Ostervortrag zum

Linux-Stammtisch

am 07.04.2017

Thema: Visualisierung mit MAPLE

Sybille Handrock

1 Computeralgebrasysteme

- Computeralgebra beschäftigt sich mit Methoden zum Lösen mathematischer Probleme durch Algorithmen zum symbolischen und algebraischen Rechnen.
- Computeralgebra ermöglicht eine formelmäßige Behandlung mathematischer Probleme auf dem Computer.
- Der numerischen Mathematik mit Gleitkomma-Arithmetik und Rundungsfehlerproblematik stehen algebraische Rechnungen sowie symbolische Manipulation von Formeln gegenüber.

Es existiert eine Fachgruppe Computeralgebra mit den Sprechern: Prof. Dr. Gregor Kemper (TU München), Prof. Dr. Florian Heß (Carl von Ossietzky Universität Oldenburg).

Homepage der Fachgruppe: www.fachgruppe-computeralgebra.de

Einige verbreitete Computeralgebrasysteme

- **Maple 18**
- **Mathematika 11**
- **Maxima**

Mit diesen Systemen sind sowohl Formelmanipulation und grafische Darstellungen als auch numerische Berechnungen möglich. Im Gegensatz dazu enthält **Matlab** keine Werkzeuge zur Formelmanipulation.

Das Computeralgebrasystem Maple wurde in Waterloo im Bundesstaat Ontario (Kanada) entwickelt, wo auch die Firma ihren Sitz hat.

- Ab Maple 11 ist ein persönlicher Aktivierungscode erforderlich.
- Sehr guter Service. Es gibt auch eine Anlaufstelle in Deutschland.
- Zur Weiterbildung werden monatlich Webinare angeboten.

Vorteile von Maple

- Einfach in der Bedienung.
- Übersichtliche elektronische Hilfe mit durchgerechneten Beispielen.
- Weitgehende analytische Vereinfachungen möglich.
- Numerische Rechnungen mit hoher Genauigkeit durchführbar.
- Zwei- und dreidimensionale Visualisierungen mithilfe des Grafikprogramms.

2 Lineare Gleichungssysteme

Wir betrachten das **inhomogene IGS**

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 &= y_1 \\ a_{21}x_1 + a_{22}x_2 &= y_2 \end{aligned} \tag{2.1}$$

zusammen mit dem **homogenen IGS**

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 &= 0 \\ a_{21}x_1 + a_{22}x_2 &= 0. \end{aligned} \tag{2.2}$$

Durch (2.1) und (2.2) sind jeweils zwei Geraden g_1 und g_2 in der Ebene festgelegt.

1. Sei (2.1) **unlösbar** (überbestimmtes System), dann sind g_1 und g_2 parallel zueinander, wobei $g_1 \neq g_2$ gilt.

2. Ist (2.1) **lösbar**, so gibt es zwei Fälle:

   (1) (2.1) besitzt **genau eine Lösung** (bestimmtes System), dann besitzen g_1 und g_2 genau einen Schnittpunkt,

   (2) (2.1) besitzt **unendlich viele Lösungen** (unterbestimmtes System), dann fallen g_1 und g_2 zusammen: $g_1 \equiv g_2$.

3. (2.2) ist **stets lösbar** und es gibt zwei Fälle:

   (1) (2.2) besitzt nur die **triviale Lösung** $(x_1, x_2) = (0, 0)$, dann schneiden sich g_1 und g_2 im Nullpunkt,

   (2) (2.2) besitzt **unendlich viele nichttriviale Lösungen, d.h.vom Nullpunkt verschiedene Lösungen**, dann fallen g_1 und g_2 zusammen und gehen durch den Nullpunkt.

¸Geometrische Interpretation siehe Maple-Datei zu linearen Gleichungssystemen

3 Funktionen

Skalare Funktionen

- **Skalare Funktion einer Variablen**

$$u = f(x) \quad x \in [a, b]$$

- **Skalare Funktion zweier Variablen (ebenes Skalarfeld)**

$$u = f(x, y) \qquad (x, y) \in [a, b] \times [c, d].$$

- **Skalare Funktion dreier Variablen (räumliches Skalarfeld)**

$$u = f(x, y, z) \qquad (x, y, z) \in [a, b] \times [c, d] \times [e, f].$$

Vektorfunktionen

- **Vektorfunktionen einer Variablen in der Ebene**

$$\mathbf{v}(t) = x(t)\mathbf{i} + y(t)\mathbf{j} \quad \text{bzw.} \quad \mathbf{v}(t) = \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} \quad t \in [t_1, t_2]$$

- **Vektorfunktionen einer Variablen im Raum**

$$\mathbf{v}(t) = x(t)\mathbf{i} + y(t)\mathbf{j} + z(t)\mathbf{k} \quad \text{bzw.} \quad \mathbf{v}(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix} \quad t \in [t_1, t_2]$$

- **Vektorfunktionen zweier Variablen (ebenes Vektorfeld)**

$$\mathbf{v}(t, r) = x(t, r)\mathbf{i} + y(t, r)\mathbf{j} \quad \text{bzw.} \quad \mathbf{v}(t, r) = \begin{pmatrix} x(t, r) \\ y(t, r) \end{pmatrix} \quad (t, r) \in [t_1, t_2] \times [r_1, r_2]$$

- **Vektorfunktionen dreier Variablen (räumliches Vektorfeld)**

$$\mathbf{v}(t, r, s) = x(t, r, s)\mathbf{i} + y(t, r, s)\mathbf{j} + z(t, r, s)\mathbf{k} \quad \text{bzw.} \quad \mathbf{v}(t, r, s) = \begin{pmatrix} x(t, r, s) \\ y(t, r, s) \\ z(t, r, s) \end{pmatrix}$$

$(t, r, s) \in [t_1, t_2] \times [r_1, r_2] \times [s_1, s_2]$

Eine Vektorfunktion mit drei Koordinatenfunktionen, die jeweils von zwei Variablen abhängen, beschreibt eine Fläche im Raum.

Beispiele siehe Maple-Dateien zu Funktionen

4 Fraktale

Der Begriff **Fraktal** wurde 1975 vom französisch-amerikanischen Mathematiker *Benoit Mandelbrot (1924-2010)* eingeführt.

Etwas vereinfacht sind **Fraktale** Mengen, die im Allg. keine ganzzahlige Dimension besitzen.

Einfachster Fall der Erzeugung von Fraktalen: Ein Iterationsprozess wird für ein Linienmuster (eindimensionales Gebilde) beliebig oft ausgeführt. Damit füllt sich mit der Zeit die gesamte Zeichenfläche mit Linien und das eindimensionale Gebilde nähert sich einem zweidimensionalen (keine ganzzahlige Dimension).

Fraktale in der Natur: Fraktale Struktur besitzen

- Blumenkohl
- grüner Blumenkohl (Romanesco)
- Farne

Figure 1: Romanesco

Anwendungen:

- Breitband Sende-und Empfangstechnik (Fraktalantennen)
- Kristallwachstum
- Belousov-Zhabotinsky-Reaktion: Veranschaulichung chaotischer Systeme
- Unmöglichkeit einer exakten Bestimmung der Küstenlänge

Einige wichtige Fraktale

- **Burning Ship-Fraktal** (*M. Michelitsch, O. E. Rössler* (1992))

  Es wird erzeugt durch Iteration der Folge

  $$z_{n+1} = (\mid \mathrm{Re}(z_n) \mid + \mathrm{i} \mid \mathrm{Im}(z_n) \mid)^2 + c, \quad z_0 = 0$$

  in der komplexen Ebene $\mathbb{C}$. Dabei ist c eine komplexe Zahl.

- **Mandelbrot-Fraktal**

  Es wird erzeugt durch Iteration der Folge

  $$z_{n+1} = z_n^2 + c, \text{ und dem Anfangsglied } z_o = 0 \tag{4.3}$$

  in der komplexen Ebene $\mathbb{C}$. Dabei ist c eine komplexe Zahl. Es spielt eine bedeutende Rolle in der Chaosforschung.

  Die **Mandelbrot Menge** ist die Menge aller komplexen Zahlen, bei welcher die rekursiv definierte Folge komplexer Zahlen $z_0, z_1, z_2, \cdots$ mit dem Bildungsgesetz (4.3) beschränkt bleibt, d.h. der Betrag der Folgenglieder ist eine reelle Zahl.

- **Julia-Fraktal** (*Gaston Maurice Julia (1893-1978)*)

  Die **Julia-Menge** J_c zu einer komplexen Zahl c ist definiert als der Rand der Menge aller Anfangswerte z_0, für die die Zahlenfolge (4.3) beschränkt bleibt.

- **Newton-Fraktal**

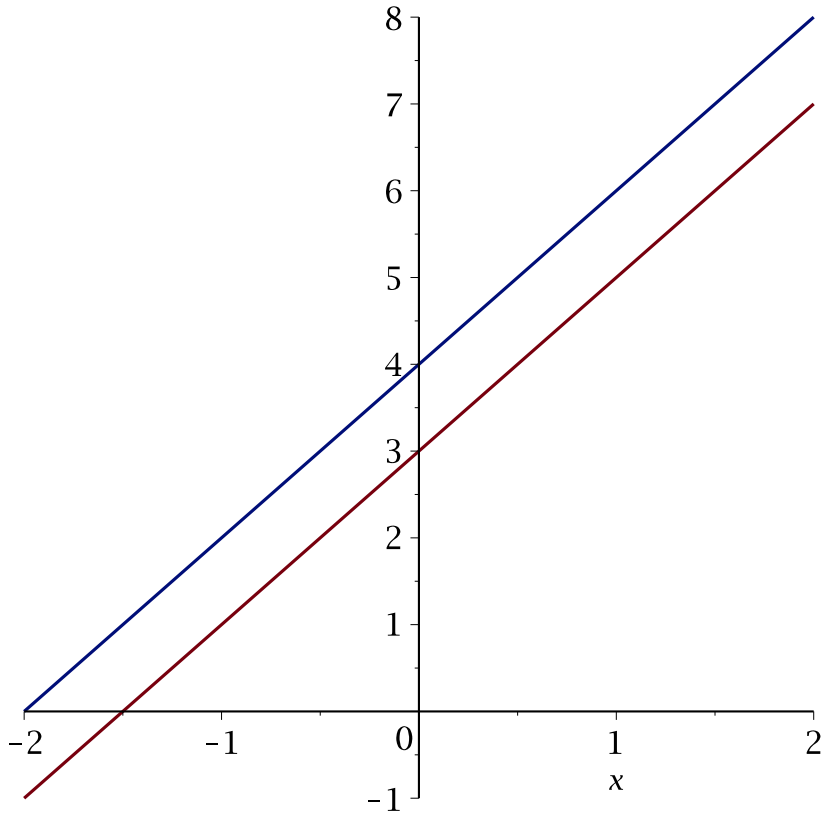
  Das Newtonverfahren dient zur näherungsweisen Nullstellenberechnung von Polynomen. Zur Erzeugung eines Newton-Fractals wird der Iterationsprozess des Newtonverfahrens

  $$z_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)}, \qquad f'(z_n) \neq 0$$

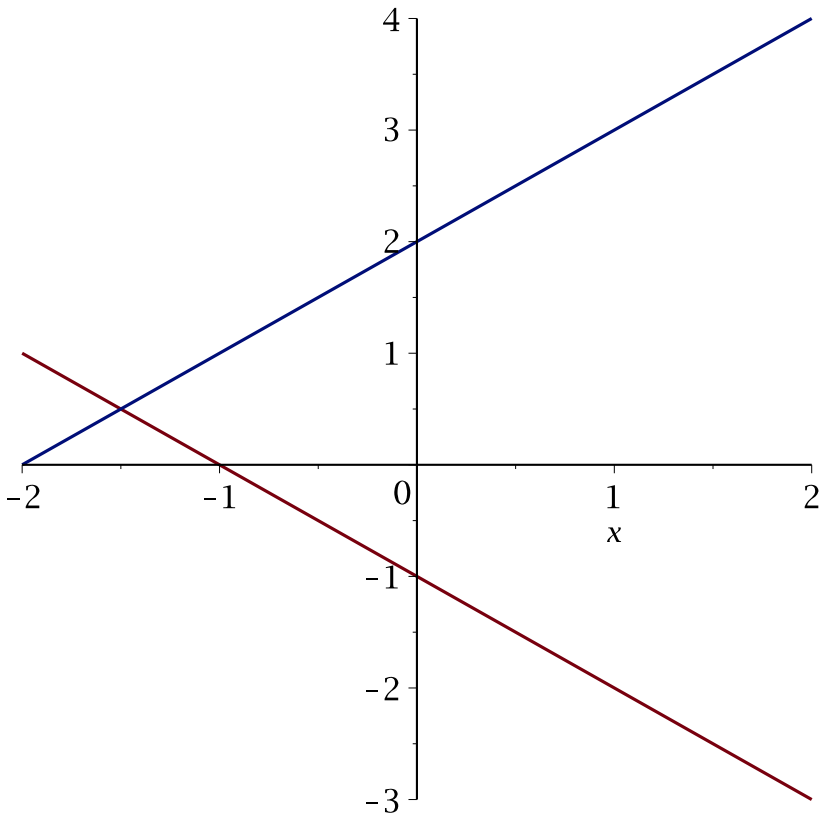
  auf ein konkretes Polynom angewandt.

Beispiele siehe Maple-Datei zu Fraktalen

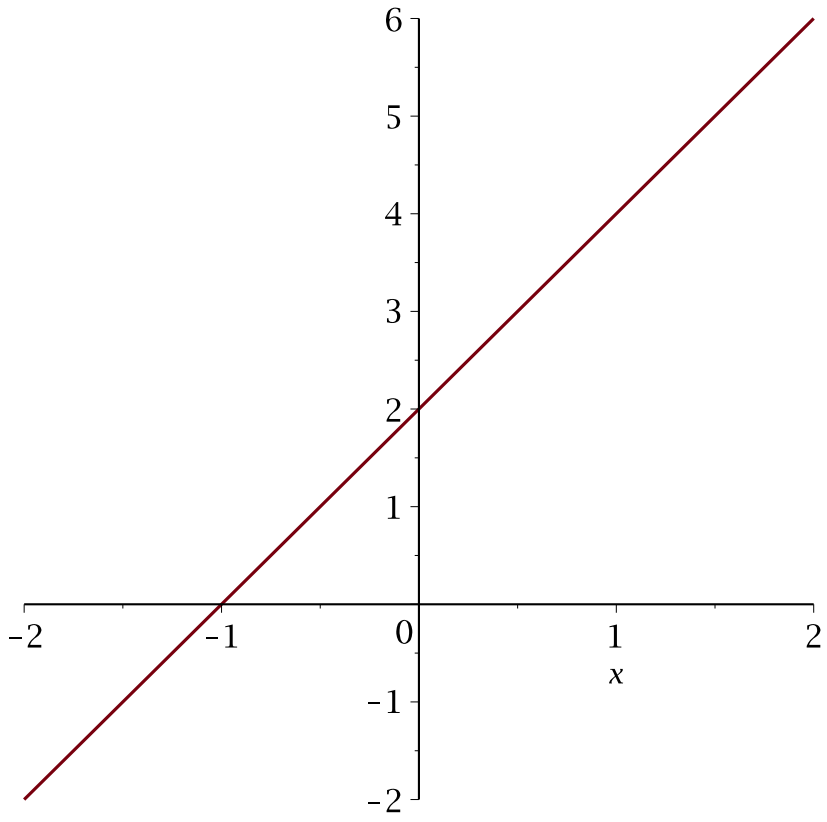
```
> restart:
> with(linalg):
> ###Lineare Gleichungssysteme (2 Gleichungen, 2 Unbekannte)
> ##Unlösbarkeit
> plot({2 x + 4, 2 x + 3}, x=-2..2)
```



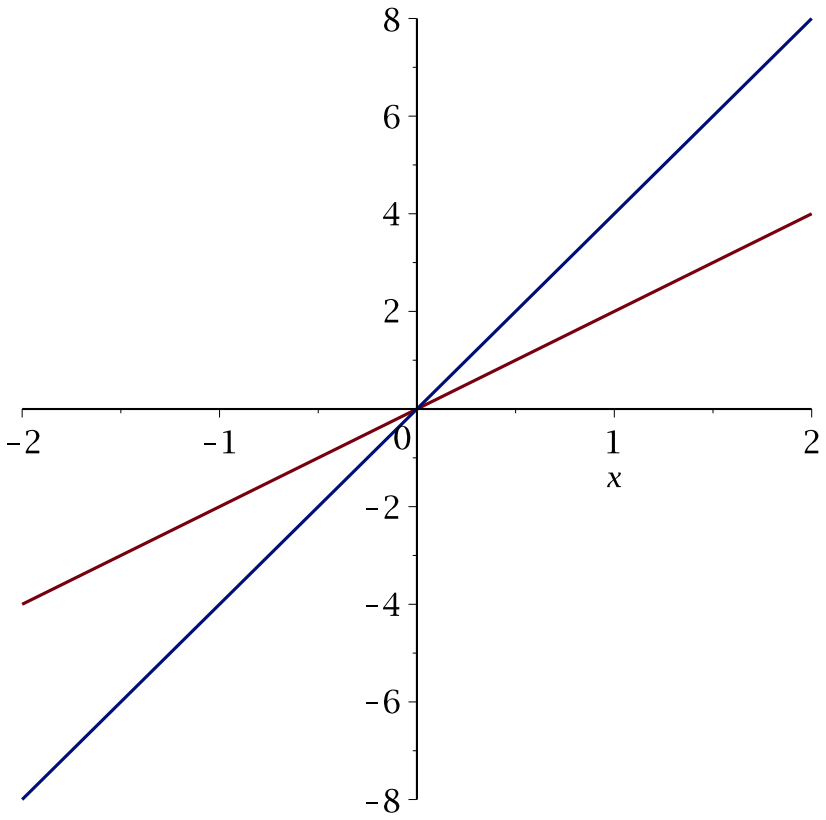

```
> ##Lösbarkeit des inhomogenen Systems
> #genau eine Lösung
> plot({x + 2, -x - 1}, x=-2..2)
```



