müsste.

Wenn man sich irgendwann völlig verfranzt hat und nicht mehr weiss welche Variablen welchen Wert haben, kann man mit dem Befehl 'restart' wieder zum Startzustand des Programs zurückkehren.

```
> restart;whattype(y);
```

symbol

7. Rechnen mit Polynomen

Polynome in einer oder mehrerer Unbekannten können als Formeln der entsprechenden Gestalt eingegeben werden, zum Beispiel

```
> p:=x^7-5*x^4+3*x^3+x^2-9;
```

$$p := x^7 - 5x^4 + 3x^3 + x^2 - 9$$

Um den Wert eines Polynoms an einer Stelle x auszuwerten gibt es verschiedene Möglichkeiten, am einfachsten ist der Befehl eval, für evaluate. Die Syntax ist

eval(Formel, Zuweisung)

```
> eval(p,x=2);
```

67

In den Koeffizienten eines Polynoms dürfen statt Zahlen auch beliebig komplizierte Ausdrücke stehen, etwa

```
> q:=sin(theta)*x^2+tan(theta)*x-1;
```

$$q := \sin(\theta)x^2 + \tan(\theta)x - 1$$

Bei eval() können auch mehrere Argumente mit Werten versorgt werden, man muss die Zuweisungen dann in {...} einschließen, etwa

```
> eval(q,{x=1,theta=Pi/3});
```

$$\frac{3\sqrt{3}}{2} - 1$$

Die arithmetischen Operatoren +,*,... sind auf beliebige Formeln also auch auf Polynome anwendbar. Um wieder Polynomform zu erhalten muss eventuell expand() verwendet werden.

```
> p^2+p-3;
```

$$(x^7 - 5x^4 + 3x^3 + x^2 - 9)^2 + x^7 - 5x^4 + 3x^3 + x^2 - 12$$

```
> expand(%);
```

$$69 - 47x^7 + 86x^4 - 51x^3 - 17x^2 + x^{14} - 10x^{11} + 6x^{10} + 2x^9 + 25x^8 - x^6 + 6x^5$$

Die einzelnen Terme des Polynoms treten dann oft in einer merkwürdigen Reihenfolge auf, sie können mit dem Maple Befehl

```
sort(Polynom)
```

nach fallenden Potenzen sortiert werden.

```
> sort(%);
```

$$x^{14} - 10x^{11} + 6x^{10} + 2x^9 + 25x^8 - 47x^7 - x^6 + 6x^5 + 86x^4 - 51x^3 - 17x^2 + 69$$

Ein weiteres Beispiel, diesmal mit freien Parametern

```
> q:=(x+a)^2*(x+b);
```

$$q := (x + a)^2 (x + b)$$

```
> expand(%);
```

$$x^3 + x^2 b + 2x^2 a + 2x a b + a^2 x + a^2 b$$

Hier wurden gleiche Potenzen von x nicht zusammengefasst da Maple nicht wissen kann das wir uns dies als ein Polynom in x denken wollen. Um die Polynomform herzustellen gibt es den Maple Befehl

```
collect(Formel in x, x)
```

wobei statt x auch jede andere Variable oder sogar ganze Formeln verwendet werden können.

```
> h:=collect(% , x);
```

$$h := x^3 + (2a + b)x^2 + (a^2 + 2ab)x + a^2 b$$

```
> eval(q, x=sin(theta));
```

$$(\sin(\theta) + a)^2 (\sin(\theta) + b)$$

```
> expand(%);
```

$$\sin(\theta)^3 + \sin(\theta)^2 b + 2 \sin(\theta)^2 a + 2 \sin(\theta) a b + a^2 \sin(\theta) + a^2 b$$

```
> collect(% , sin(theta));
```

$$\sin(\theta)^3 + (2a + b) \sin(\theta)^2 + (a^2 + 2ab) \sin(\theta) + a^2 b$$

Maple kennt die folgenden Grundbefehle für das Rechnen mit Polynomen

coeff(p,x,i) - i-ter Koeffizient wenn p ein Polynom in x ist

degree(p,x) - Grad von p als Polynom in x

quo(p,q,x) - Quotient von p/q als Polynome in x bei Division mit Rest.

rem(p,q,x) - Rest von p/q als Polynome in x bei Division mit Rest.

gcd(p,q) - GGT der Polynome p und q

lcm(p,q) - KGV der Polynome p,q

gcdex(p,q,x,f,g) - GGT der Polynome p,q in x. Die beiden anderen Argumente sind die Namen von Variablen in denen Polynome f,g mit $f*p+g*q=GGT$ gespeichert werden.

