

Maple-Befehle

Algebra

<code>evalf</code>	Liefert einen dezimalen Näherungswert des Ausdruckes <code>evalf((4/5)^8);</code>
<code>expand</code>	Multipliziert einen Ausdruck aus oder entwickelt ihn <code>expand((x - 3)*(2*x + 5));</code>
<code>factor</code>	Faktorisiert den Ausdruck, falls möglich <code>factor(x^2 - 3*x - 4);</code>
<code>simplify</code>	Vereinfacht den Ausdruck <code>simplify(1/x - 3/(x + 2));</code>
<code>numer</code>	Liefert den Zähler eines Bruches <code>numer(5*x/(2*x^2 - 3*x + 5));</code>
<code>denom</code>	Liefert den Nenner des Bruches <code>denom(5*x/(2*x^2 - 3*x + 5));</code>
<code>solve</code>	Löst eine Gleichung exakt <code>solve(x^2 - 3*x - 4 = 0);</code>
<code>fsolve</code>	Bestimmt eine Näherungslösung für Gleichungen <code>fsolve(x^2 - 3*x - 5 = 0);</code>
<code>subs</code>	Substituiert die ersten Argumente im letzten Argument <code>subs(x=2, y =3, 2*x + 5*y);</code>

Funktion. Graphische Darstellung

<code>-></code>	Definiert eine Funktion <code>f:=x -> 3*x^2 - 5;</code>
<code>plot</code>	Zeichnet zweidimensionale Diagramme <code>plot(x^2 - 2);</code> oder <code>plot(x^2 - 2, x=-3..3);</code> oder

oder

```
plot(x^2 - 2, x=-3..3, y=0..5);
```

oder

```
f:=x -> x^2 - 2; plot(f(x), x=-3..3, y=0..5);
```

oder

```
plot({x^2-2, x+4}, x=-3..3); (mehrere Kurven)
```

oder

```
plot([sin(t),cos(2*t)], t=0..2*Pi); (Parameterdarstellung)
```

plot3d Zeichnet dreidimensionale Diagramme

```
plot3d(x^2+y^2, x=-2..2, y=-2..2);
```

POINT Option für die Anweisung plot. Zeichnet den Graphen nur *punktweise*

```
plot(x^2 - 2, style=POINT);
```

Infinitesimalrechnung

limit Berechnet den Grenzwert einer Funktion

```
limit(sin(x)/x, x=0);
```

```
limit(sqrt(x^2-4*x)-x, x=infinity);
```

diff Differenziert einen Ausdruck nach den angegebenen Variablen

```
diff(cos(2*x), x);
```

int Berechnet bestimmte und unbestimmte Integrale

```
int(tan(5*x), x);
```

sum Berechnet endliche und unendliche Summen

```
sum(k^2, k=1..50);
```

taylor Liefert die Taylor-Reihenentwicklung um den angegebenen Punkt

```
taylor(exp(x), x=0.4);
```

Differentialgleichungen

dsolve Löst Differentialgleichungen (algebraische Lösung)

```
dsolve(diff(y(x), x)+2*x*y(x)=0); (-> allg. Lös.)
```

```
dsolve({diff(y(x), x)+2*x*y(x)=0, y(0)=1}); (-> part. Lös.)
```

Lineare Algebra

- vector** Erzeugt einen Vektor
`v:=vector([1, 2, 3]);`
- matrix** Erzeugt eine Matrix
`A:=matrix([[5, 1],[-3, 6]]);` oder
`A:=matrix(2, 2, [5, 1, -3, 6]);`
- diag** Erzeugt eine Diagonalmatrix
`diag(1, 5, 2, 4, 3);`
- transpose** Berechnet die Transponierte einer Matrix
`A:=matrix([[1, -1, 4],[3, 2, -1],[2, 1, -1]]);`
`transpose(A);`
- det** Berechnet die Determinante einer Matrix
`A:=matrix([[-3, 5],[8, -1]]);`
`det(A);`
- dotprod** Berechnet das Skalarprodukt zweier Vektoren
`dotprod([1, a, 5], [b, 2, c]);`
- charpoly** Berechnet das charakteristische Polynom einer quadratischen Matrix
`A:=matrix([[3, 7],[8, 1]]);`
`charpoly(A, x);`
- eigenvals** Berechnet die Eigenwerte einer Matrix
`A:=matrix([[2, 2],[3, 1]]);`
`eigenvals(A);`
- genmatrix** Erzeugt die Koeffizientenmatrix für ein lin. Gleichungssystem
`genmatrix([x+2*y=3, 3*x-5*y=0],[x,y]);`
- linsolve** Liefert die Lösung eines lin. Gleichungssystems
`A:=matrix([[1, -1, 4],[3, 2, -1],[2, 1, -1]]);`
`v:=vector([-2, 3, 1]);`
`linsolve(A,v);`