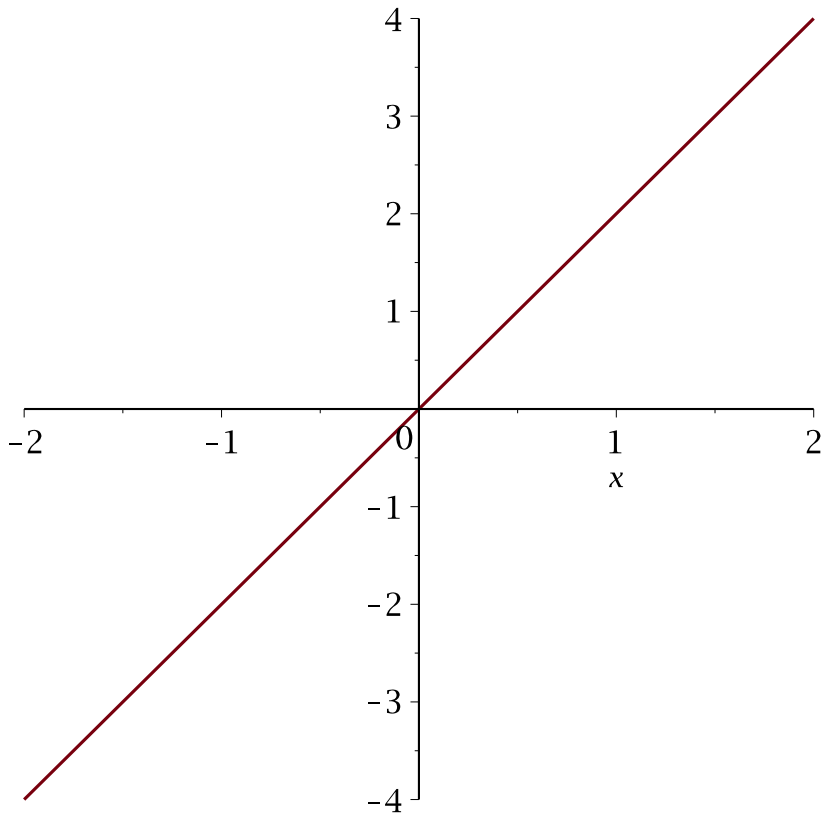

```
> #unendlich viele Lösungen
> plot({2 x + 2, 2 x + 2}, x=-2..2)
```




```
> ##Lösbarkeit des homogenen Systems
> #nur Null-Lösung
> plot({2 x, 4 x}, x=-2..2)
```




```
> #unendlich viele Lösungen
> plot({2 x, 2 x}, x=-2..2)
```




```
> ##Numerische Verfahren
> n := 5 :
> f := (i, j) → evalf(1/n*(sqrt(((2*i-1)
    /(2*n))^2 + (j/n)^2)), 7) :
> A := matrix(n, n, f)
```

$$A := \begin{bmatrix} 0.04472136 & 0.08246212 & 0.1216553 & 0.1612452 & 0.2009976 \\ 0.07211102 & 0.1000000 & 0.1341641 & 0.1708801 & 0.2088062 \\ 0.1077033 & 0.1280625 & 0.1562050 & 0.1886796 & 0.2236068 \\ 0.1456022 & 0.1612452 & 0.1843909 & 0.2126030 & 0.2441312 \\ 0.1843909 & 0.1969772 & 0.2163331 & 0.2408318 & 0.2690724 \end{bmatrix} \quad (1)$$

```
> g := i → evalf(1/3*(sqrt(1 + ((2*i-1)
    /(2*n))^2)^3-((2*i-1)/(2*n))^3),
    7) :
> v := vector(n, g)
```

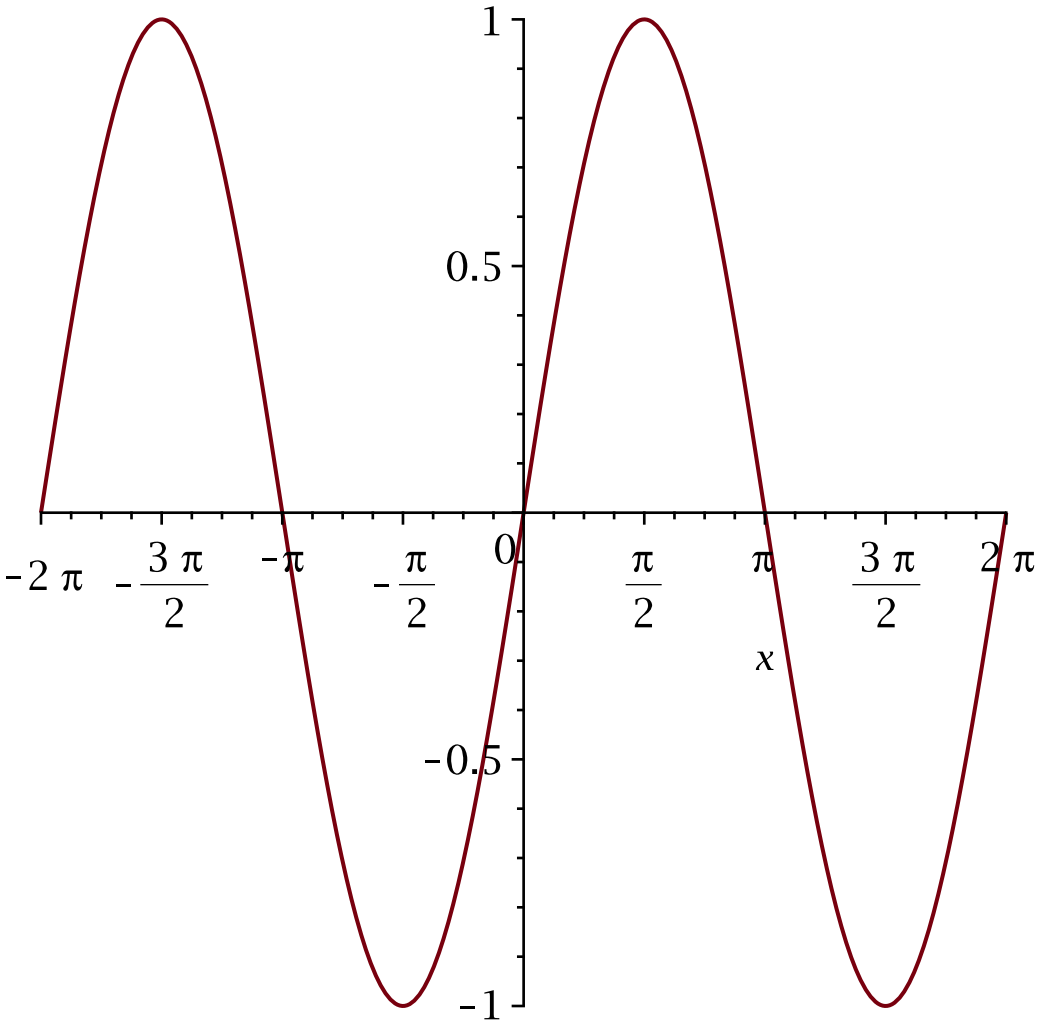
$$v := \left[\begin{array}{ccccc} 0.3380127 & 0.3703312 & 0.4241807 & 0.4919259 & 0.5687017 \end{array}\right] \quad (2)$$

```
> x := linsolve(A, v)
```

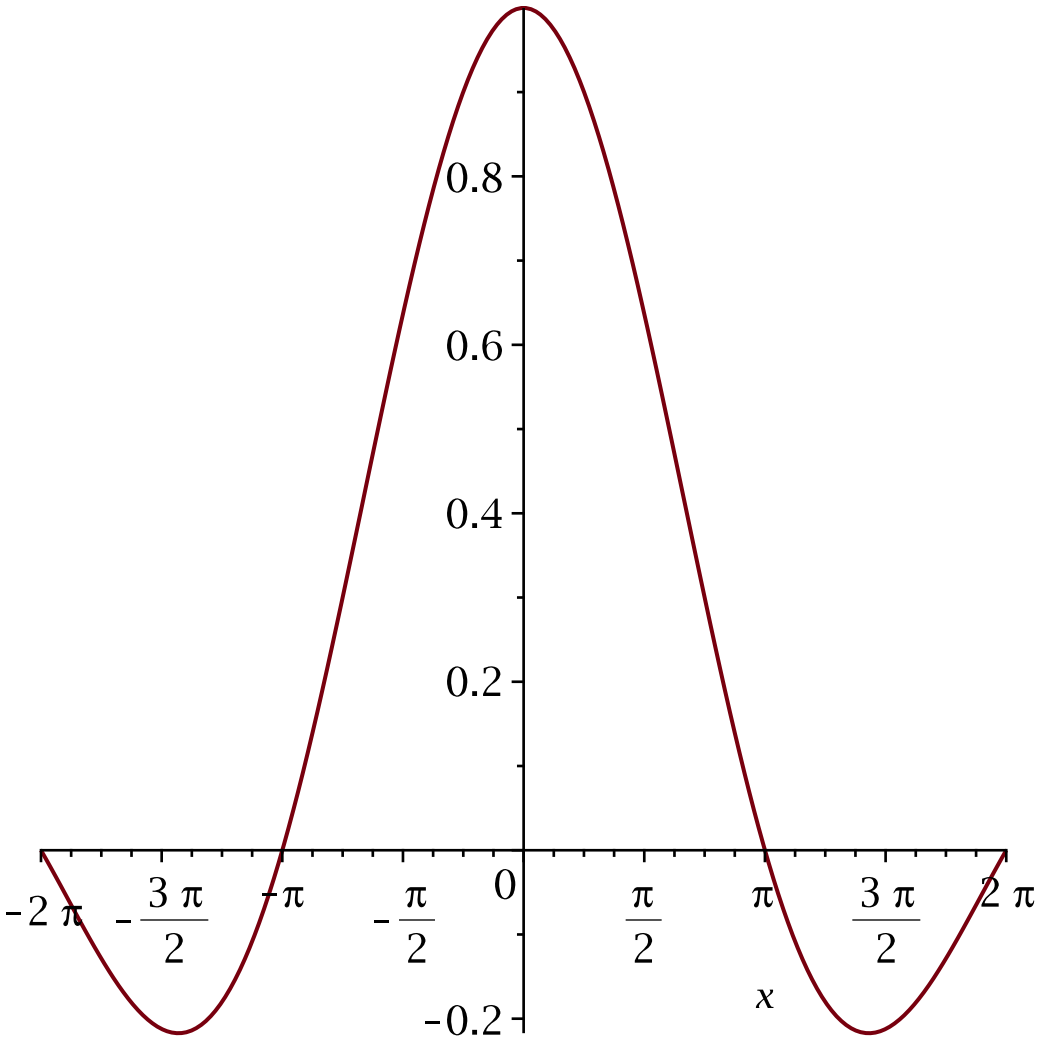
$$x := [0.3045428972, 0.04135941970, 1.237006504, 0.3426002733, 0.573397370] \quad (3)$$

```
>
```

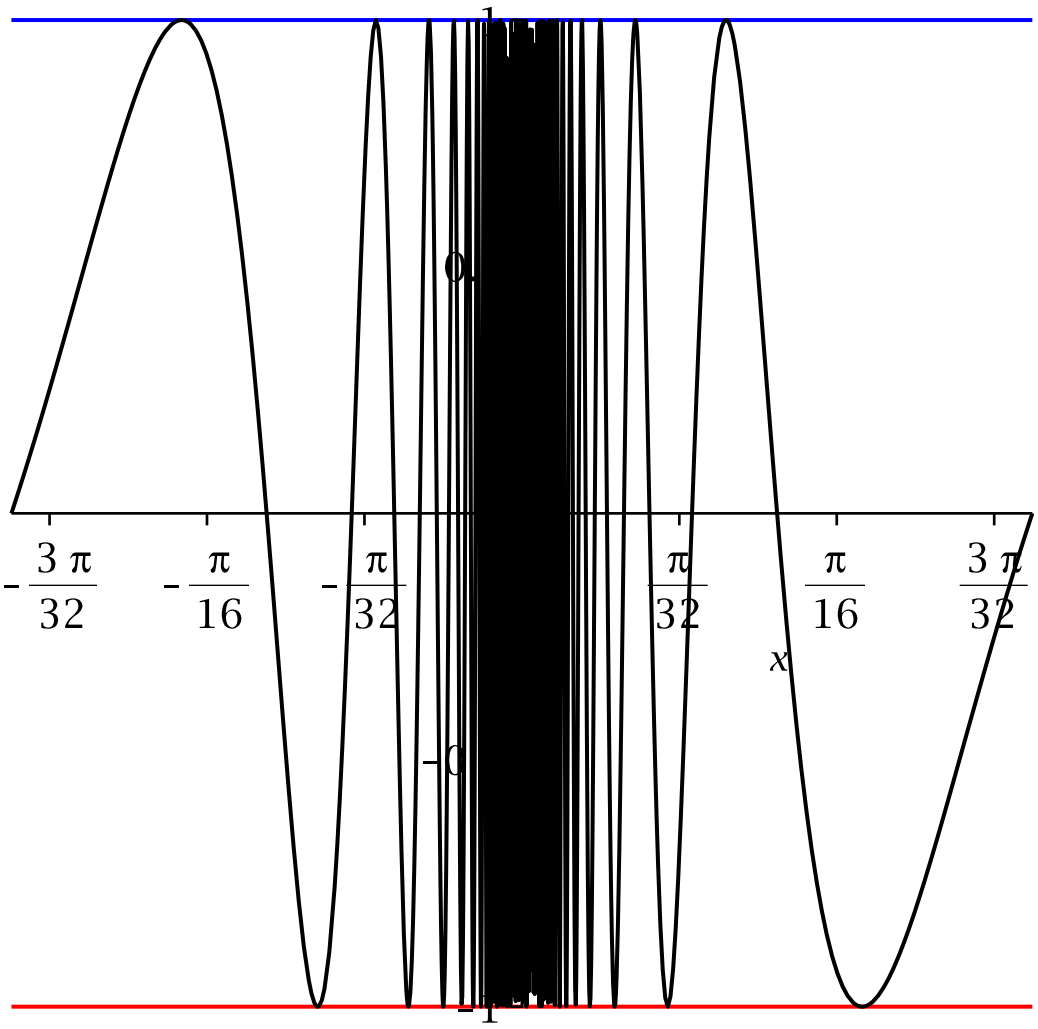
```
> restart:
> with(plots):
> ##Funktionen einer Variablen
> #Trigonometrische Funktionen
> plot(sin(x), x=-2 Pi..2 Pi)
```



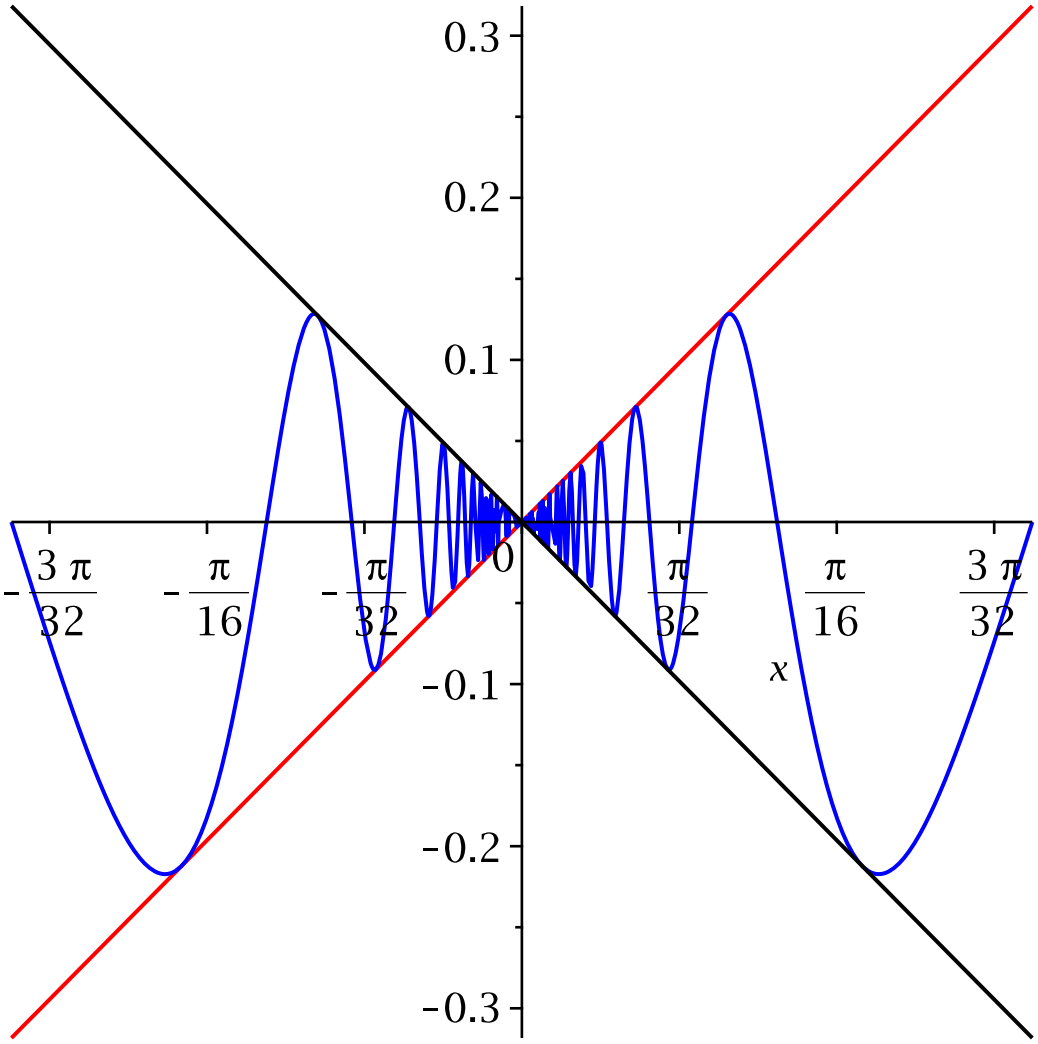

```
> plot(sin(x)/x, x=-2 Pi..2 Pi)
```