Um uns hierzu einige Beispiele anzuschauen verschaffen wir uns erst einmal ein komplizierteres Polynom indem wir x^2+1 in das obige Polynom $h(x)=x^3+(2a+b)x^2+(a^2+2ab)x+a^2b$ einsetzen

```
> q:=collect(eval(h, x=x^2+1), x);
```

$$q := x^6 + (3 + 2a + b)x^4 + (3 + a^2 + 2ab + 4a + 2b)x^2 + 1 + a^2b + a^2 + 2ab + 2a + b$$

> `p := x^3 - 7*x + 1;`

$$p := x^3 - 7x + 1$$

Grad und der Koeffizient von q bei x^4 sind dann

> `degree(q); coeff(q, x, 4);`

6

$$3 + 2a + b$$

Quotient und Rest bei Division von q durch p sind

> `quo(q, p, x); rem(q, p, x);`

$$x^3 + (10 + 2a + b)x - 1$$

$$(73 + a^2 + 2ab + 18a + 9b)x^2 + (-17 - 2a - b)x + 2 + a^2b + a^2 + 2ab + 2a + b$$

Test das es geklappt hat. Wir verwenden sort() diesmal in der Form

`sort(Polynomial, variable)`

damit sort() weiss nach welcher Variable sortiert werden soll.

> `sort(collect(%*p%, x), x);`

$$x^6 + (3 + 2a + b)x^4 + (3 + a^2 + 2ab + 4a + 2b)x^2 + 1 + a^2b + a^2 + 2ab + 2a + b$$

Der größte gemeinsame Teiler und das kleinste gemeinsame Vielfache der beiden Polynome sind

> `gcd(p, q);`

1

> `lcm(p, q);`

$$(x^3 - 7x + 1)$$

$$(x^6 + (3 + 2a + b)x^4 + (3 + a^2 + 2ab + 4a + 2b)x^2 + 1 + a^2b + a^2 + 2ab + 2a + b)$$

8. Ein Beispiel: Die Primärzerlegung

Wir erinnern uns an ein klein wenig lineare Algebra. Angenommen wir haben eine nxn Matrix A mit Minimalpolynom m, also $m(A)=0$. Schreiben wir dann $m=p*q$ als Produkt zweier teilerfremder Polynome, so zerfällt der K^n in die direkte Summe $K^n = \text{Kern}(p(A)) + \text{Kern}(q(A))$. Dies war leicht zu sehen, ist $f*p+g*q=1$ mit Polynomen f,g, so folgt für u im K^n

$$u = f(A)p(A)u + g(A)q(A)u$$

mit $f(A)p(A)u$ in $\text{Kern}(q(A))$ und $g(A)q(A)u$ in $\text{Kern}(p(A))$. Die Projektionen auf die beiden Faktoren sind also $(g*q)(A)$ und $(f*p)(A)$.

Wir wollen Maple verwenden ein konkretes Beispiel zu rechnen. Die Funktionen zum Rechnen mit Matrizen sind standardmäßig nicht aktiv, sie müssen explizit geladen werden. Maple fasst Sammlungen zusätzlicher Befehle in sogenannten Paketen zusammen. Jedes dieser Pakete hat einen Namen und kann mit dem Befehl

