```
>
```

```
####################################
# Zeichnen von Funktionen mit Maple #
####################################
```

```
>
```

```
# Marc Wagner
# Zimmer-Nr.: 02.734 (bei Fragen oder Problemen koennt Ihr
#             jederzeit vorbeikommen)
# Datum: 21. Mai 2003
# Version: Maple 7
```

```
>
```

```
#######################
# Allgemeine Hinweise #
#######################

# Kommentare beginnen mit   #   (diese Zeilen werden von Maple
# ignoriert).

# Gross- und Kleinschreibung muss beachtet werden, d. h. X ist
# nicht das gleiche wie x.

# Befehle die mit   ;   enden   -->   Bildschirmausgabe.
# Befehle die mit   :   enden   -->   Keine Bildschirmausgabe.
```

```
>
```

```
############################
# Zuweisen von Ausdruecken #
############################

# A := B   -->   A wird der Ausdruck B zugewiesen.

x := 2+3;  # x wird der Wert 2+3=5 zugewiesen; das Ergebnis der
           # Zuweisung wird ausgegeben.

y := 7:  # y wird der Wert 7 zugewiesen; das Ergebnis der
         # Zuweisung wird nicht ausgegeben.

x+y;  # x+y wird berechnet und ausgegeben.

z := x+y:  # z wird der x+y, d. h. 12, zugewiesen; das Ergebnis
           # der Zuweisung wird nicht ausgegeben.

z;  # Der Wert von z wird ausgegeben.
```

$$x := 5$$

$$12$$

$$12$$

```
>
```

```
#############################################
# Loeschen aller bisher berechneten Ergebnisse #
#############################################

# restart   -->   Alle gespeicherten Ergebnisse werden
#                 geloescht. Tip: restart zu Beginn jedes
#                 Maple-Programms ausfuehren.

x;  # Der Wert von x, d. h. 12, wird ausgegeben.

restart:  # x und alle anderen Variablen werden geloescht.

x;  # x besitzt momentan keinen Wert; es wird daher 'x'
    # ausgegeben.
```

$$5$$

$$x$$

```
>
```

```
#############################################
# Definition von Funktionen in einer Variablen #
#############################################

f := x -> x^2;  # f wird als Funktion f(x)=x*x definiert.

f(1);  # f wird an der Stelle 1 ausgewertet; das Ergebnis wird
       # ausgegeben.
f(2);
f(3);
```

$$f := x \rightarrow x^2$$

$$1$$

$$4$$

$$9$$

```
>
```

```
g := x -> sin(x);

g(0);
g(Pi/2);
```

```
# Tip: Maple besitzt sehr viele vordefinierte Funktionen (z. B.
# cos, tan, cot, sinh, cosh, exp, ln, sqrt, ...). In der
# umfangreichen Hilfe nachlesen (Button in der rechten oberen
# Ecke)!
```

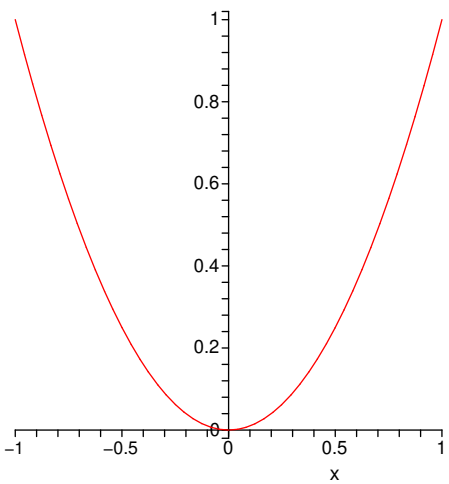
$$g := x \rightarrow \sin(x)$$

$$0$$

$$1$$

```
>
```

```
#############################################
# Zeichnen einer Funktion in einer Variablen #
#############################################

plot(f(x), x=-1..1);  # f(x) wird von x=-1 bis x=1 gezeichnet.
```




```
>
```

```
#####################################################
# Zeichnen mehrerer Funktionen in einer Variablen. #
#####################################################

# Es werden die Funktionen f(x), g(x) und f(g(x)) von x=-Pi bis
# x=Pi gezeichnet. Eine derartige Liste von Funktionen kann
# durch   [f_1(x),f_2(x), ... ,f_n(x)]   angegeben werden.

plot([f(x), g(x), f(g(x))], x=-Pi..Pi);
```

Error, invalid plot structure

```
>
```

```
#####################################################
# Einschraenken des darzustellenden Definitions- und #
# Wertebereiches beim Zeichnen von Funktionen        #
#####################################################

# Die Option   view=[x_min..x_max,f_min..f_max]   erlaubt die
# Angabe der minimalen und maximalen Ausdehnungen der
# Koordinatenachsen.

plot([f(x), g(x), f(g(x))], x=-Pi..Pi, view=[-Pi..Pi,-1.5..2]);
```