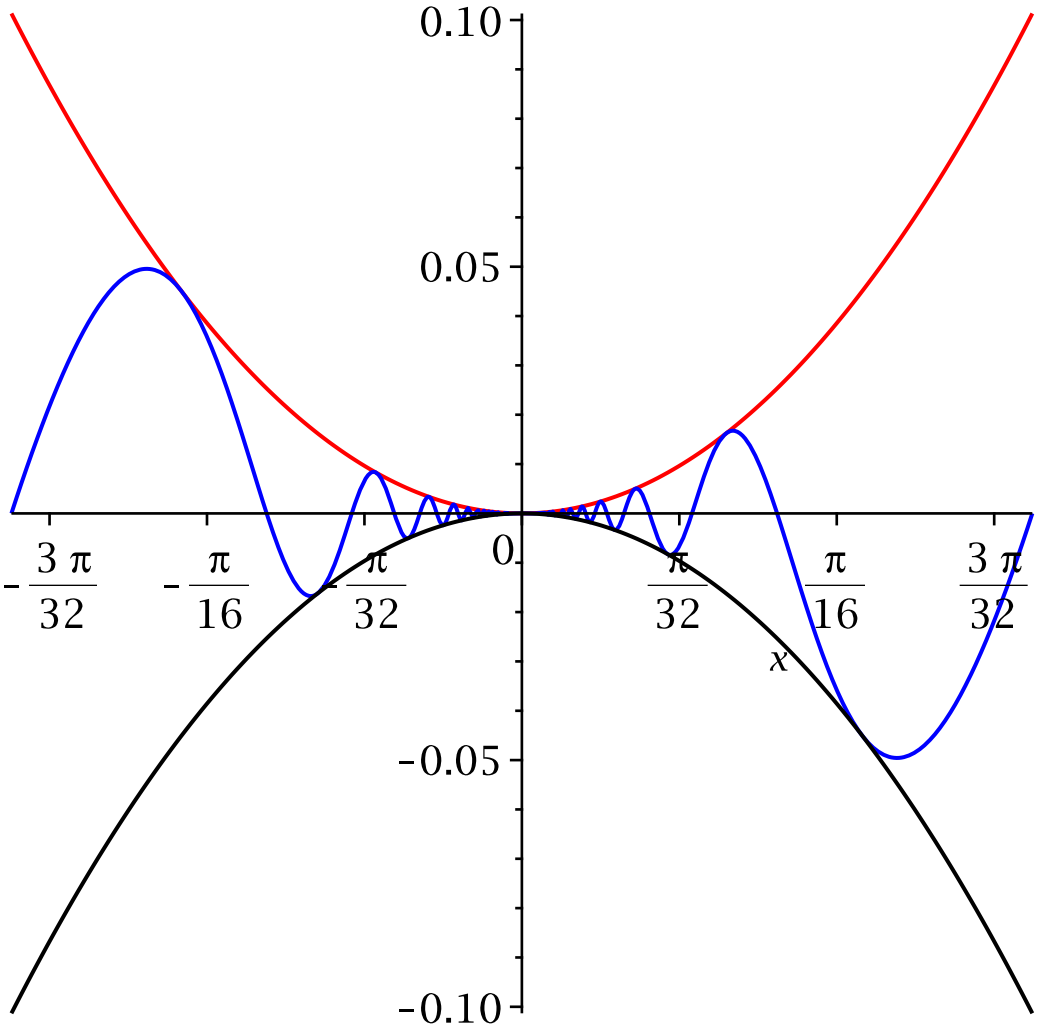

```
> plot({sin(1/x), 1, -1}, x=-1/Pi..1
  /Pi, color=[RED, BLUE, BLACK])
```



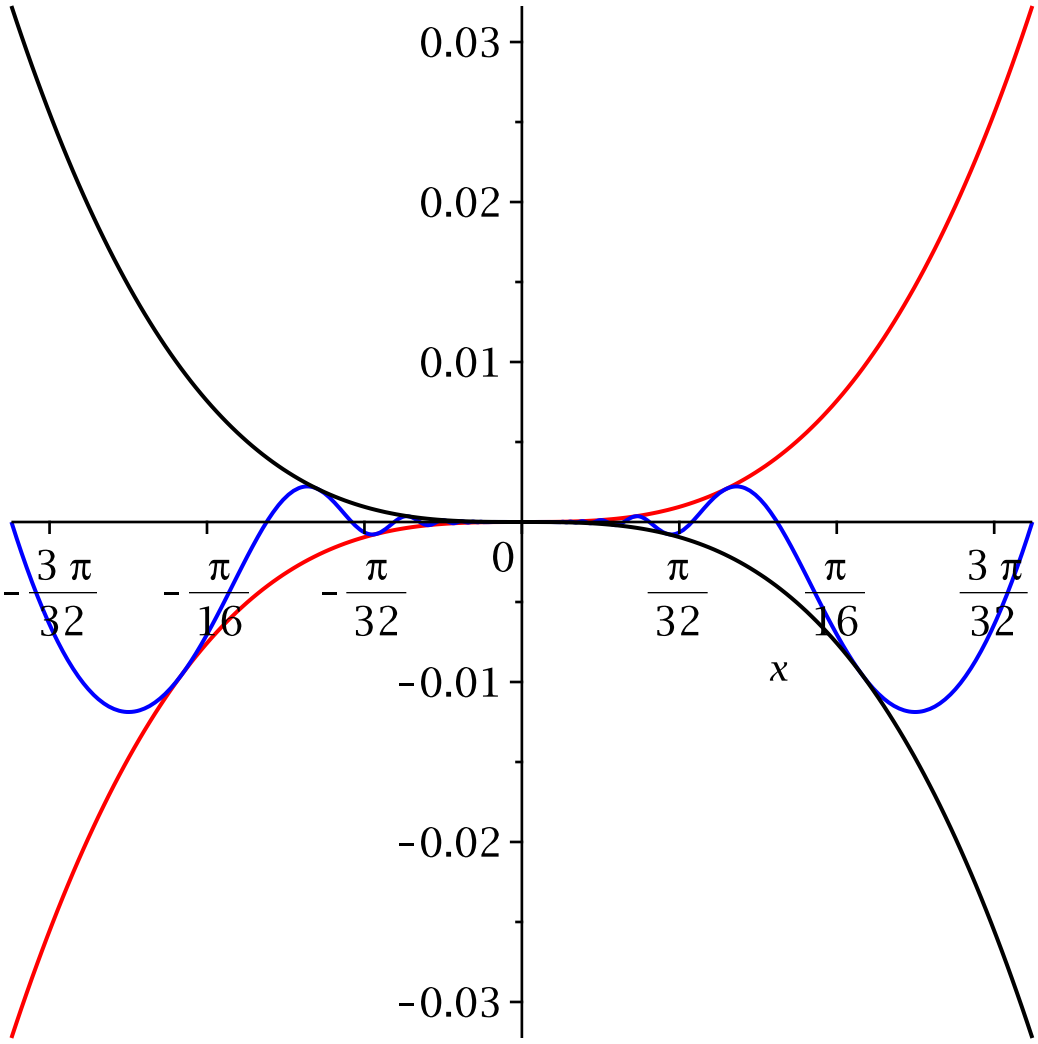

```
> plot({x sin(1/x), x, -x}, x=-1/Pi..1
  /Pi, color=[RED, BLUE, BLACK])
```



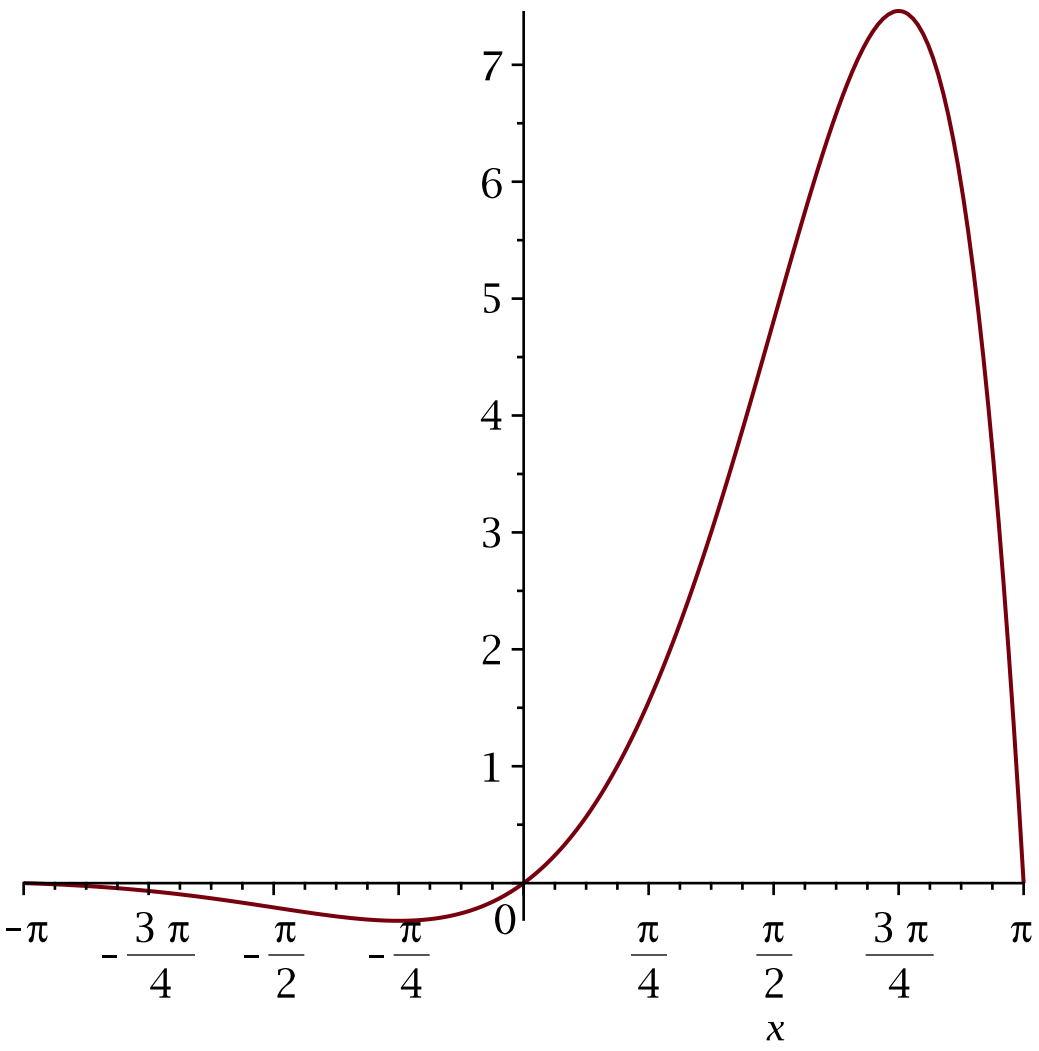

```
> plot({x^2 sin(1/x), x^2, -x^2}, x=-1/Pi
  ..1/Pi, color=[RED, BLUE, BLACK])
```



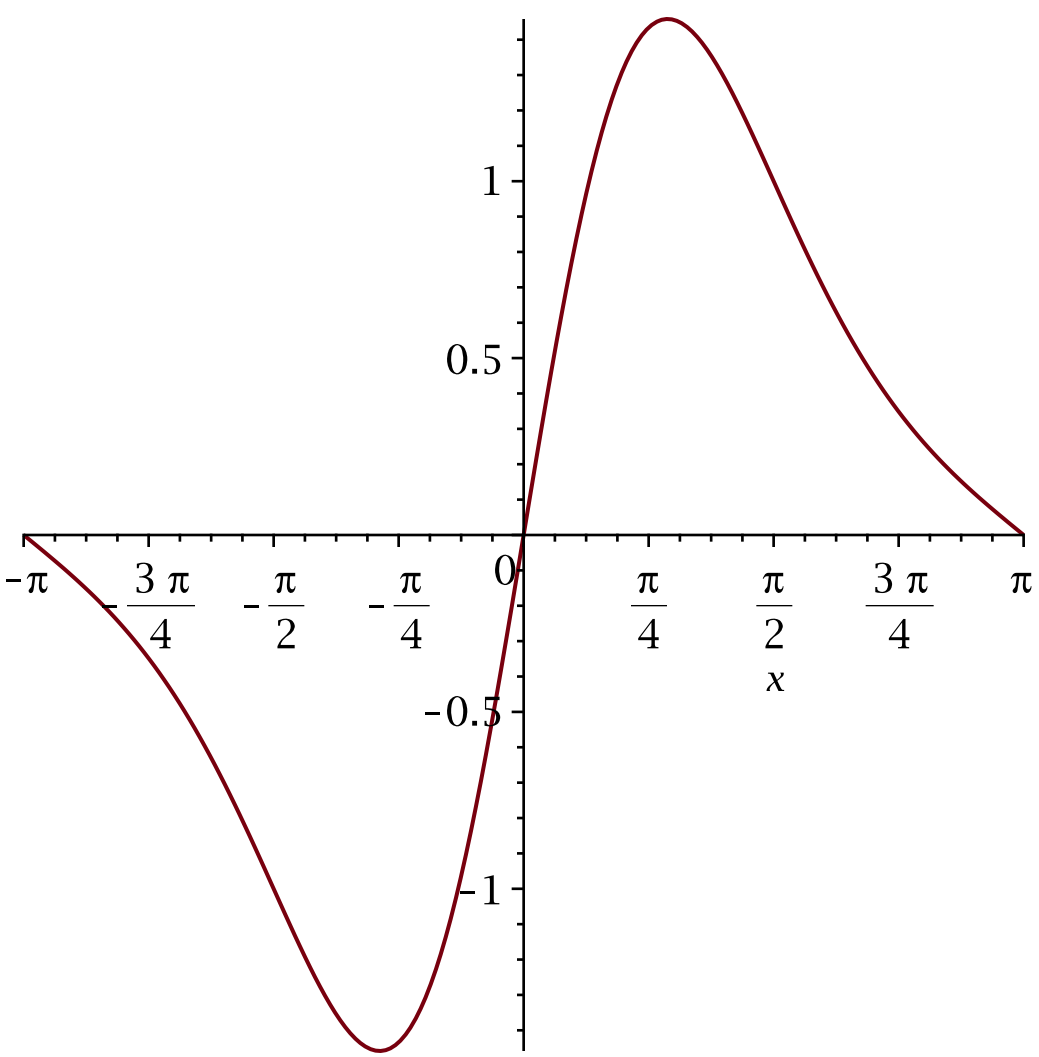

```
> plot({x^3 sin(1/x), x^3, -x^3}, x=-1/Pi
  ..1/Pi, color=[RED, BLUE, BLACK])
```