`with(Paketname)`

geladen werden. Das für lineare Algebra zuständige Paket heißt LinearAlgebra, also

```
> with(LinearAlgebra);
```

```
[&x, Add, Adjoint, BackwardSubstitute, BandMatrix, Basis, BezoutMatrix, BidiagonalForm, BilinearForm, CharacteristicMatrix, CharacteristicPolynomial, Column, ColumnDimension, ColumnOperation, ColumnSpace, CompanionMatrix, ConditionNumber, ConstantMatrix, ConstantVector, Copy, CreatePermutation, CrossProduct, DeleteColumn, DeleteRow, Determinant, Diagonal, DiagonalMatrix, Dimension, Dimensions, DotProduct, EigenConditionNumbers, Eigenvalues, Eigenvectors, Equal, ForwardSubstitute, FrobeniusForm, GaussianElimination, GenerateEquations, GenerateMatrix, Generic, GetResultDataType, GetResultShape, GivensRotationMatrix, GramSchmidt, HankelMatrix, HermiteForm, HermitianTranspose, HessenbergForm, HilbertMatrix, HouseholderMatrix, IdentityMatrix, IntersectionBasis, IsDefinite, IsOrthogonal, IsSimilar, IsUnitary, JordanBlockMatrix, JordanForm, KroneckerProduct, LA_Main, LUdecomposition, LeastSquares, LinearSolve, LyapunovSolve, Map, Map2, MatrixAdd, MatrixExponential, MatrixFunction, MatrixInverse, MatrixMatrixMultiply, MatrixNorm, MatrixPower, MatrixScalarMultiply, MatrixVectorMultiply, MinimalPolynomial, Minor, Modular, Multiply, NoUserValue, Norm, Normalize, NullSpace, OuterProductMatrix, Permanent, Pivot, PopovForm, QRdecomposition, RandomMatrix, RandomVector, Rank, RationalCanonicalForm, ReducedRowEchelonForm, Row, RowDimension, RowOperation, RowSpace, ScalarMatrix, ScalarMultiply, ScalarVector, SchurForm, SingularValues, SmithForm, StronglyConnectedBlocks, SubMatrix, SubVector, SumBasis, SylvesterMatrix, SylvesterSolve, ToeplitzMatrix, Trace, Transpose, TridiagonalForm, UnitVector, VandermondeMatrix, VectorAdd, VectorAngle, VectorMatrixMultiply, VectorNorm, VectorScalarMultiply, ZeroMatrix, ZeroVector, Zip]
```

Spaltenvektoren werden als <a,b,c,...> eingegeben und Zeilenvektoren als <a|b|c|...>. Matrizen sind dann Spaltenvektoren deren Einträge Zeilenvektoren sind oder Zeilenvektoren deren Einträge Spaltenvektoren sind, also zum Beispiel

```
> A:=<<-9, -11, 8> | <4, 6, -4> | <-8, -8, 7>>;
```

$$A := \begin{bmatrix} -9 & 4 & -8 \\ -11 & 6 & -8 \\ 8 & -4 & 7 \end{bmatrix}$$

Zum Berechnen des Minimalpolynoms von A als Polynom in x gibt es in LinearAlgebra den Befehl

```
> mu:=MinimalPolynomial(A, x);
```

$$\mu := 6 + x - 4x^2 + x^3$$

Mit dem Befehl factor() können wir Polynome faktorisieren

```
> factor(mu);
```

$$(x - 2)(x - 3)(x + 1)$$

und wir betrachten die Zerlegung $\mu=p*q$ mit

```
> p:=collect(op(%)[1]*op(%)[2], x); q:=op(%%)[3];
```

$$p := 6 + x^2 - 5x$$

$$q := x + 1$$

Nun berechnen wir die Projektionen auf die zugehörigen Primärkomponenten und hierzu brauchen wir die Polynome f,g mit $f \cdot p + g \cdot q = 1$

```
> gcdex(p,q,x,'f','g'); 'f'=f; 'g'=g;
```

$$1$$

$$f = \frac{1}{12}$$

$$g = \frac{1}{2} - \frac{x}{12}$$

```
> collect(g*q,x);
```

$$\frac{1}{2} - \frac{1}{12}x^2 + \frac{5}{12}x$$

Wir können mit eval() jetzt direkt $x=A$ einsetzen, die Formel in die eingesetzt wird muss dazu aber in einer geeigneten Form sein, nicht

```
> eval(g*q,x=A);
```