Error, invalid plot structure

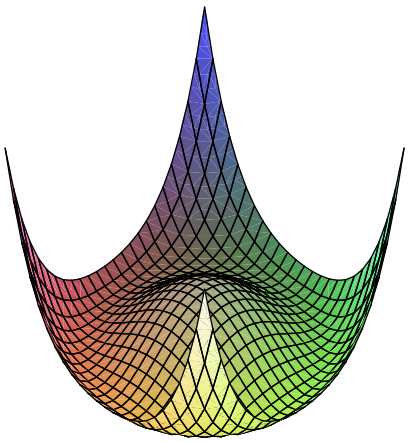
```
>
```

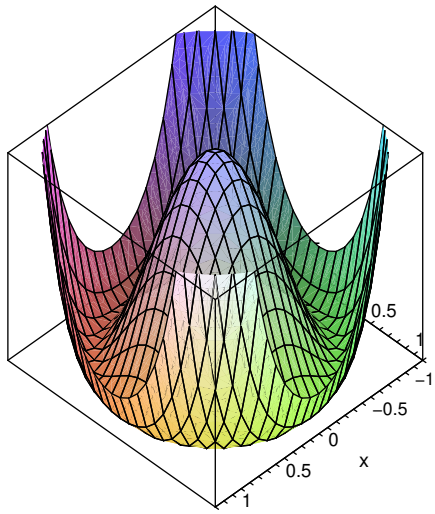
```
#####################################################
# Definition und Zeichnen einer Funktion in mehreren #
# Variablen                                          #
#####################################################

# Definition einer bivariaten Funktion h.
h := (x,y) -> (x^2 + y^2 - 1)^2;

# Zeichnen der bivariaten Funktion h.
plot3d(h(x,y), x=-1.2..1.2, y=-1.2..1.2);

# Zeichnen der bivariaten Funktion h, wobei der darzustellende
# Bereich durch   view[...]   eingeschraenkt wurde.
plot3d(h(x,y), x=-1.2..1.2, y=-1.2..1.2,
  view=[-1.2..1.2,-1.2..1.2,0..1], axes=boxed);

# Tip:   plot   und   plot3d   besitzen viele nuetzliche
# Optionen. In der Hilfe nachlesen!
```

```
>
```

```
#####################################################
```

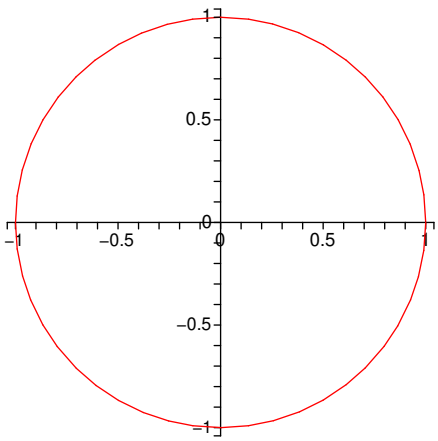
$$h := (x, y) \to (x^2 + y^2 - 1)^2$$



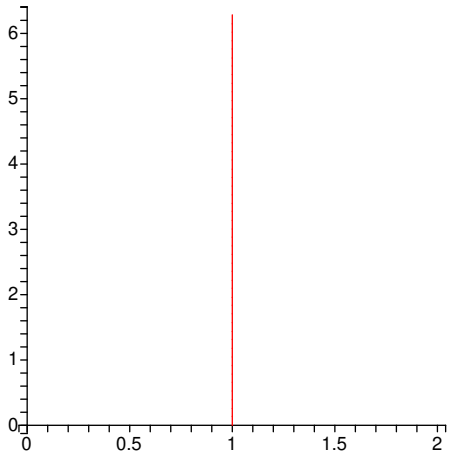



```
>
```

```
#######################################################
# Definition und Zeichnen einer parametrischen Kurve #
#######################################################

# Zeichnen der parametrischen Kurve
#   x = sin(t)
#   y = cos(t)
# von t=0 bis t=2*Pi.
plot([sin(t), cos(t), t=0..2*Pi]);
```