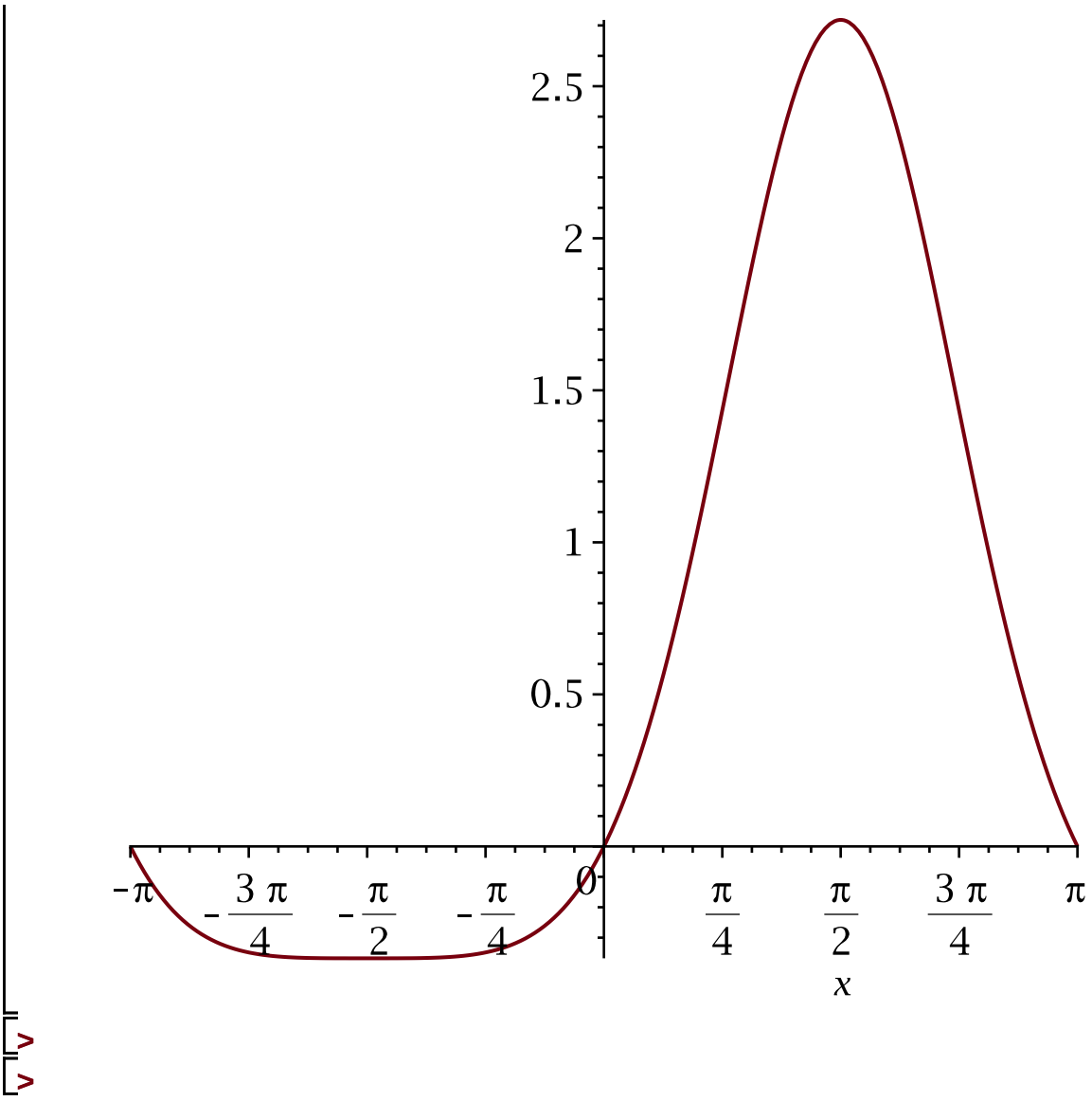

```
> #Produkte trigonometrischer mit
  Exponentialfunktionen
> plot(exp(x) sin(x), x=-Pi..Pi)
```



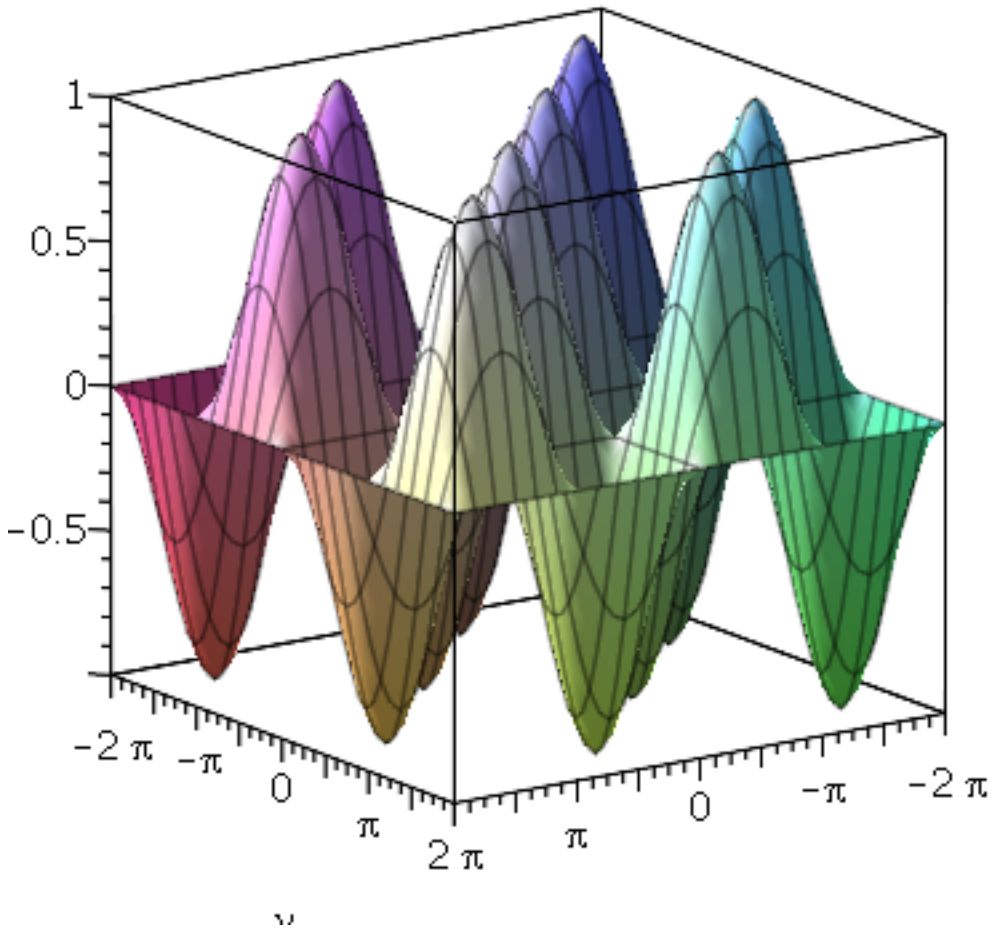

```
> plot(e^cos(x) sin(x), x=-Pi..Pi)
```



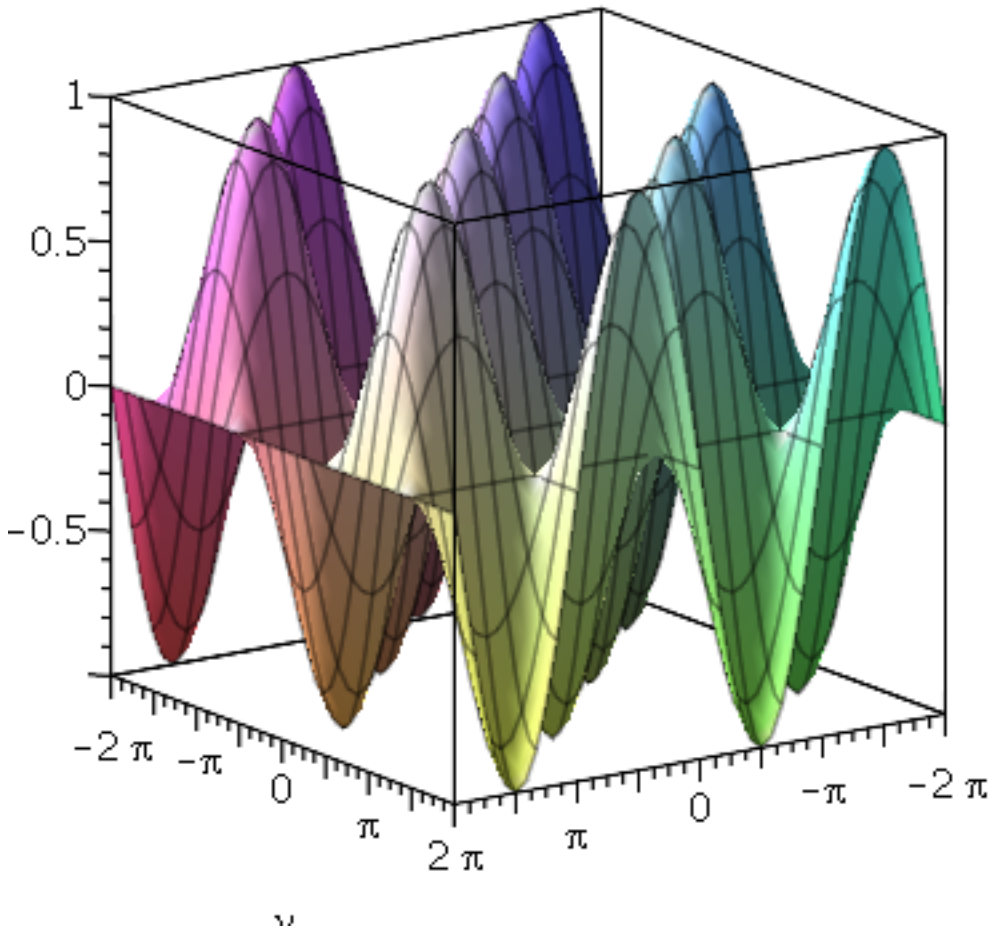

```
> plot(e^sin(x) sin(x), x=-Pi..Pi)
```


2.5
2
1.5
1
0.5
−π
−3 π/4
−π/2
−π/4
0
π/4
π/2
3 π/4
π
x

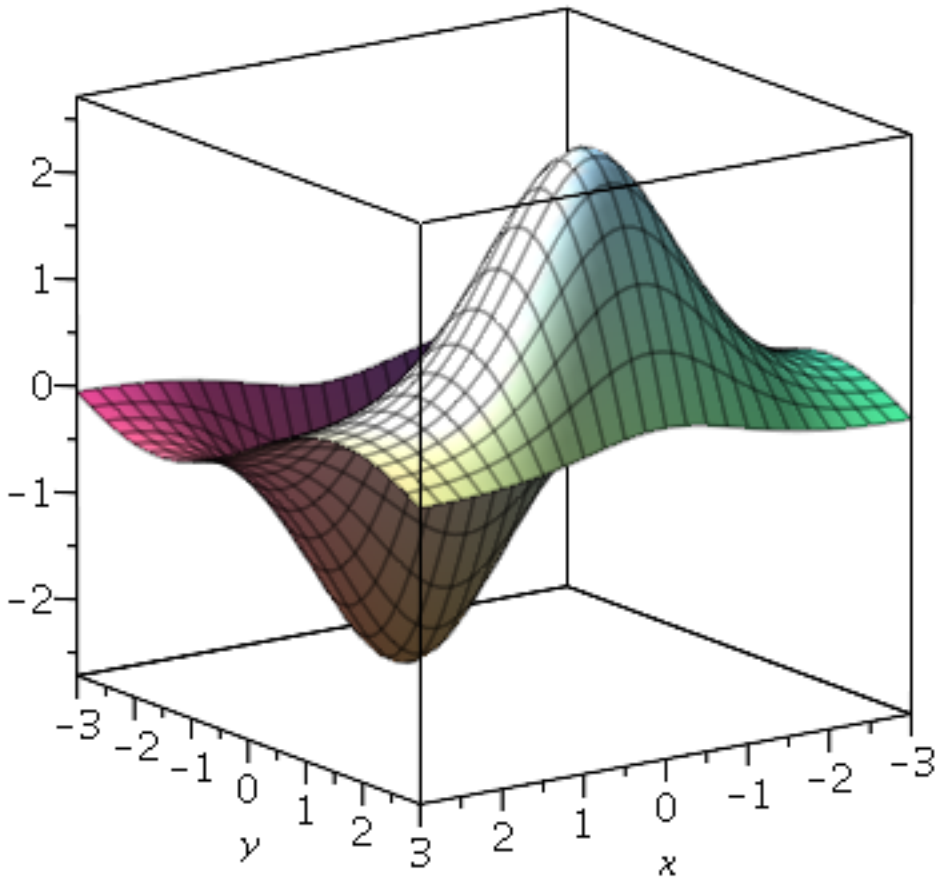
```
> restart:
> with(plots):
> ##Funktionen zweier Variablen
> #Trigonometrische Funktionen
> plot3d(sin(x)sin(y), x=-2 Pi..2 Pi, y=
    -2 Pi..2 Pi, axes=boxed)
```



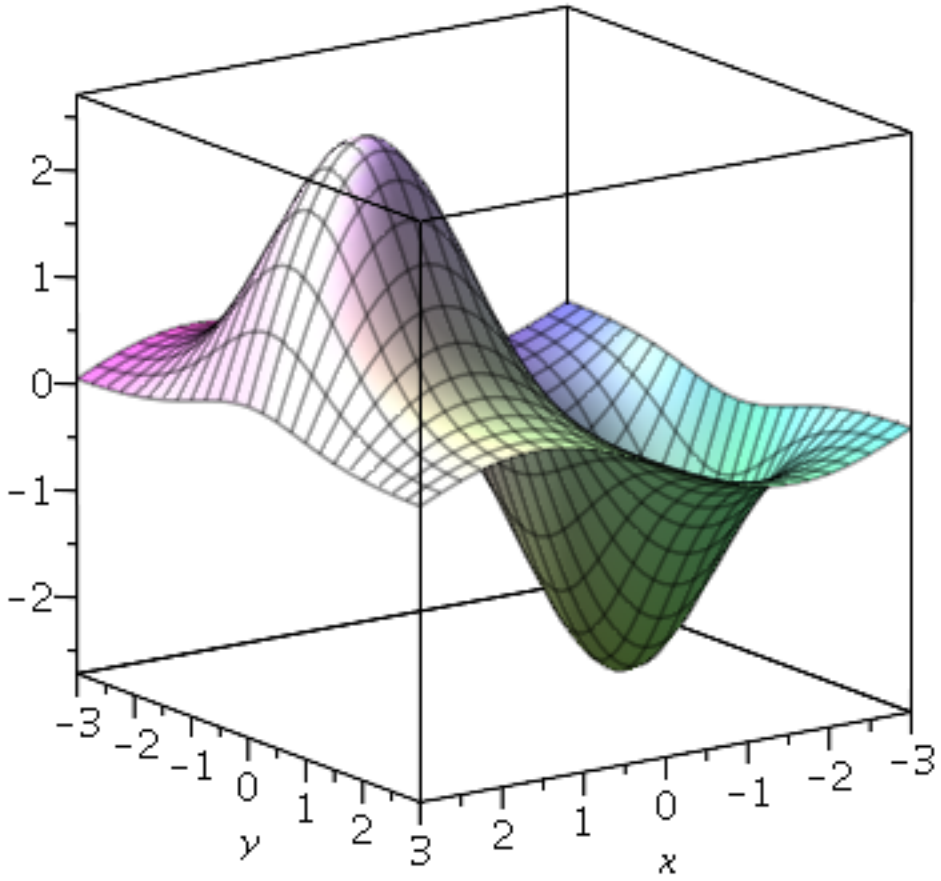

```
> plot3d(sin(x)cos(y), x=-2 Pi..2 Pi, y=
    -2 Pi..2 Pi, axes=boxed)
```



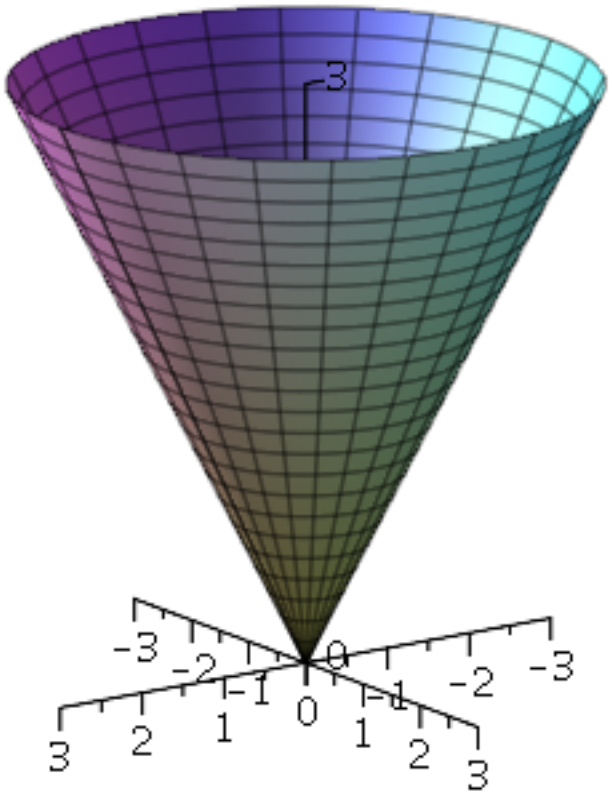

```
> #Produkte trigonometrischer mit
  Exponentialfunktionen
> plot3d(sin(y) exp(cos(x)), x=-3..3, y=
    -3..3)
```