Error, (in rtable/Product) invalid arguments

sondern

```
> eval(%,x=A);
```

$$\begin{bmatrix} -1 & 0 & -2 \\ -2 & 1 & -2 \\ 1 & 0 & 2 \end{bmatrix}$$

Alternativ kann man solche Matrixrechnungen auch ohne Einsetzen direkt durchführen

```
> P:=(ScalarMatrix(1/2,3,3)-(1/12)*A).(A+IdentityMatrix(3));
```

$$P := \begin{bmatrix} -1 & 0 & -2 \\ -2 & 1 & -2 \\ 1 & 0 & 2 \end{bmatrix}$$

In dieser Rechnung sind

ScalarMatrix(v,n,m) - nxm Matrix mit v auf der Diagonalen

IdentityMatrix(n) - nxn Einheitsmatrix

. - Matrixmultiplikation, * wird nur für kommutative Verknüpfungen verwendet

Kommen wir zur Projektion auf den zweiten Faktor

```
> f;p;
```

$$\frac{1}{12}$$

$$6 + x^2 - 5x$$

```
> Q:=(A^2-5*A+6*IdentityMatrix(3))/12;
```

$$Q := \begin{bmatrix} 2 & 0 & 2 \\ 2 & 0 & 2 \\ -1 & 0 & -1 \end{bmatrix}$$

Testen wir das alles geklappt hat, da P,Q die Projektionen einer direkten Zerlegung des K^n sein sollen muss $P+Q=1$ und $PQ=0$ gelten, und tatsächlich sind

> $P+Q;$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

> $P.Q;$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Schauen wir uns auch noch einen konkreten Vektor u an den wir als $u = v + w$ mit v im Kern von p(A) und w im Kern von q(A) schreiben wollen

> $u := \langle 1, 2, 3 \rangle;$

$$u := \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

> $v := P.u; w := Q.u;$

$$v := \begin{bmatrix} -7 \\ -6 \\ 7 \end{bmatrix}$$

$$w := \begin{bmatrix} 8 \\ 8 \\ -4 \end{bmatrix}$$

Und wieder das Ergebnis überprüfen

> $v+w;$

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

> $p;q;$

$$\begin{array}{l} 6 + x^2 - 5x \\ x + 1 \end{array}$$

> $eval(p, x=A).v;$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

```
> (A+IdentityMatrix(3)).w;
```

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

9. Rationale Funktionen

Rationale Funktionen sind Brüche von Polynomen, diese können mit dem schon in der letzten Sitzung behandelten Befehl `normal()` erzeugt werden. Auf Zähler und Nenner solch eines Bruchs kann mit

`numer(B)` - Zähler von B
`denom(B)` - Nenner von B

zugegriffen werden. Ein Beispiel einer solchen Funktion ist

```
> f := (x^5 - 2*x^4 - 5*x^3 + 2*x^2 - 16*x - 10) / (x^3 - x^2 - 10*x - 8);
```

$$f := \frac{x^5 - 2x^4 - 5x^3 + 2x^2 - 16x - 10}{x^3 - x^2 - 10x - 8}$$

Eine oftmals auftauchende Problemstellung ist die Berechnung der sogenannten Partialbruchzerlegung einer solchen rationalen Funktion. Wir wollen uns erst einmal daran erinnern wie diese per Hand durchgeführt wird. Im ersten Schritt wird eine Polynomdivision durchgeführt um den polynomialen Anteil von f abzuspalten (dies ist nur nötig wenn der Zählergrad nicht schon echt kleiner als der Nennergrad ist). In unserem Beispiel wird $f = Q + r(x)/(x^3 - x^2 - 10x - 8)$ mit

```
> Q := quo(numer(f), denom(f), x);
```

$$Q := x^2 - x + 4$$

```
> r := rem(numer(f), denom(f), x);
```

$$r := 22 + 16x + 4x^2$$

Damit ist schon einmal

```
> 'f(x)' = Q + r/denom(f);
```

$$f(x) = x^2 - x + 4 + \frac{22 + 16x + 4x^2}{x^3 - x^2 - 10x - 8}$$

Im nächsten Schritt wird der Nenner des rationalen Anteils in seine (hier) Linearfaktoren zerlegt. Den Befehl hierfür haben wir schon im vorigen Abschnitt gesehen

```
> factor(denom(f));
```

$$(x - 4)(x + 2)(x + 1)$$

Nun wollen wir A,B,C bestimmen die die Bedingung

```
> r/denom(f) = A/(x+2) + B/(x+1) + C/(x-4);
```

$$\frac{22 + 16x + 4x^2}{x^3 - x^2 - 10x - 8} = \frac{A}{x+2} + \frac{B}{x+1} + \frac{C}{x-4}$$

erfüllen. Dies kann man auf verschiedene Weisen tun, eine (allerdings nicht die beste) Methode ist es die rechte Seite der obigen Gleichung auf den Hauptnenner zu bringen

