

Mathematik mit Maple

- ```
restart;
with(plots):
```

löscht alle Variablen,  
ermöglicht eine Grafikausgabe,
- ```
solve(x^2+4*x = 0, x);  
g:= solve(2^x = 5, x);  
evalf(g, 4);
```

löst die Gleichung,
ein Doppelpunkt unterdrückt die Ausgabe,
berechnet (evaluiert) g auf 4 Stellen genau,
- ```
f:= x->x^2+3;
f(3);
```

eine Funktion wird eingegeben,  
berechnet einen Funktionswert,
- ```
plot(f(x), x = 0 .. 5);  
plot(f(x), x = 0 .. 5, y = 0 .. 10);  
plot(x^3-x^2, x = -2 .. 2);
```

zeichnet den Graphen einer Funktion,
- ```
f:= x->2*x;
g:= x->1/x;
p1:= plot(f(x), x = -5 .. 5, y = -2 .. 2);
p2:= plot(g(x), x = -5 .. 5);
display([p1, p2]);
```

zeichnet die Graphen zweier Funktionen,  
die Grafiken von  $f$  und  $g$  werden zunächst in den  
Variablen  $p1$  und  $p2$  gespeichert,
- ```
solve(f(x) = 0, x);  
diff(f(x), x);  
int(f(x), x = 0 .. 2);
```

berechnet die Nullstellen,
leitet f ab,
integriert f in den Grenzen von 0 bis 2,
- ```
f:= x->x^3-3*x^2;
g:= diff(f(x), x);
solve(g = 0, x);
```

berechnet die Nullstellen der 1. Ableitung,
- ```
normal(g); normal(1/a+1/b);  
factor(x^3+x^2);  
expand((x-3)*(x+4));  
b:= (x^3+1)/x^2;  
convert(b, parfrac, x);
```

bringt den Bruchterm auf einen Bruchstrich,
zerlegt den Term in Faktoren,
löst Klammern auf,

zerlegt den Bruchterm (in Partialbrüche),
- ```
f:= x->x^4-x^2;
f1:= unapply(diff(f(x), x), x);
plot(f1(x), x = -5 .. 5, y = -4 .. 4,
 numpoints = 500);
```

zeichnet den Graphen der 1. Ableitung,  
berechnet 500 Werte für eine genaue Zeichnung,
- ```
with(plots):  
animate(x^2*sqrt(k-x^2),  
        x = -5 .. 5, k = 1 .. 10, frames = 40);
```

zeichnet nacheinander die Graphen der
Funktionenschar (mit dem Parameter k),
- ```
evalf(Pi, 50);
evalf(E, 50);
```

gibt  $\pi$  und  $e$  auf 50 Stellen genau aus,
- ```
gl1:= x+4*y = 5;  
gl2:= 5*x-y = 4;  
solve({gl1, gl2}, {x, y});
```

löst ein Gleichungssystem mit 2 Variablen,
entsprechend werden Gleichungssysteme
mit mehr als 2 Variablen gelöst,
- ```
simplify(...);
```

vereinfacht Terme,

## Mathematik mit Maple Fortsetzung

14. `with(plots);  
plot3d(x^2+y^2, x = -2 .. 2, y = -2 .. 2,  
axes = boxed, style = patchcontour);` zeichnet Funktionen mit 2 Variablen,  
Orientierung kann mit den Pfeiltasten  
geändert werden,  
weitere Optionen:  
`style = patchnograd, shading = zhue,  
light = [60, 45, 1, 1, 1], ambientlight = [0.3, 0.3, 0.3]`
15. `int(int(x^2+y^2, x = 0 .. 1), y = 1 .. 2);` berechnet ein Doppelintegral,
16. `dgl:=diff(y(x),x)=y(x)+1;  
f:=dsolve( {dgl, y(0)=2}, y(x) );  
g:=unapply(rhs(f), x);  
plot (rhs(f),x=-5..5, y=-2..10);  
oder  
plot (g(x),x=-5..5, y=-2..10);` löst Differentialgleichungen und gibt den Graph aus,
17. `f:=x->1/(2^n);  
sum( f(n), n = 0 .. infinity);` berechnet endliche und unendliche Summen,
18. `a := 3*[3,4,1] - [5,-2,7];  
evalm(a);` fasst Vektoren zusammen, evaluieren,  
m steht für Matrix (rechteckiges Zahlenschema),
19. `restart; with(plots):  
  
p1 := plot3d([r-s, r+2*s, 1+2*r+s], r=-20..20, s=-20..20, axes=framed, grid=[30, 30], color=blue):  
  
p2 := plot3d([10,5,2], r=0..1, s=0..1, axes=framed, style=point, grid=[30, 30], color=red):  
  
p3 := spacecurve([3+s*3, 1+5*s, 1-s], s=-10..10, axes=boxed, grid=[50, 50], color=black):  
  
display([p1,p2,p3], style=hidden, scaling=constrained, axes=boxed, view=-20 .. 20, labels=[x,y,z]);`
- p1 Die Grafik einer Ebene benötigt zwei Parameter, z.B. r und s.  
p2 Um einen Punkt zu plotten, ist auch ein zweidimensionaler Parameterbereich anzugeben.  
p3 Für Geraden mit einem eindimensionalen Parameterbereich ist spacecurve zu verwenden.
20. `plot3d(x/2*sqrt(8-x), alpha=0..2*Pi, x=0..8, grid=[50,50], coords=cylindrical, axes=boxed);`  
Rotationskörper