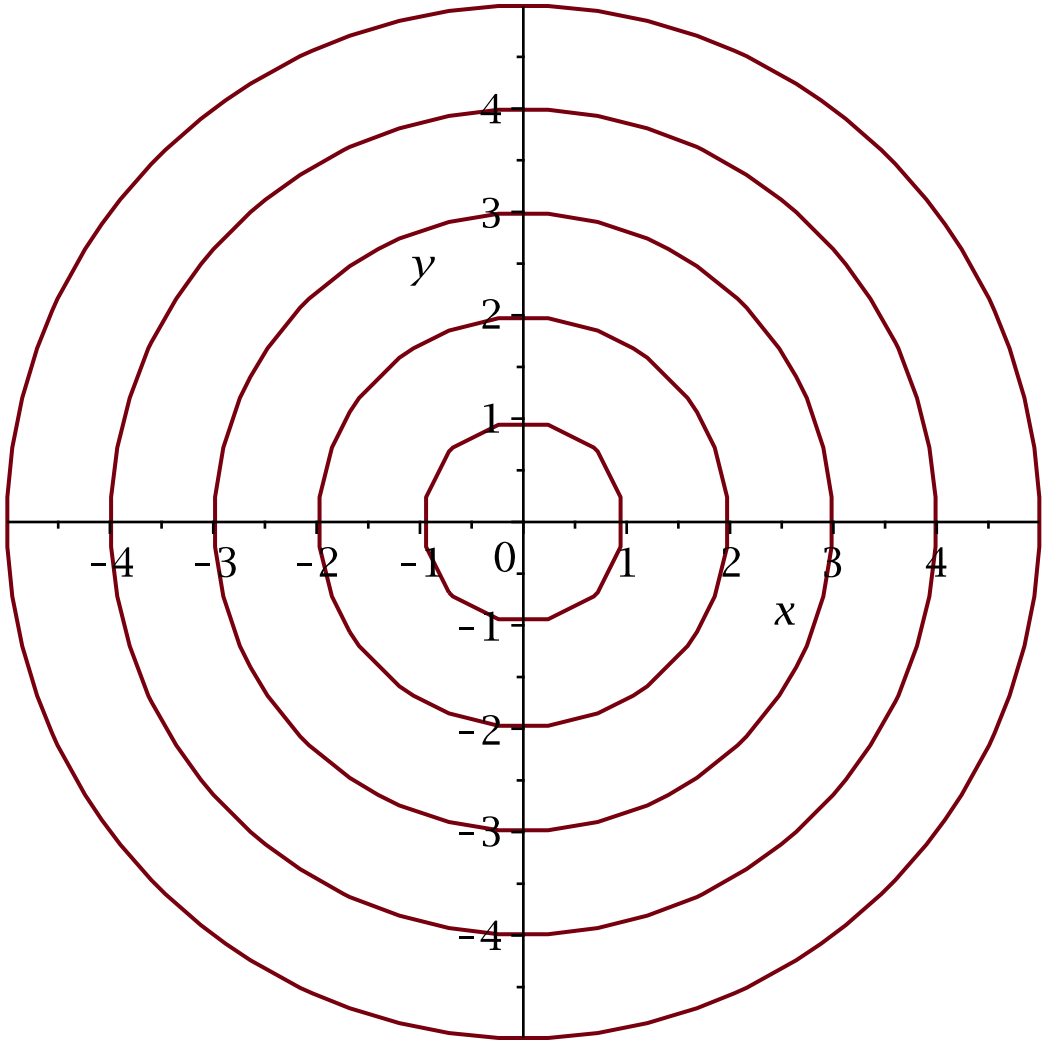

```
> plot3d(sin(x) exp(cos(y)), x=-3..3, y=
    -3..3)
```




```
> #Kreiskegel
> plot3d([u cos(v), u sin(v), u], u=0
    ..3, v=0..2 Pi, axes=normal)
```

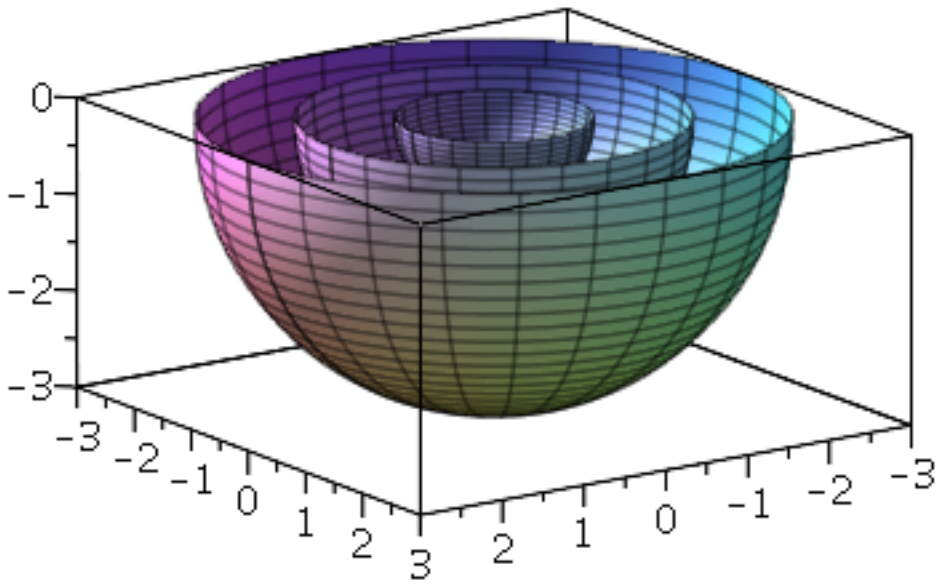



```
> ##Skalarfelder
> #Ebenes Skalarfeld
> implicitplot({x^2+y^2=1, x^2+y^2=4, x^2+y^2=9, x^2+y^2=16, x^2+y^2=25}, x=-6..6, y=-6..6, scaling=constrained)
```



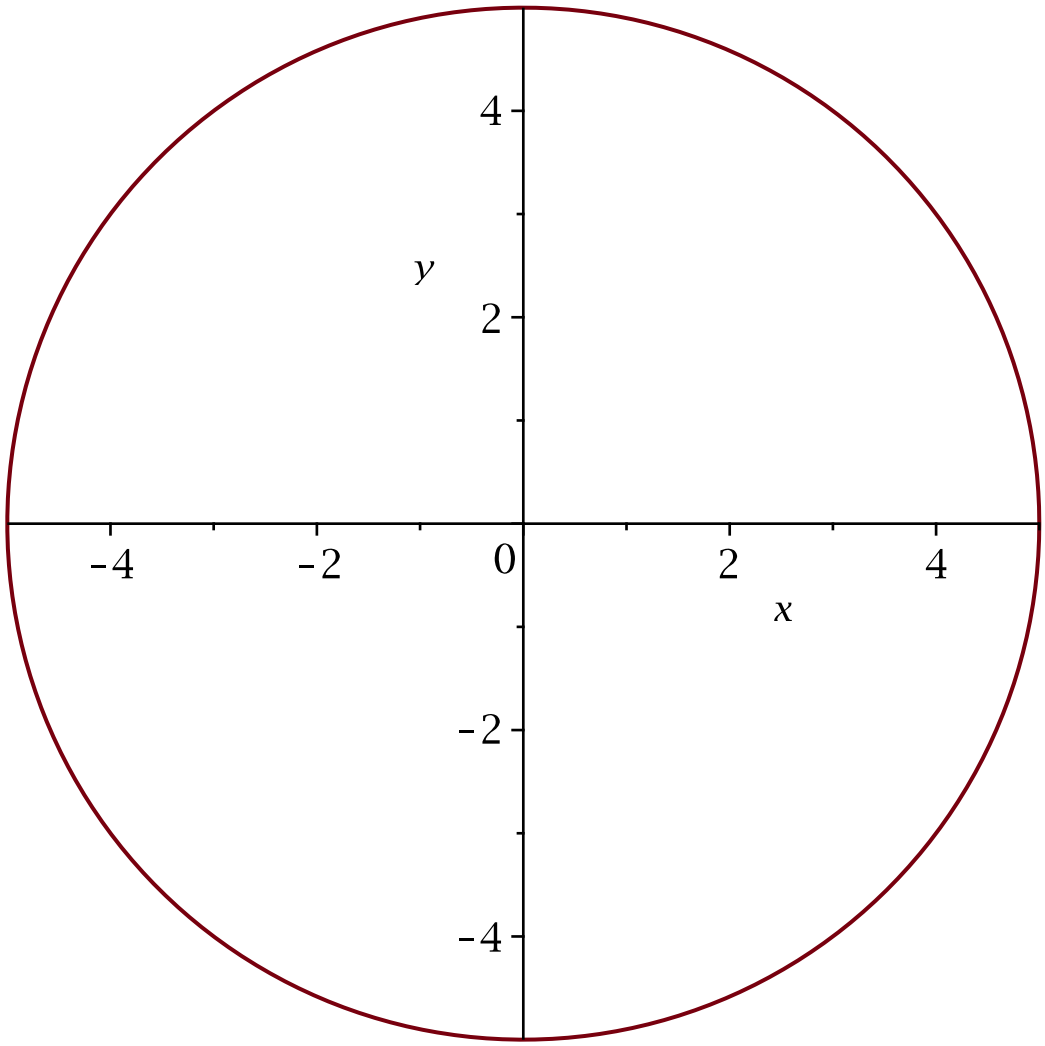

```
> #Räumliches Skalarfeld
> plot3d({[cos(s) sin(t), sin(s) sin(t),
    cos(t)], [2 cos(s) sin(t),
   2 sin(s) sin(t), 2 cos(t)],
   [3 cos(s) sin(t), 3 sin(s) sin(t),
    3 cos(t)]}, s = 0 ..2 Pi, t = Pi/2 ..Pi,
```

```
scaling = constrained
```



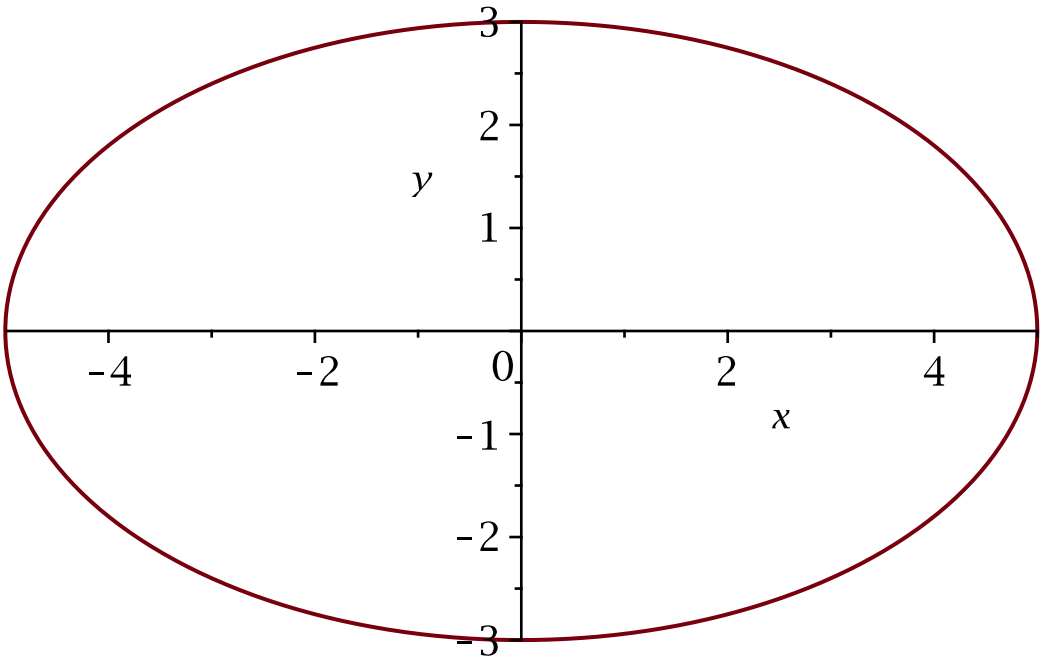

```
>
>
>
```

```
> restart:
> with(plots):
> ###Vektorfunktionen einer Variablen
> ##Ebene Kurven
> #Kreis
> plot([5 cos(t), 5 sin(t), t = 0..2 Pi],
     labels = ['x', 'y'], scaling
    = constrained)
```



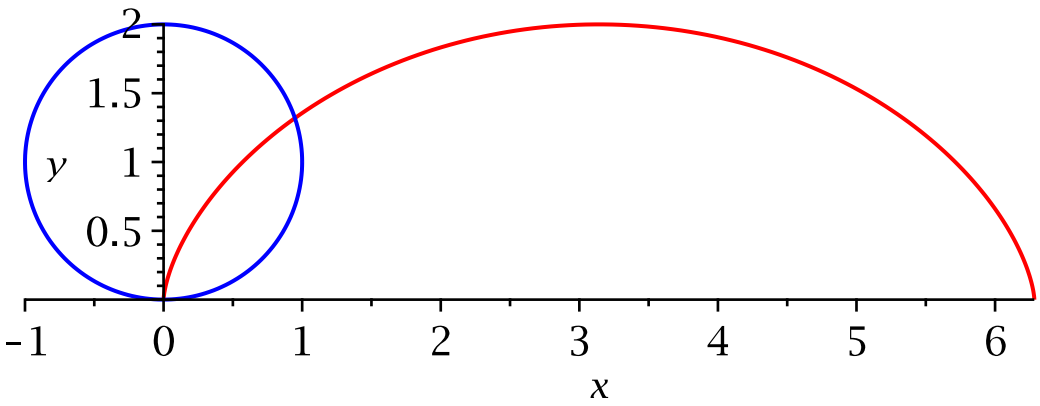

```
> #Ellipse
> plot([5 cos(t), 3 sin(t), t = 0..2 Pi],
     labels = ['x', 'y'], scaling
```

```
    = constrained)
```

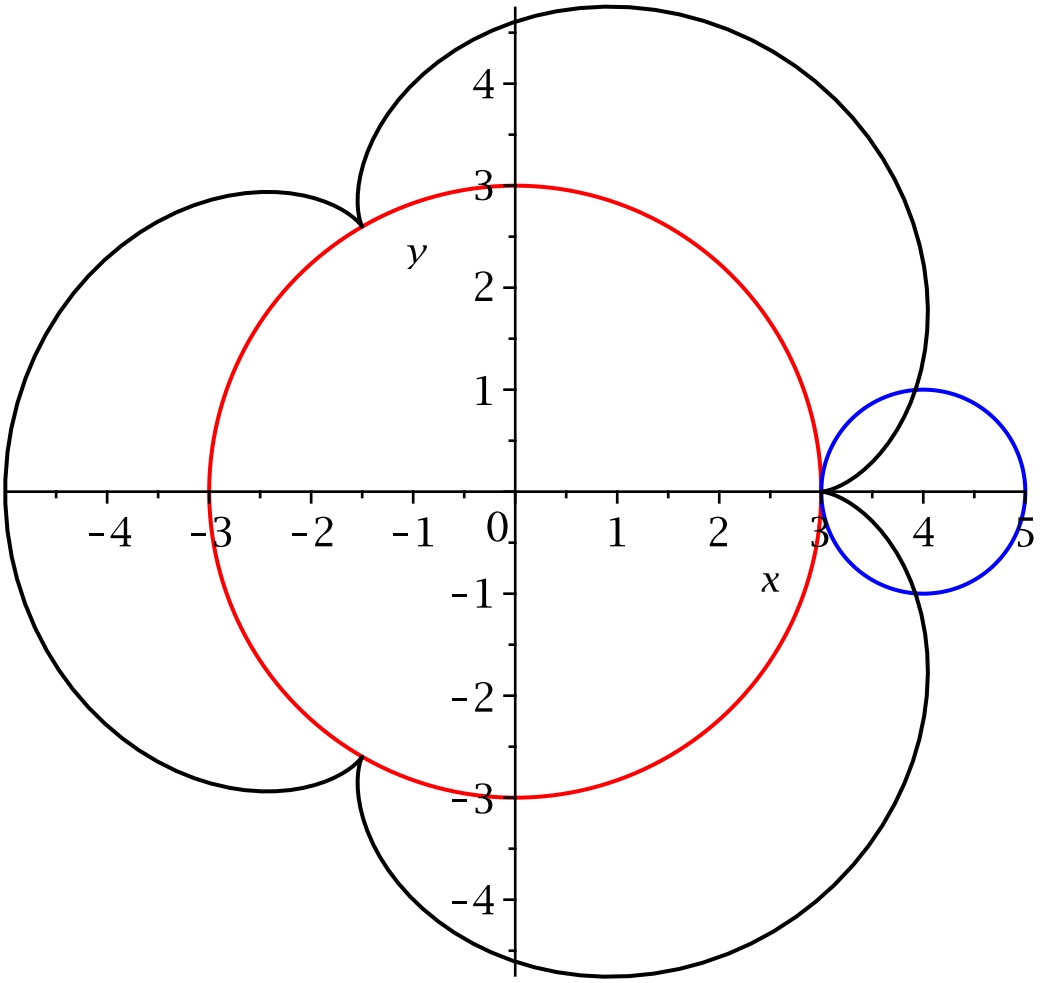



```
> restart:
> with(plots):
> A:=3:
> a:=1:
> m:=A/a:
> #Gewöhnliche Zykloide
> plot({[a (t - sin(t)), a (1 - cos(t)), t
    = 0..2 Pi], [a cos(t), a + a sin(t),
     t = 0..2 Pi]}, scaling = constrained,
     color = [RED, BLUE], labels = ['x', 'y
```