> `normal(A/(x+2)+B/(x+1)+C/(x-4));`

$$\frac{Ax^2 - 3Ax - 4A + Bx^2 - 2Bx - 8B + Cx^2 + 3Cx + 2C}{(x-4)(x+2)(x+1)}$$

> `collect(numer(%),x);`

$$(C+B+A)x^2 + (3C-2B-3A)x + 2C-8B-4A$$

und dann einen Koeffizientenvergleich durchzuführen

> `Vergleich:=coeff(%,x,2)=4,coeff(%,x,1)=16,coeff(%,x,0)=22;`

$$\text{Vergleich} := C + B + A = 4, 3C - 2B - 3A = 16, 2C - 8B - 4A = 22$$

Das Lösen von Gleichungen mit Maple werden wir erst später genauer behandeln, hier verwenden wir den Befehl

`solve({Gleichungen}, {Unbekannte})`

zum Lösen unseres Gleichungssystems

> `solve({Vergleich},{A,B,C});`

$$\{A = 1, B = -2, C = 5\}$$

Damit haben wir die gesuchte Partialbruchzerlegung gefunden

> `'f(x)'=Q+1/(x+2)-2/(x+1)+5/(x-4);`

$$f(x) = x^2 - x + 4 + \frac{1}{x+2} - \frac{2}{x+1} + \frac{5}{x-4}$$

Maple hat natürlich einen "fertigen" Befehl dies alles zu tun, dazu kommen wir aber erst am Ende dieses Abschnitts.

Zuvor wollen wir uns als ein Beispiel die Verwendung von Maple zur Produktion solcher Aufgaben anschauen.

Nehme an wir wollen eine Aufgabe zur Partialbruchzerlegung herstellen und darauf achten das bei der Aufgabe alles ganzzahlig bleibt. Hierzu orientiert man sich am Ergebnis und rechnet rückwärts. Im Nenner hätten wir gerne ein Polynom dritten Grades mit drei ganzzahligen Nullstellen von denen eine leicht zu sehen ist, etwa bei 1. Sagen wir die Nullstellen sind 1, -6 und 7. Die Koeffizienten in der Partialbruchzerlegung sollen A=1, B=-2 und C=1 sein. Außerdem hätten wir gerne das zu Beginn eine Polynomdivision nötig ist, sagen wir der Zähler sollte Grad 5 haben und als Quotient soll sich x^2+x+1 abspalten. Der Rest soll also

> `1/(x-1)-2/(x+6)+1/(x-7);`

$$\frac{1}{x-1} - \frac{2}{x+6} + \frac{1}{x-7}$$

> `normal(%,expanded);`

$$\frac{20x - 62}{x^3 - 2x^2 - 41x + 42}$$

sein. Um den Zähler von Grad 5 zu kriegen, rechne

```
> sort(collect((x^2+x+1)*denom(%) + numer(%), x));
```

$$x^5 - x^4 - 42x^3 - x^2 + 21x - 20$$

und die gesuchte rationale Funktion ist

```
> f := % / denom(%%);
```

$$f := \frac{x^5 - x^4 - 42x^3 - x^2 + 21x - 20}{x^3 - 2x^2 - 41x + 42}$$

Dies ist ein bequemer Weg zur Herstellung solcher Aufgaben.
Zur umgekehrten Berechnung der Partialbruchzerlegung kennt Maple den `convert()` Befehl. Dies ist ein allgemeiner Befehl zur Umwandlung von Formeln zwischen verschiedenen Formen. Die Syntax ist

```
convert(Formel, Zielformat, ...)
```