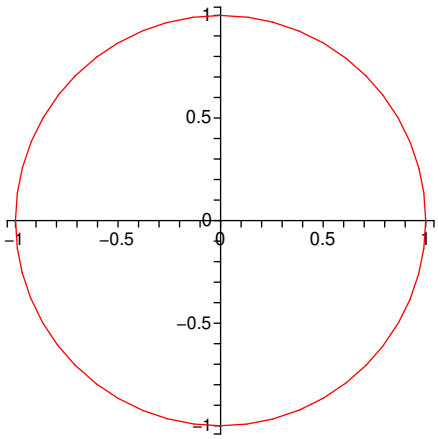
$$\phi := t \to t$$




```
>
```

```
#########################################################
# Definition und Zeichnen einer parametrischen Kurve in #
# Polarkordinaten                                       #
#########################################################

r := t -> 1;  # Radialkoordinate.
phi := t-> t;  # Winkelkoordinate.

# Hier wird die Radialkoordinate nach rechts und die
# Winkelkoordinate nach oben aufgetragen.
plot([r(t), phi(t), t=0..2*Pi]);

# Bei Angabe der Option   coords=polar   werden die angegebenen
# Koordinaten als Polarkoordinaten interpretiert und die
# Funktion entsprechend gezeichnet.
plot([r(t), phi(t), t=0..2*Pi], coords=polar);
```

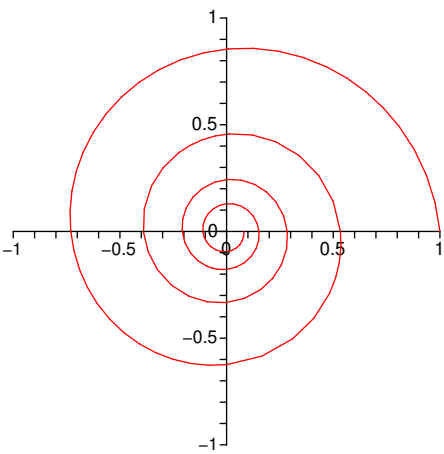
$$r := t \to 1$$

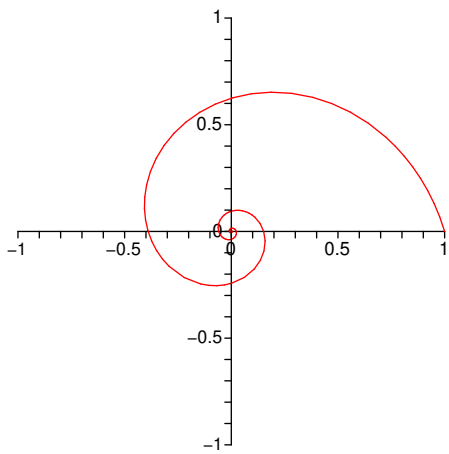



```
plot(r(phi), phi=0..4*Pi, coords=polar, view=[-1..1,-1..1]);

# --> Fehler weil Konstanten noch unbestimmt sind. Vor dem
#     Zeichnen muessen alle Konstanten mit Zahlenwerten belegt
#     werden.
```

$$r := \phi \to r0\, e^{(-\kappa \phi)}$$

```
Warning, unable to evaluate the function to numeric values in the region; see the
plotting command's help page to ensure the calling sequence is correct

Error, empty plot

>
# Konstanten koennen mit   subs   fuer den aktuellen Befehl
# Zahlen zugewiesen werden.

# Zeichnen der Loesung mit der speziellen Wahl der Konstanten
# r0=1 und kappa=0.1.
plot(subs(r0=1, kappa=0.1, r(phi)), phi=0..8*Pi, coords=polar,
  view=[-1..1,-1..1]);

# Zeichnen der Loesung mit der speziellen Wahl der Konstanten
# r0=1 und kappa=0.3.
plot(subs(r0=1, kappa=0.3, r(phi)), phi=0..8*Pi, coords=polar,
  view=[-1..1,-1..1]);

>
##########################################################
# So haette man Maple bei vergangenen Theoretikas und #
# Hausaufgaben einsetzen koennen                       #
##########################################################

>
#############################
# 3. Hausaufgabe, Aufgabe 4 #
#############################

# Bewegung eines Teilchens im Potential U=-alpha/R^2 mit den
# Anfangsbedingungen l<sqrt(2*mu*alpha) und E=0. Die Loesung
# ist eine "logarithmische Spirale".

# Das ist die berechnete Loesung, die zwei unbestimmte
# Konstanten enthaelt.
r := phi -> r0 * exp(-kappa*phi);
```






```
>
# Eine andere Moeglichkeit der Zuweisung besteht durch   :=   .
# Nachteil: Die Konstanten sind im Weiteren keine Konstanten
# mehr sondern Zahlen.

subs(r0=7, r(phi));  # Der Konstanten r0 wird nur in dieser
                     # Zeile der Wert 7 zugewiesen.

r(phi);  # r0 ist nach wie vor unbestimmt.

r0 := 5:  # r0 wird fuer immer der Wert 5 zugewiesen.

r(phi);  # Die Konstante r0 tritt in der Loesung nicht mehr
         # auf.
```