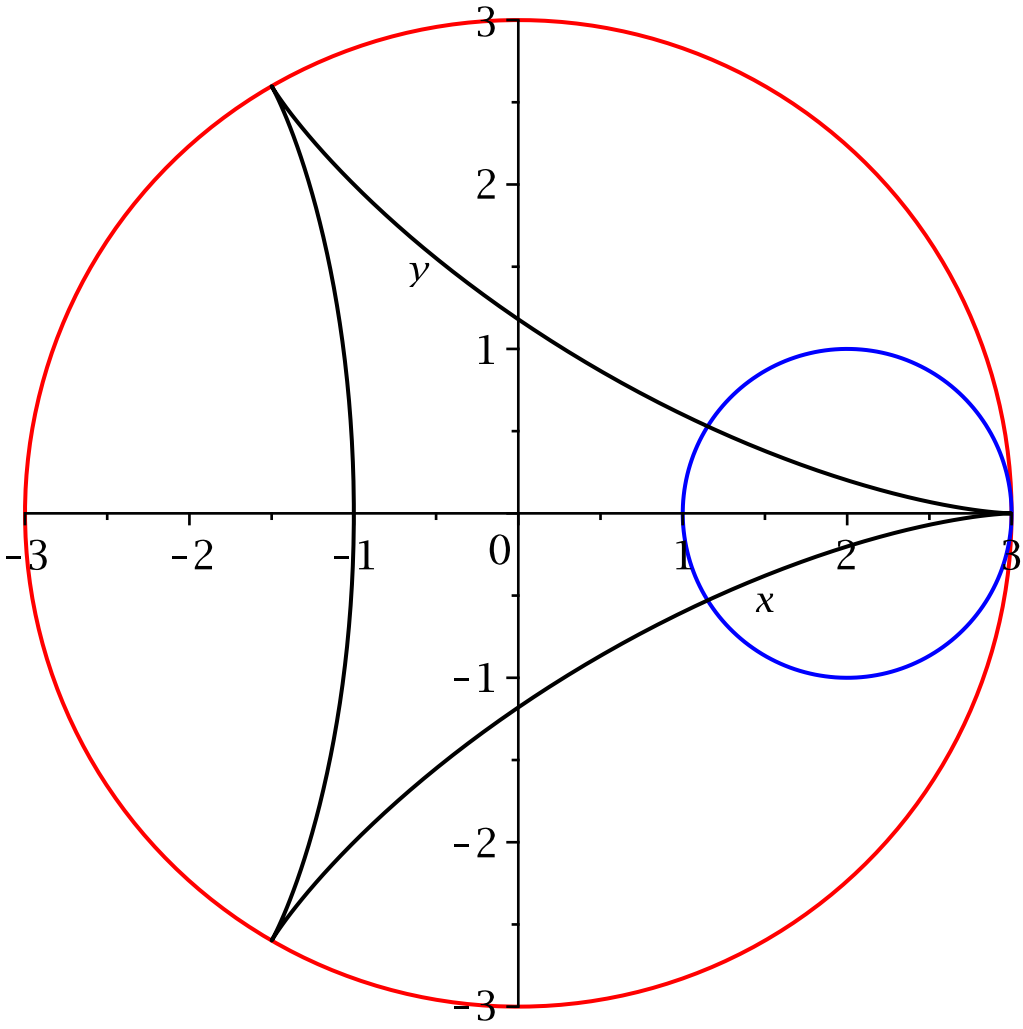
```
    ']):
```



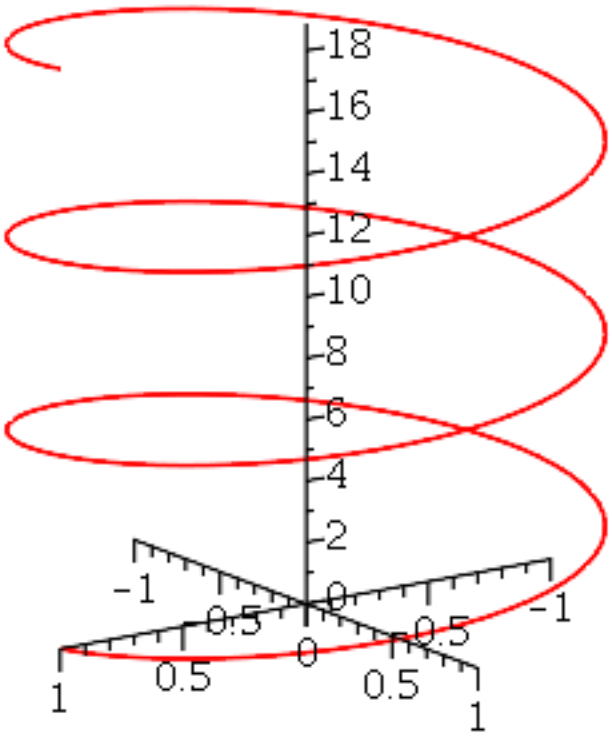

```
> #Epizykloiden
> plot({[(A + a) cos(t) - a cos((m
    + 1) t), (A + a) sin(t) - a sin((m
    + 1) t), t = 0..2 Pi], [A cos(t),
     A sin(t), t = 0..2 Pi], [(A + a)
    + a cos(t), sin(t), t = 0..2  Pi]},
     color = [RED, BLUE, BLACK], labels
    = ['x', 'y'], scaling = constrained)
```



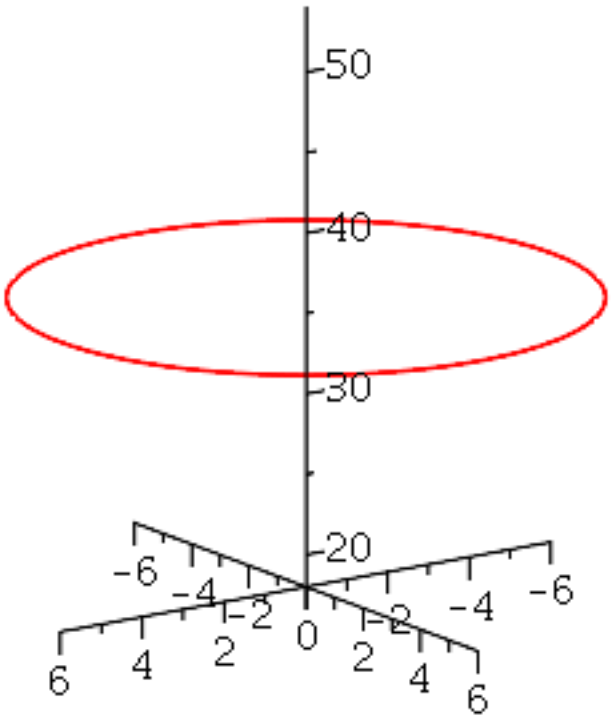

```
> #Hypozykloiden
> plot({[(A-a) cos(t) + a cos((m-1) t),
     (A-a) sin(t) - a sin((m-1) t), t = 0
    ..2 Pi], [A cos(t), A sin(t), t = 0
    ..2 Pi], [(A-a) + a cos(t), sin(t), t
    = 0..2  Pi]}, color = [RED, BLUE,
     BLACK], labels = ['x', 'y'], scaling
    = constrained)
```



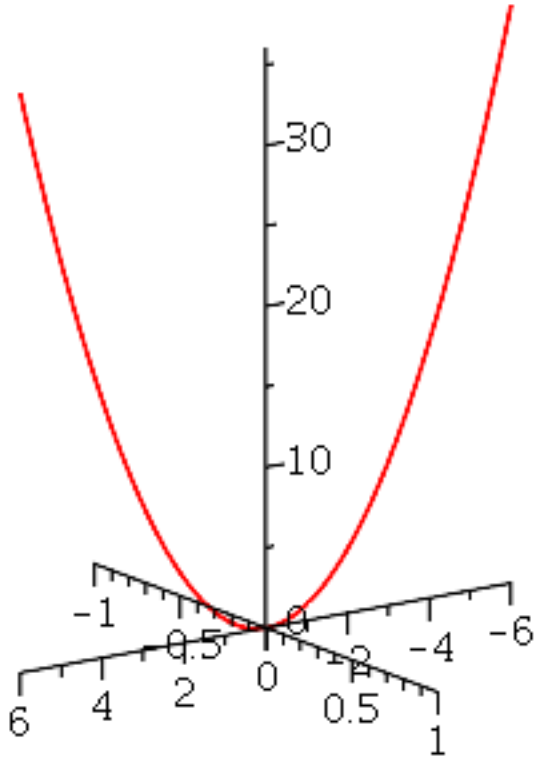

```
> ##Raumkurven
> #Schraubenlinie
> spacecurve([a cos(t), a sin(t), t], t = 0
    ..6 Pi, axes = normal, color = RED)
```



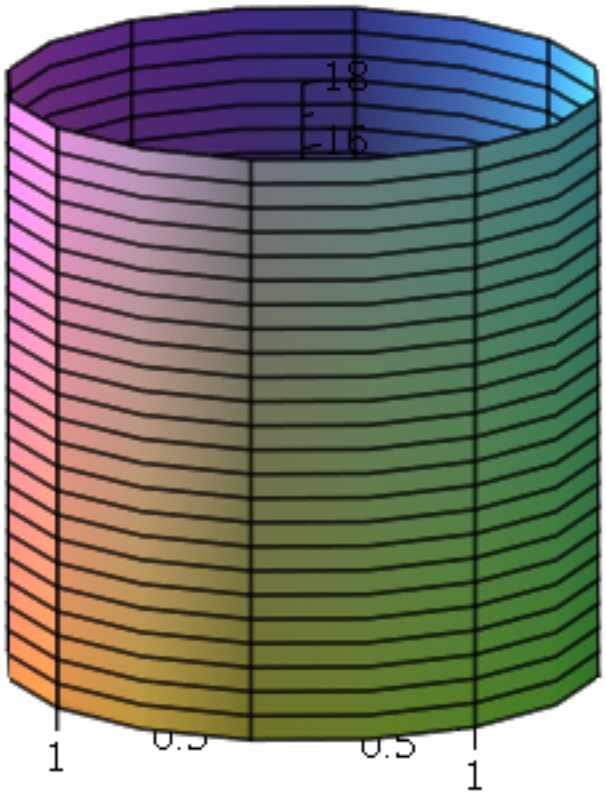

```
> #Kreis im Raum
> spacecurve([6 cos(v), 6 sin(v), 36], v
    = 0 ..2 Pi, axes = normal, color = RED)
```



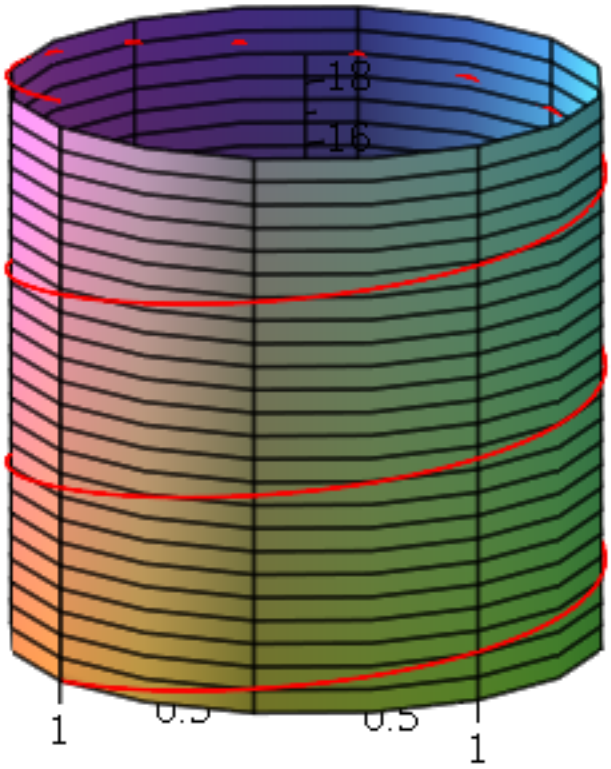

```
> #Parabel im Raum
> spacecurve([u cosh(0), u sinh(0), u^2],
    u=-6..6, color=RED, axes=normal)
```



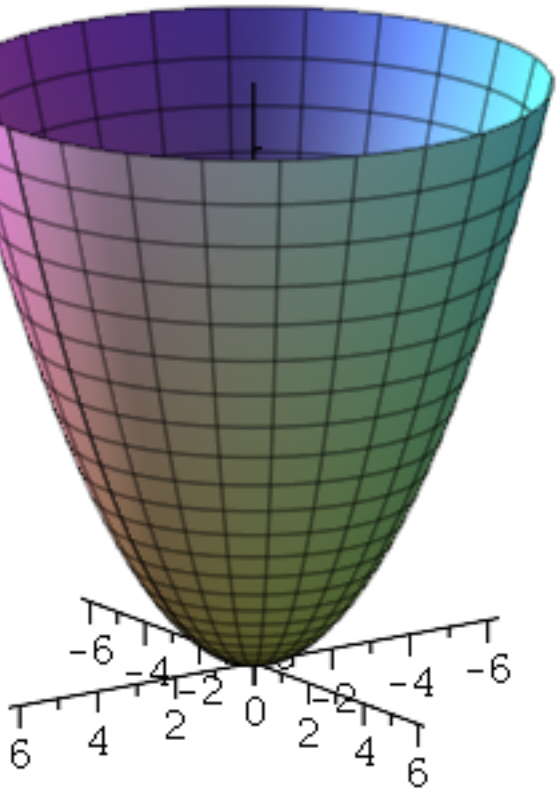

```
> ##Kurven auf Flächen
> #Zylinder
> plot3d([a cos(t), a sin(t), z], t = 0
    ..6 Pi, z = 0 ..18, axes = normal)
```




```
> #Schraubenlinie auf einem Zylinder
> a1 := spacecurve([a cos(t), a sin(t),
    t], t = 0..6 Pi, axes = normal, color
    = RED):
> b1 := plot3d([a cos(t), a sin(t), z], t
    = 0..6 Pi, z = 0..18, axes = normal):
> display([a1, b1])
```

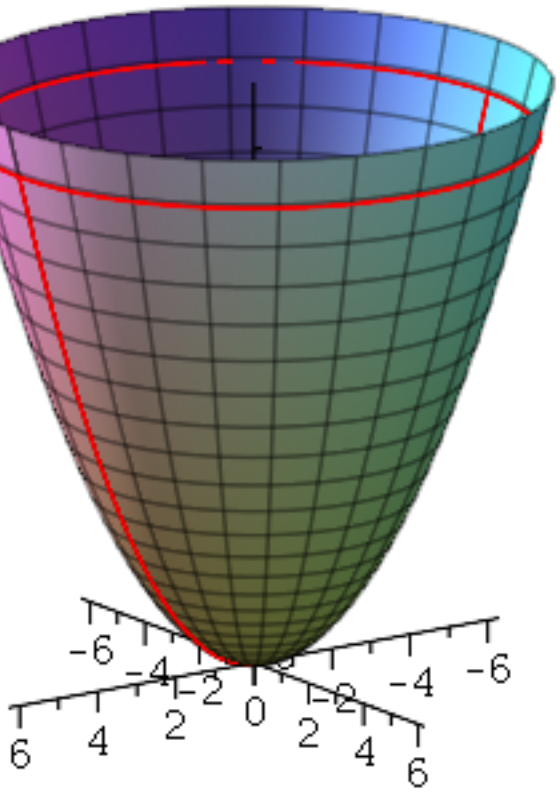



```
> #Rotationsparaboloid
> plot3d([u cos(v), u sin(v), u^2], u = 0
    ..2  Pi, v = 0 ..2*Pi, axes = normal)
```

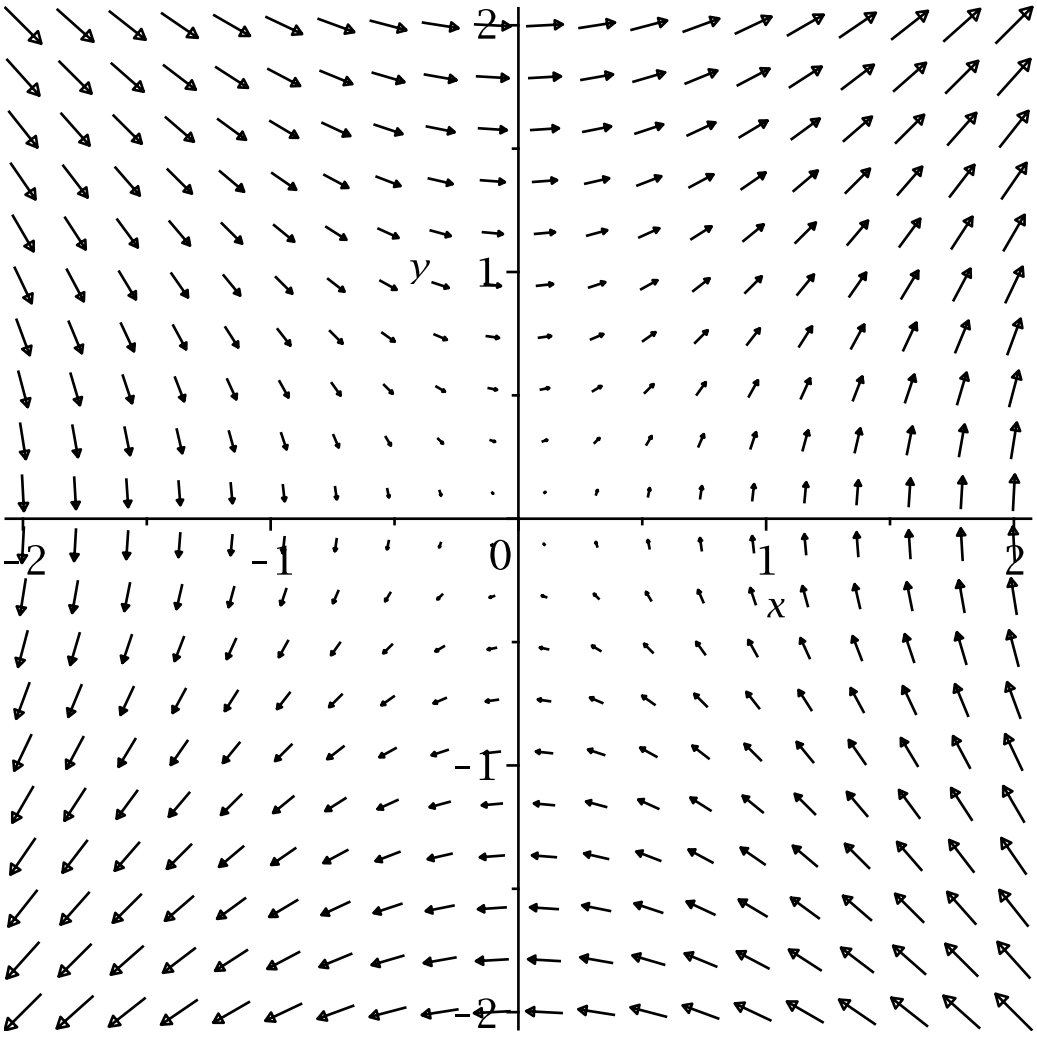



```
> #Kreis und Parabel auf einem
  Rotationsparaboloiden
> a2 := spacecurve([6 cos(v), 6 sin(v),
    36], v = 0..2 Pi, axes = normal, color
    = RED):
> b2 := spacecurve([u cosh(0), u sinh(0),
    u^2], u = -6..6, color = RED, axes
    = normal):
> c2 := plot3d([u cos(v), u sin(v), u^2], u
    = 0..2 Pi, v = 0..2 Pi, axes = normal)
    :
```