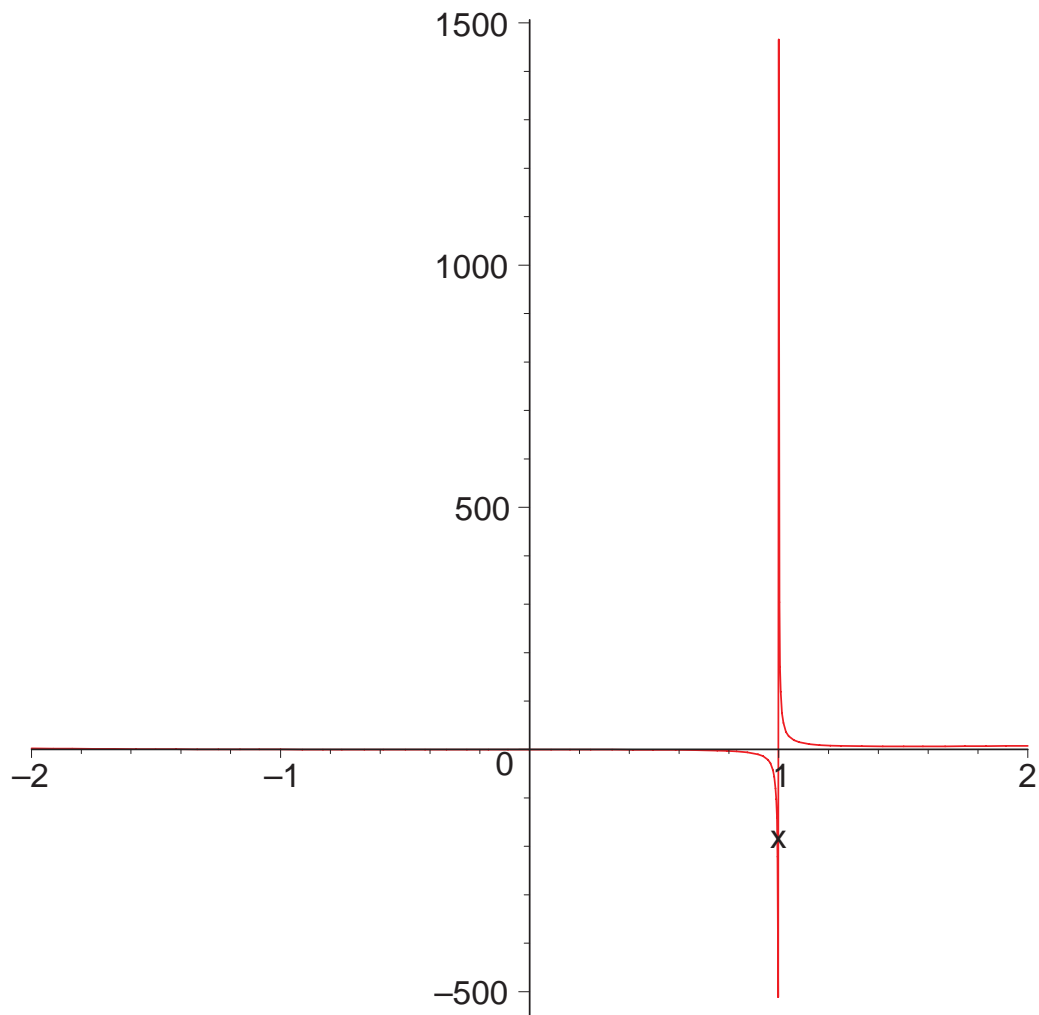
wobei für zusätzliche nach Zielformat unterschiedliche Argumente steht. Für die Partialbruchzerlegung ist das Zielformat 'parfrac', für partial fractions, und als zusätzliches Argument wird die Variable nach der Polynome gebildet werden angegeben

```
> convert(f, parfrac, x);
```