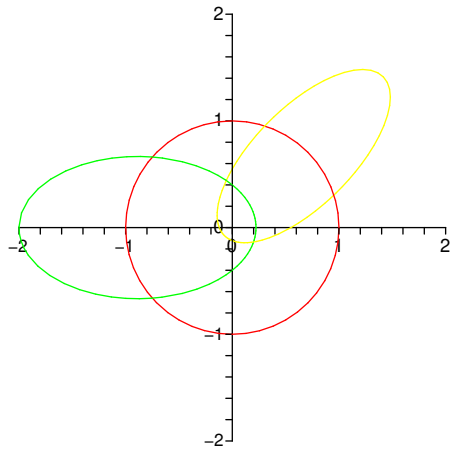
$$7\, e^{(-\kappa \phi)}$$

$$r0\, e^{(-\kappa \phi)}$$

$$5\,e^{(-\kappa\,\phi)}$$

```
>
  ###################
  # Kepler-Ellipsen #
  ###################

  # Die allgemeine Loesung des Kepler-Problems.
  r := phi -> p / (1 + epsilon * cos(phi+phi0));

  # Zeichnen von drei verschiedenen Kepler-Ellipsen in ein
  # Diagramm.
  plot([
    subs(p=1.0, epsilon=0, phi0=0, r(phi)),
    subs(p=0.4, epsilon=0.8, phi0=0, r(phi)),
    subs(p=0.2, epsilon=0.9, phi0=3*Pi/4, r(phi))
  ], phi=0..2*Pi, coords=polar, view=[-2..2,-2..2]);
```

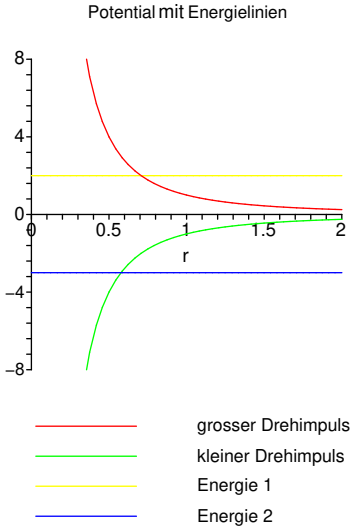
$$r := \phi \rightarrow \frac{p}{1 + \varepsilon \cos(\phi + \phi 0)}$$




```
>
  ##############################
  # 3. Hausaufgabe, Aufgabe 2 #
  ##############################

  # Das (1/r^2)-Potential mit Energielinien.

  U := r -> beta / r^2;

  plot([
      subs(beta=1, U(r)),
      subs(beta=-1, U(r)),
      2,
      -3
    ], r=0..2, view=[0..2,-8..8],
    title="Potential mit Energielinien",
    legend=["grosser Drehimpuls", "kleiner Drehimpuls", "Energie 1",
    "Energie 2"]
    );
```

```
  # Mit den Optionen   title    und   legend   koennen Diagramme
  # lesbarer gestaltet werden.
```

$$U := r \rightarrow \frac{\beta}{r^2}$$

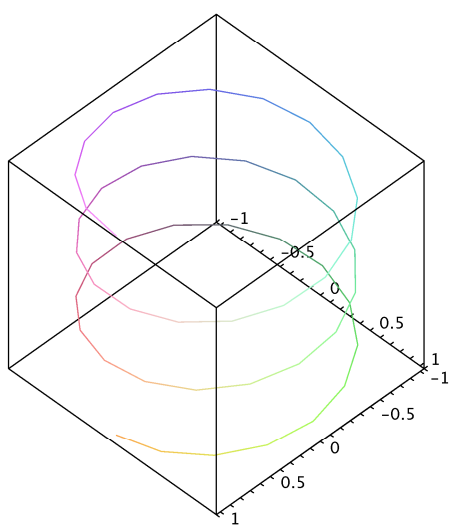



```
>
  ##############################################
  # Zeichnen einer dreidimensionalen Raumkurve. #
  ##############################################

  # Eine derartige Spiralbahn tritt z. B. auf, wenn sich ein
  # geladenes Teilchen in einem konstanten Magnetfeld bewegt
  # (Lorenz-Kraft).

  r_ := 1.0:
  omega := 1.0:
  vz := 1.0:
```

```
  x := t -> r_ * cos(omega*t);
  y := t -> r_ * sin(omega*t);
  z := t -> vz*t;


  # Mit   with(...)   werden spezielle Befehle verfuegbar
  # gemacht, z. B. das Zeichnen einer Raumkurve.

  with(plots):
  spacecurve([x(t), y(t), z(t), t=0...6*Pi], axes=boxed);
```

$$x := t \rightarrow r\_ \cos(\omega\, t)$$
$$y := t \rightarrow r\_ \sin(\omega\, t)$$
$$z := t \rightarrow vz\, t$$

Warning, the name changecoords has been redefined




```
>
```