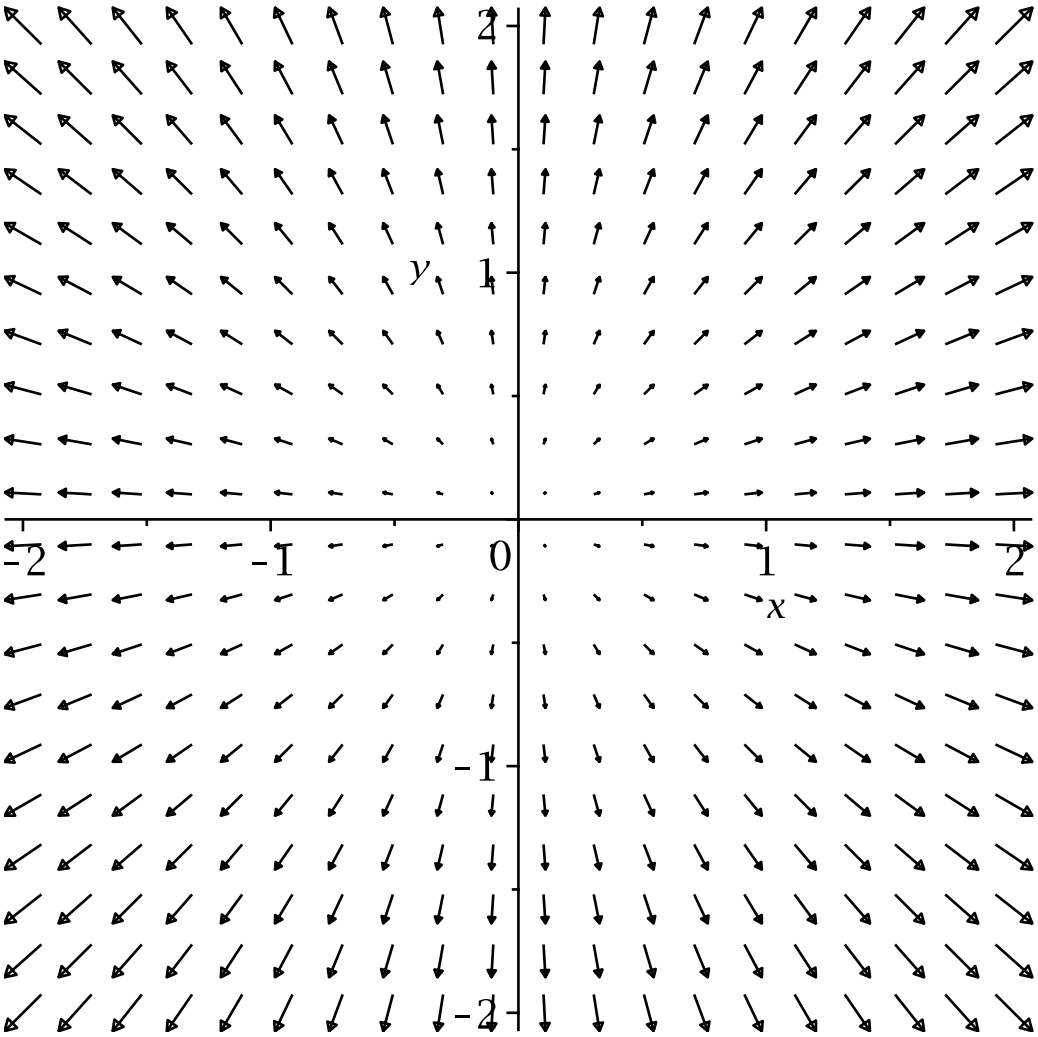
```
> display([a2, b2, c2])
```



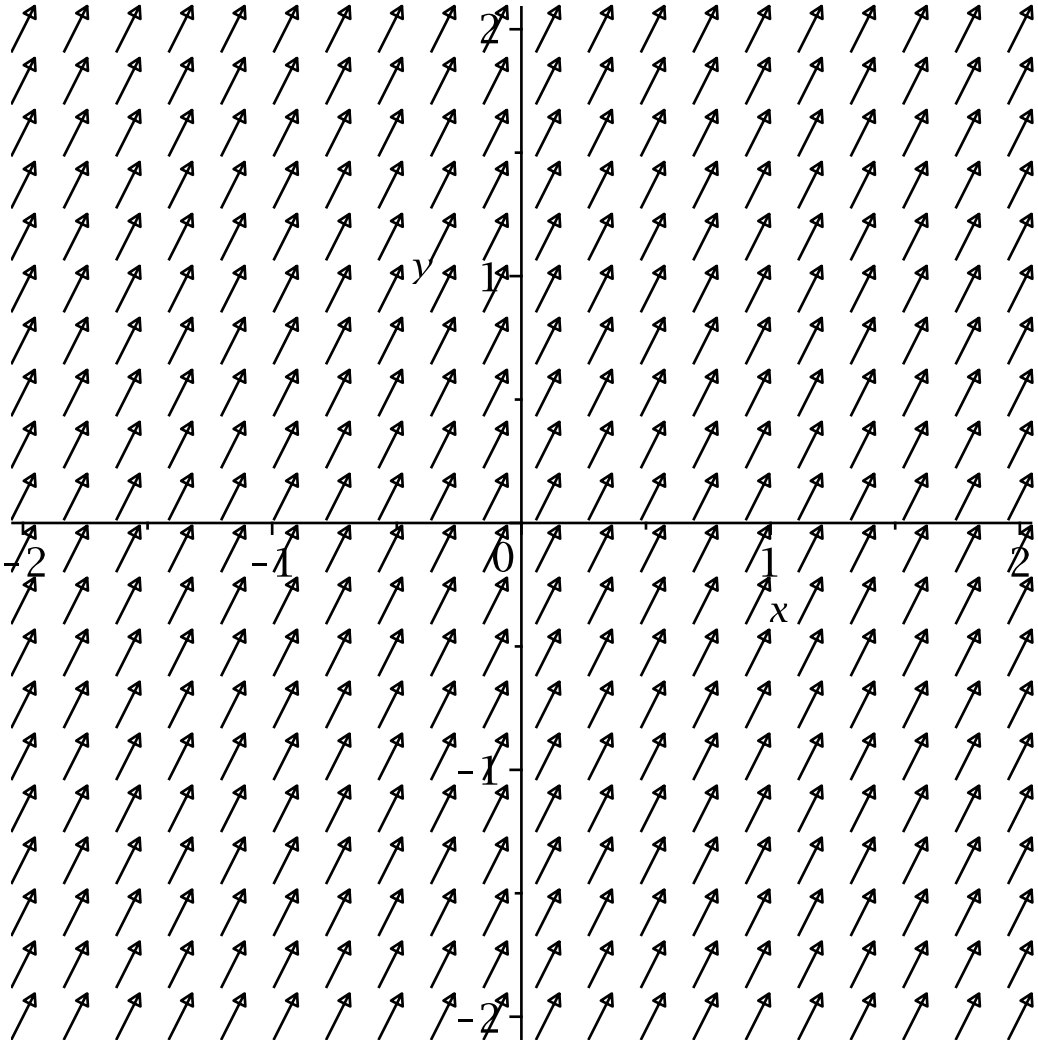
```
> restart:
> with(plots):
> ###Vektorfunktionen zweier Variabler
> ##Ebene Vektorfelder
> #Beispiel 1
> fieldplot([y, x], x=-2..2, y=-2..2,
    arrows=SLIM)
```



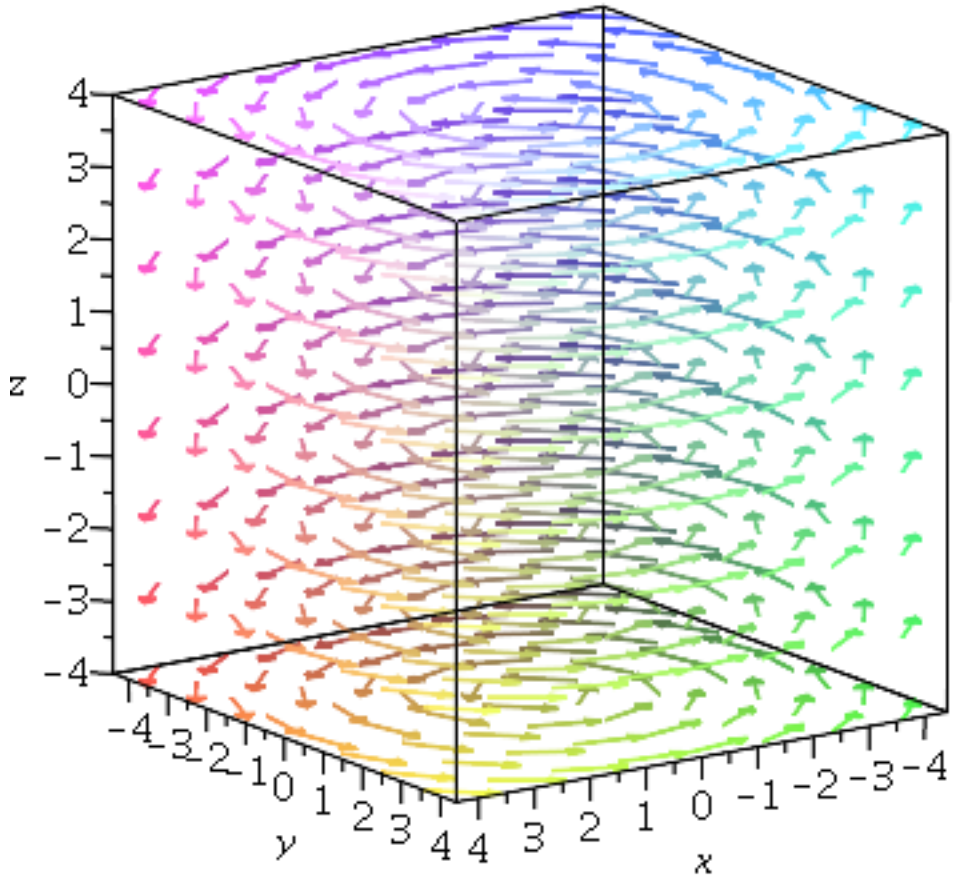

```
> #Beispiel 2
> fieldplot([x, y], x=-2..2, y=-2..2,
    arrows=SLIM)
```



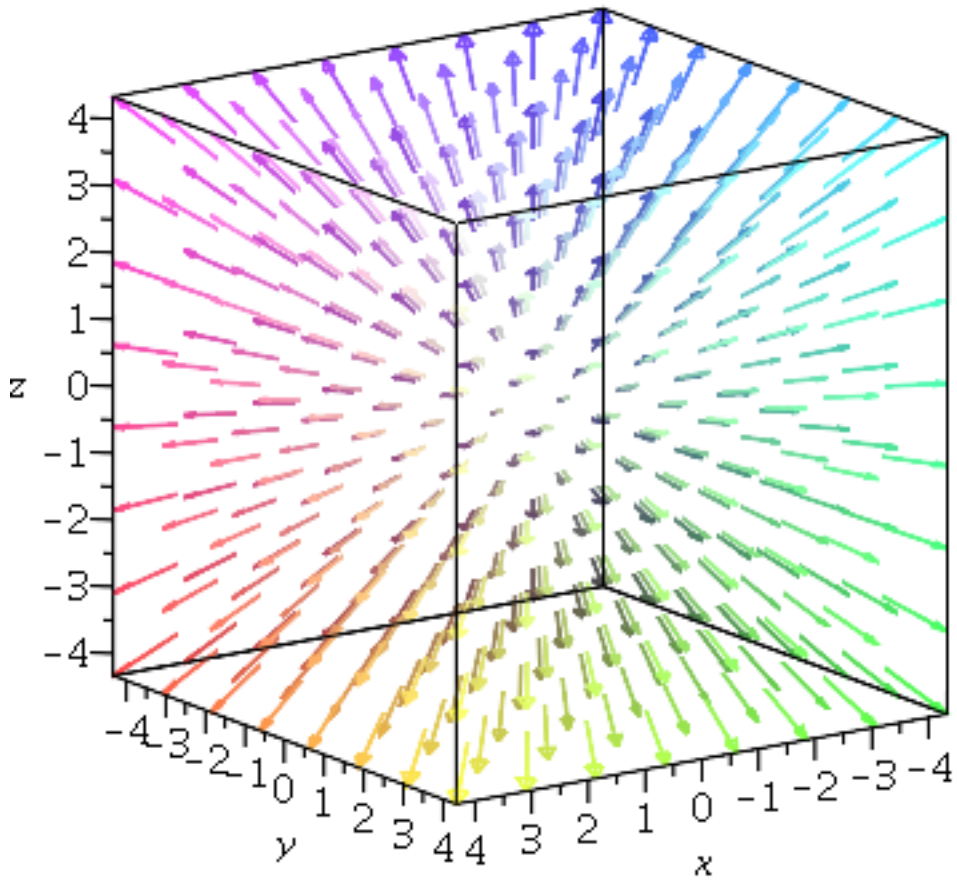

```
> #Beispiel 3
> fieldplot([2, 4], x=-2..2, y=-2..2,
    arrows=SLIM)
```



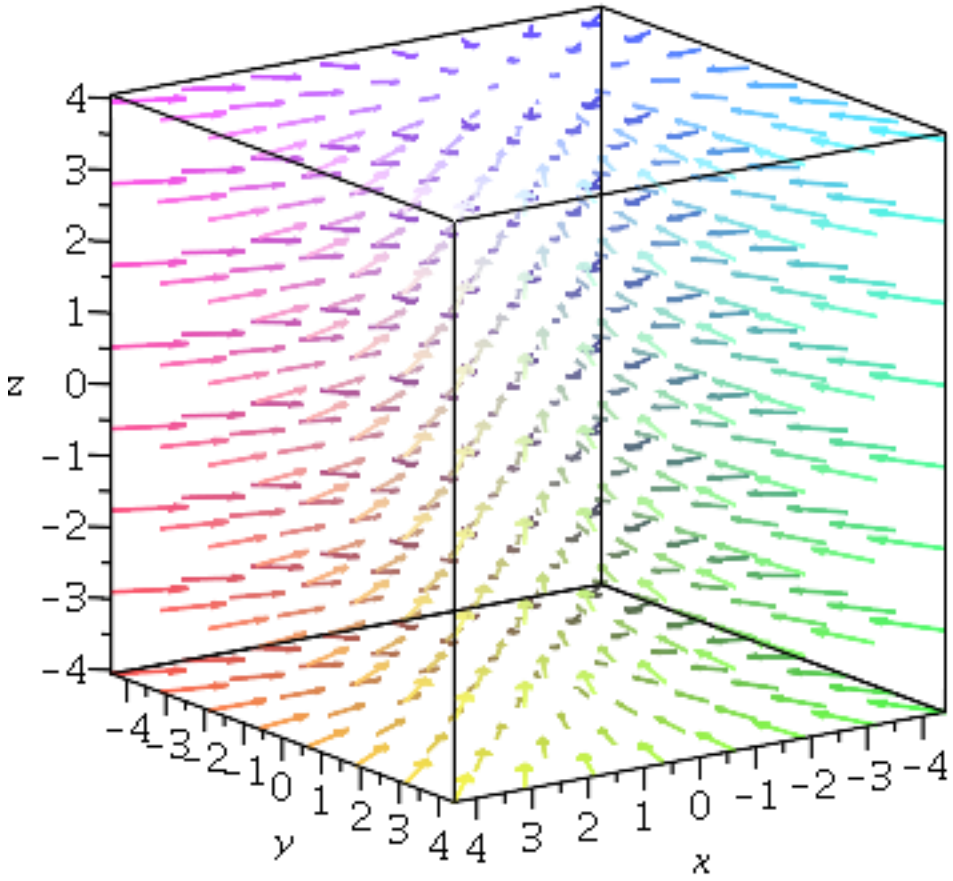

```
> ###Vektorfunktionen dreier Variabler
> ##Räumliche Vektorfelder
> #Beispiel 1
> fieldplot3d([-y/sqrt(x^2+y^2), x
  /sqrt(x^2+y^2), 0], x=-4..4, y=-4
  ..4, z=-4..4, axes=boxed, arrows
  =SLIM)
```