$$x^2 + x + 1 + \frac{1}{x-1} - \frac{2}{x+6} + \frac{1}{x-7}$$

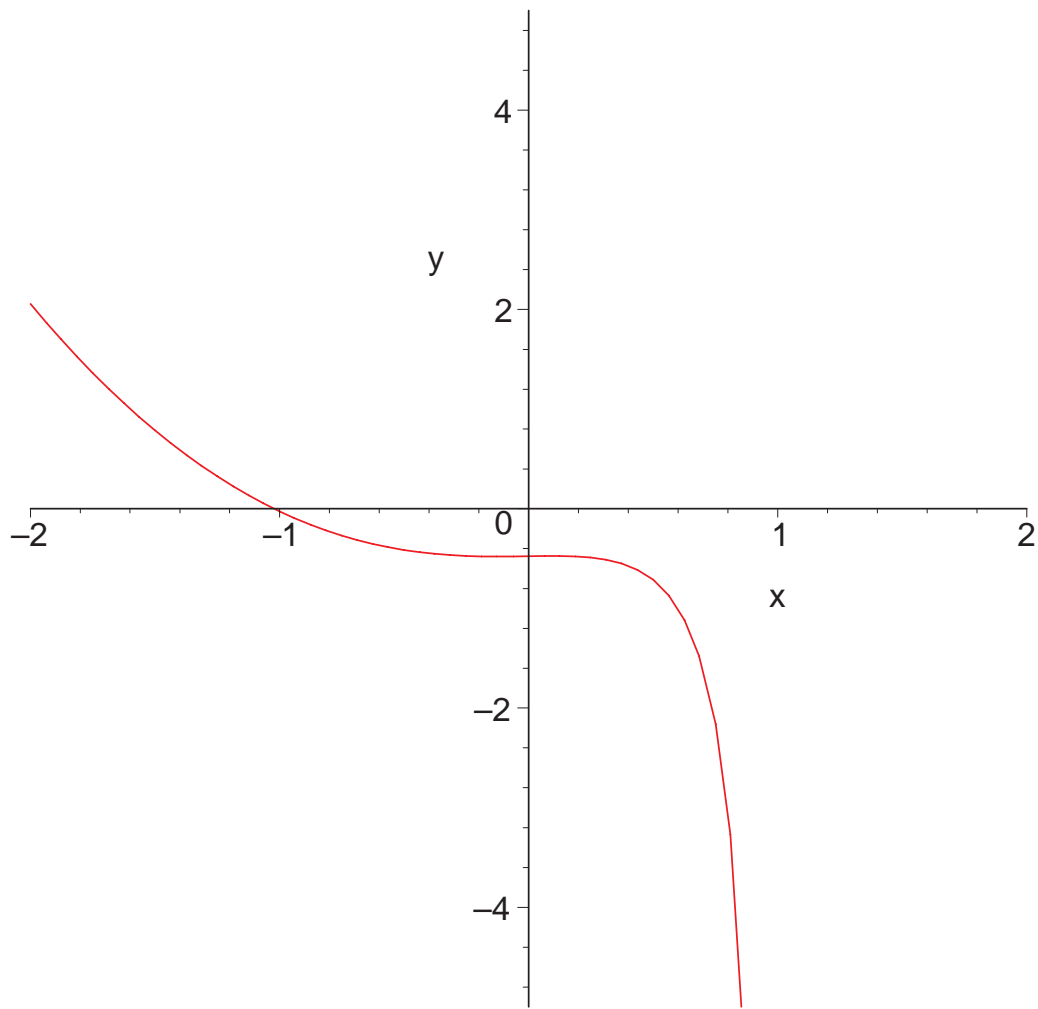
Zum Abschluß des Beispiels wollen wir die Funktion f noch plotten, sagen wir zwischen x=-2 und x=2

```
> plot(f, x=-2..2);
```



Das ist nicht besonders zufriedenstellend. Das Problem ist das wir über eine Polstelle hinweg plotten. Dies kann man durch Einschränkung des anzuzeigenden y-Bereichs korrigieren. Die Option `discont=true` sagt Maple außerdem das es Unstetigkeiten gibt.

```
> plot(f,x=-2..2,y=-5..5,discont=true);
```



[Was ist mit dem Graph für $x > 1$ los?

[`> eval(f, x=1.1);`

12.85881833

[`> eval(f, x=2.0);`

7.550000000

[Die Funktionswerte rechts bleiben also bis $x=2$ zu gross und sind daher nicht im y -Bereich unseres Plots zu sehen.