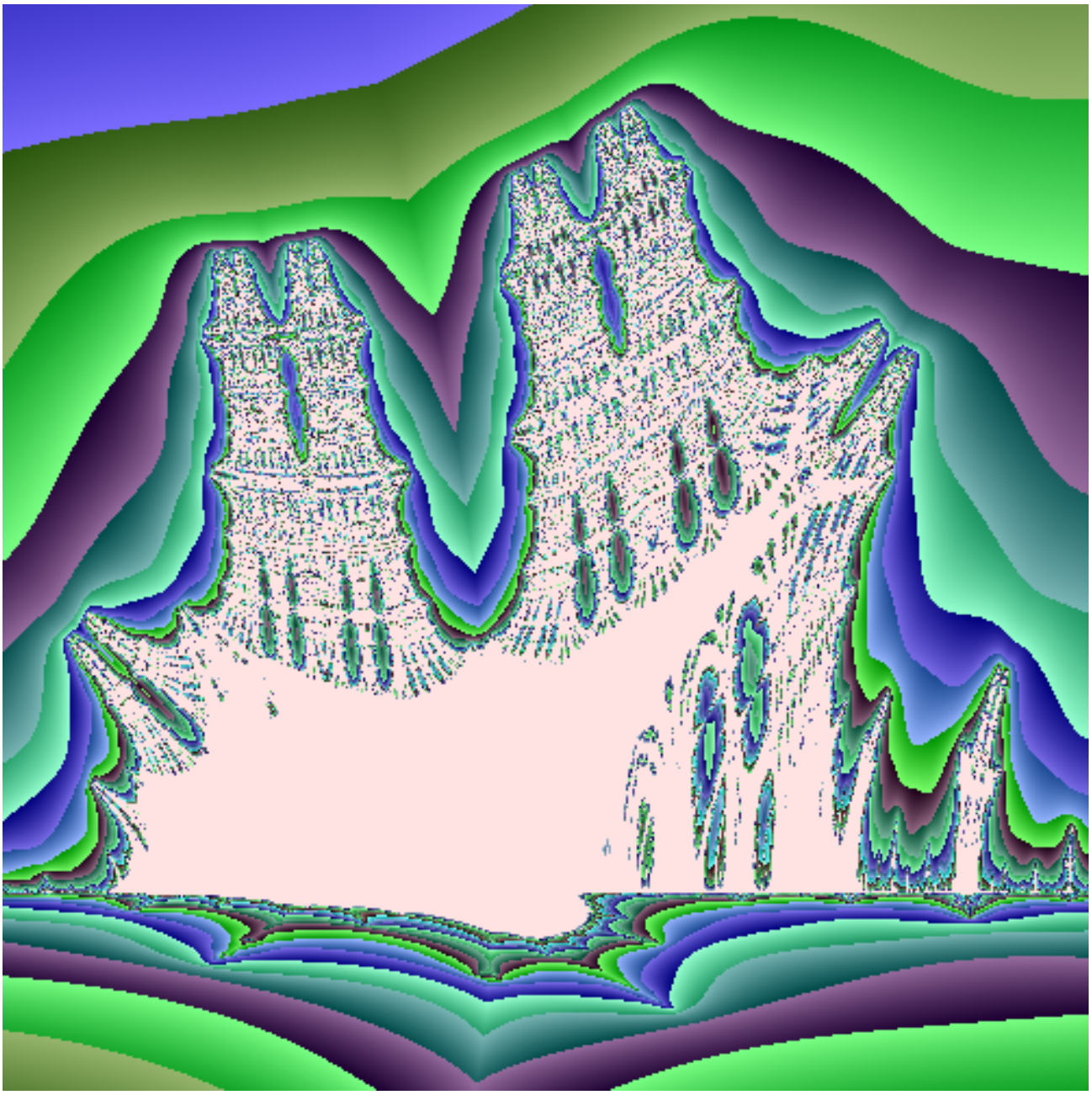

```
> #Beispiel 2
> fieldplot3d([x, y, z], x=-4..4, y=-4
    ..4, z=-4..4, axes=boxed, arrows
    =SLIM)
```



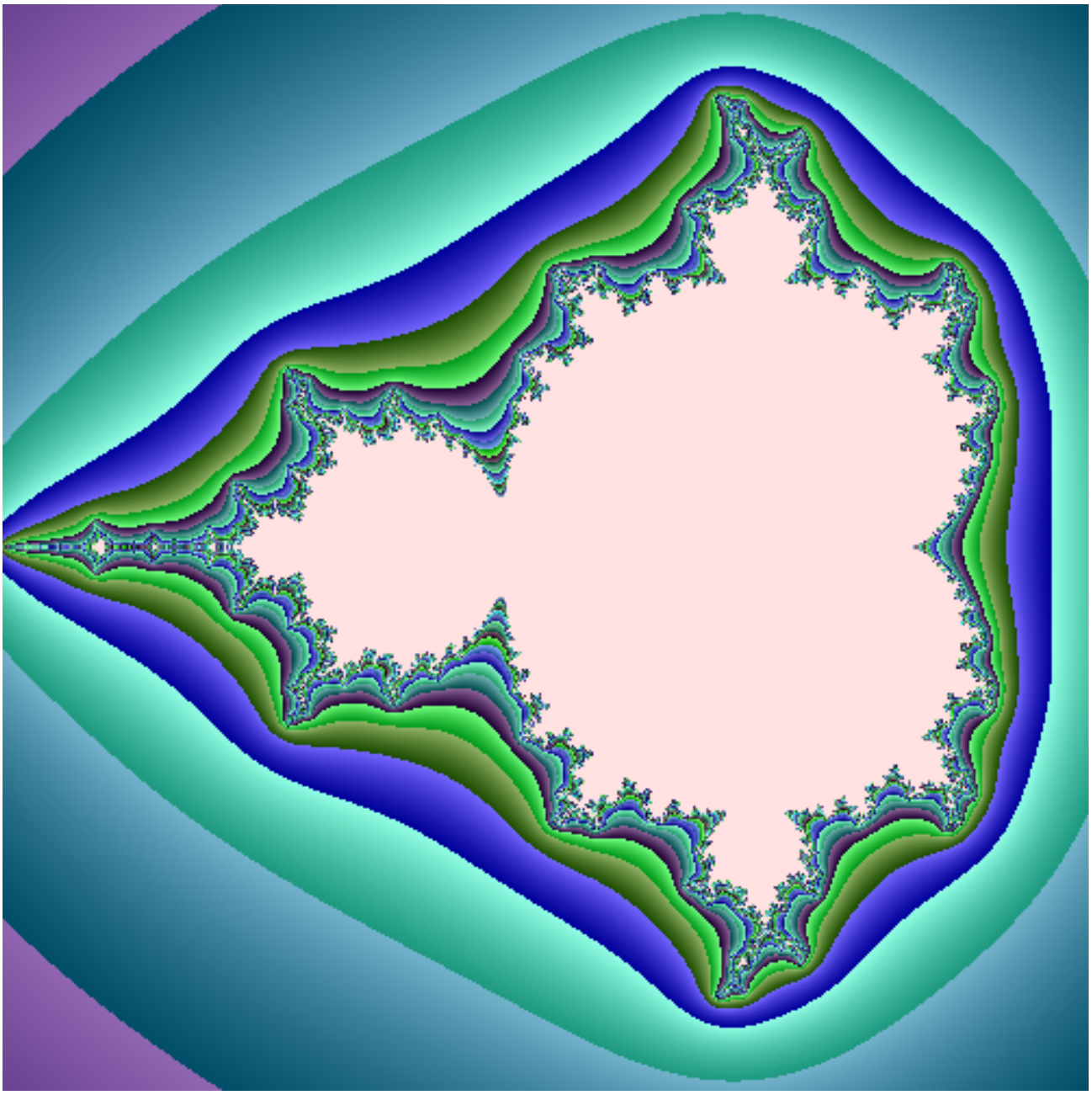

```
> #Beispiel 3
> fieldplot3d([-2 x,-2 y, 1], x=-4..4,
    y=-4..4, z=-4..4, axes=boxed,
   arrows=SLIM)
```


z
4
3
2
1
0
-1
-2
-3
-4
y
-4 -3 -2 -1 0 1 2 3 4
x
4 3 2 1 0 -1 -2 -3 -4

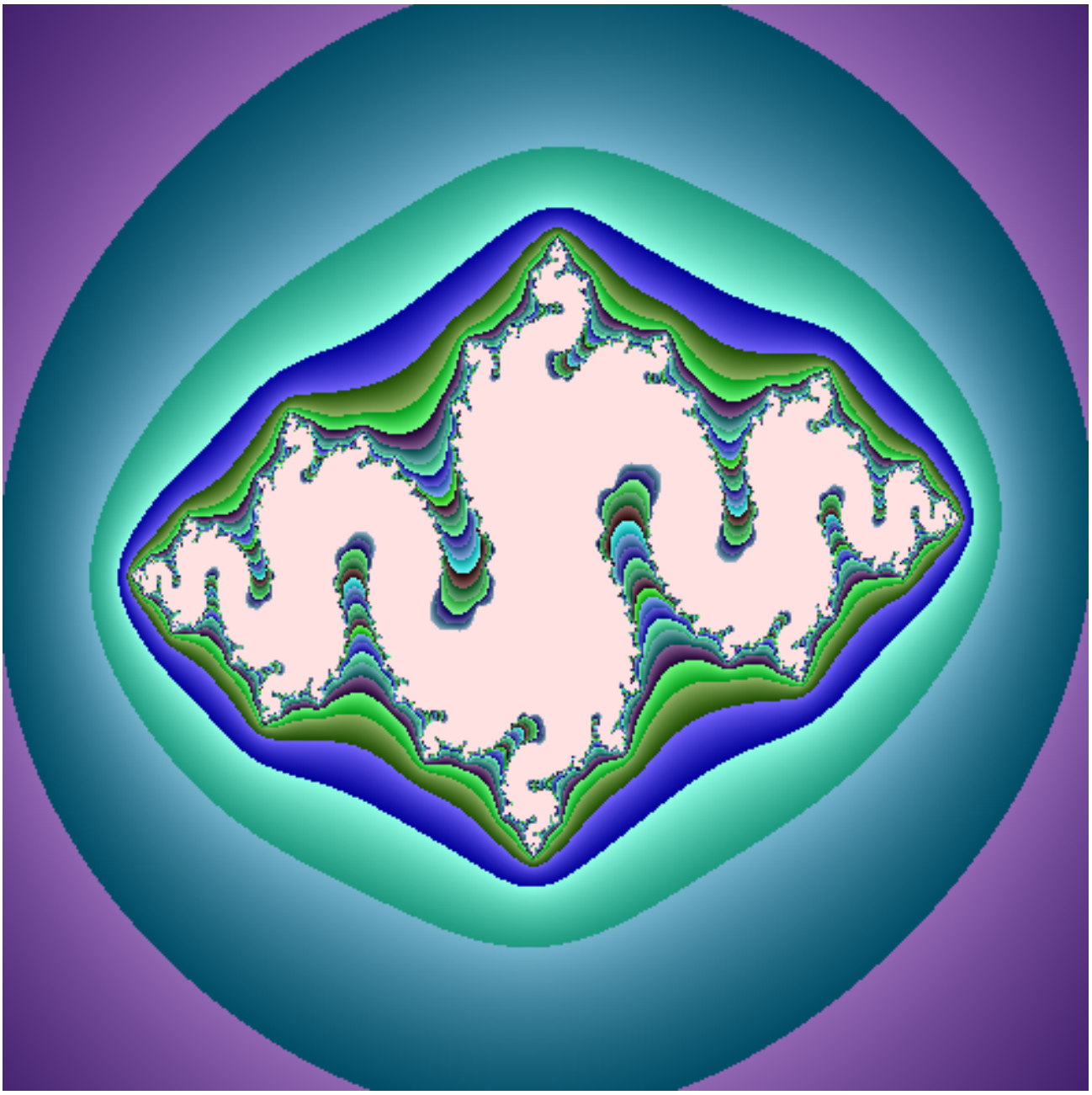
```
> restart:
> ##Fraktale
> with(Fractals:-EscapeTime):
> with(ImageTools):
> #Burning Ship-Menge
> B := BurningShip(500, -1.8 - 0.09 I,
    -1.7 + 0.02 I):
> Embed(B)
```



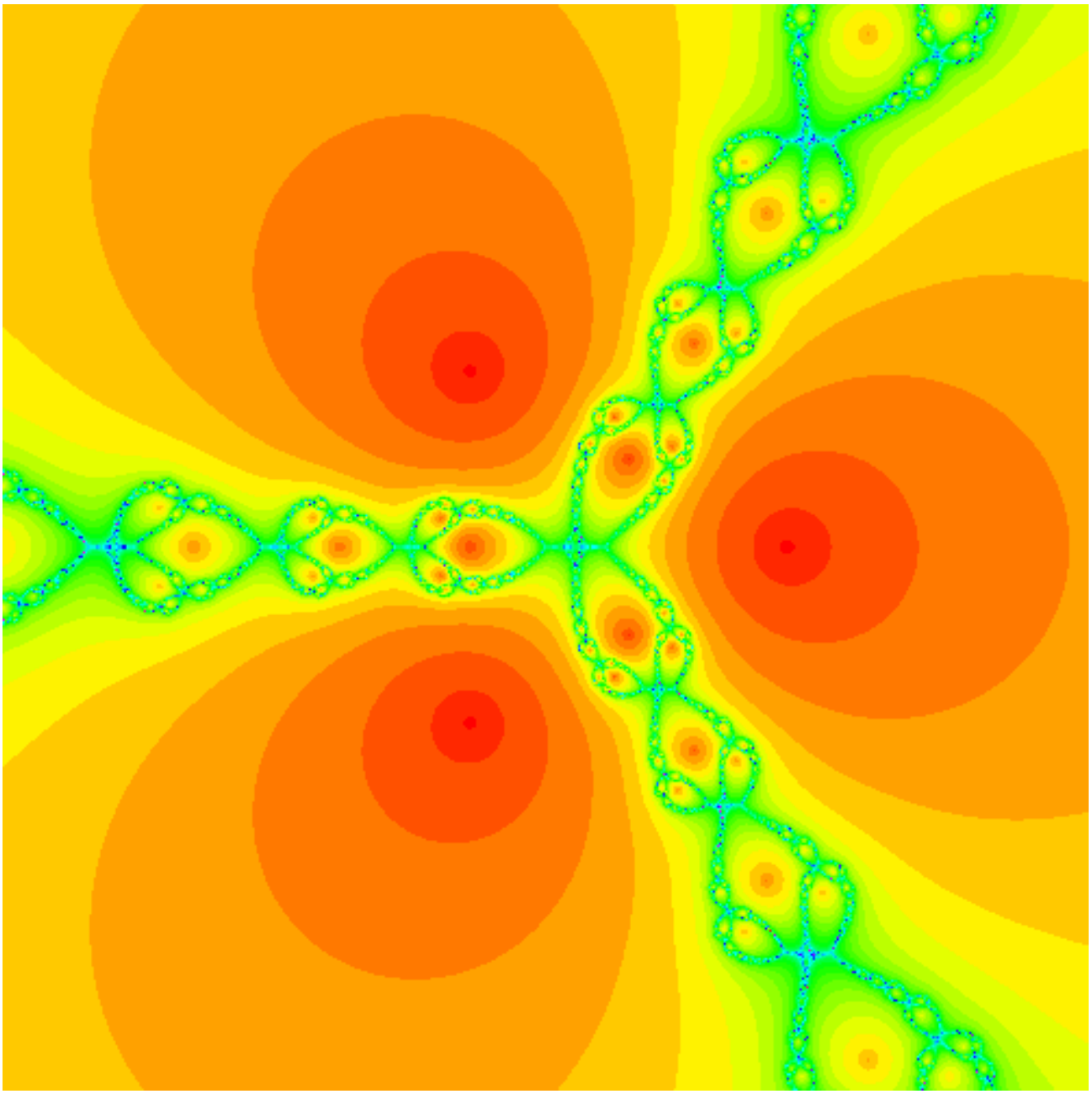
```
> #Mandelbrot-Menge
> M := Mandelbrot(500, -2.0 - 1.35 I, 0.7
    + 1.35 I):
> Embed(M)
```



```
> #Julia-Menge
> J := Julia(500, -2.0 - 1.5 I, 2.0
    + 1.5 I, -0.8 + 0.156 I):
> Embed(J)
```



```
> #Newton-Menge
> N := Newton(500, -6 - 6 I, 6 + 6 I, t^3 - t^2 - 12):
> Embed(N)
```



>