[10. Faktorisierung von Polynomen

[Wir haben bereits den Befehl `factor()` zur multiplikativen Zerlegung eines Polynoms gesehen. Dieser faktorisiert über den rationalen Zahlen, beispielsweise

[`> p:=x^4-6*x^3+9*x^2+4*x-12;`

$p := x^4 - 6x^3 + 9x^2 + 4x - 12$

[`> factor(p);`

$$(x+1)(x-3)(x-2)^2$$

```
> p:=x^3-11*x^2+23*x-13;
```

$$p := x^3 - 11x^2 + 23x - 13$$

```
> factor(p);
```

$$(x-1)(x^2-10x+13)$$

Hier wird p nicht ganz in Linearfaktoren zerlegt, da der zweite Faktor keine rationalen Nullstellen mehr hat. Schauen wir einmal nach wo die Nullstellen liegen. Zum Lösen von Gleichungen hatten wir schon im vorigen Abschnitt den Befehl

solve(Gleichung, Variable)

kennen gelernt, der die gegebene Gleichung nach der gegebenen Variable löst. Da es sich nur um eine Gleichung und eine Variable handelt brauchen wir diesmal keine geschweiften Klammern. Versuchen wir unser Glück:

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

$$1, 5 + 2\sqrt{3}, 5 - 2\sqrt{3}$$

Das Polynom p sollte also über dem Körper $Q(\sqrt{3})$ in Linearfaktoren zerfallen. Dies kann man Maple durch ein zusätzliches Argument in factor() mitteilen

```
> factor(p,sqrt(3));
```

$$(-x+5+2\sqrt{3})(x-5+2\sqrt{3})(x-1)$$

Schauen wir uns ein komplizierteres Beispiel an

```
> p:=x^5-7*x^4-22*x^3+90*x^2-75*x+13;
```

$$p := x^5 - 7x^4 - 22x^3 + 90x^2 - 75x + 13$$

```
> factor(p);
```

$$(x-1)(x^2-10x+13)(x^2+4x-1)$$

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

$$1, 5 + 2\sqrt{3}, 5 - 2\sqrt{3}, -2 + \sqrt{5}, -2 - \sqrt{5}$$

Diesmal brauchen wir den Körper $Q(\sqrt{3},\sqrt{5})$ zum Zerlegen. Auch hiermit kommt Maple zurecht

```
> factor(p,{sqrt(3),sqrt(5)});
```

$$(-x-2+\sqrt{5})(x+2+\sqrt{5})(x-5+2\sqrt{3})(-x+5+2\sqrt{3})(x-1)$$

Maple kann auch Polynome über endlichen Körpern faktorisieren. Nehmen wir als ein Beispiel den Primkörper $K = GF(11)$ mit 11 Elementen. Um modulo 11 zu rechnen kann man einfach "mod 11" anfügen, allerdings führt dies noch nicht zum gewünschten Ergebnis

```
> factor(p) mod 11;
```

$$(x+10)(x^2+x+2)(x^2+4x+10)$$

Dies ist zwar eine Faktorisierung von p über GF(11) aber nicht diejenige in irreduzible Faktoren. Was hier passiert ist das zuerst factor(p) ausgewertet wird, was p wie oben über Q zerlegt, und anschließend die Faktoren modulo 11 reduziert werden.

Um die Faktorisierung über GF(11) zu erreichen müssen

wir `Factor(p)` statt `factor(p)` verwenden. Diese groß geschriebene Form ist die "nicht ausgewertete" Variante von `factor`, es wird erst einmal nur vermerkt das `p` zu faktorisieren ist, die Rechnung wird aber nicht durchgeführt, ein Verhalten das manchmal als "lazy evaluation" bezeichnet wird.

Wenn wir nun

```
> Factor(p) mod 11;
```

```
(x + 6)(x + 7)(x + 9)(x + 5)(x + 10)
```

rechnen so wird die Berechnung vom "mod" angestossen und es wird von vornherein in $GF(11)$ gerechnet. Über $GF(11)$ zerfällt `p` dann tatsächlich in Linearfaktoren.