

## Darstellung von Flächen, die durch Gleichungen oder Parameterdarstellungen gegeben sind, in POV-Ray

### Flächendarstellung durch implizite Gleichungen: `isosurface`

Durch Gleichungen der Form  $f(x, y, z) = 0$  gegebene Flächen werden in POV-Ray durch folgende Anweisungen dargestellt:

<code>isosurface {</code>	
<code>function { f(x,y,z) }</code>	Funktionsterm $f(x, y, z)$ (linke Seite der impliziten Gleichung $f(x, y, z) = 0$ der Fläche)
<code>contained_by { box{a,b} }</code>	dargestellter Ausschnitt der Fläche ( $x$ , $y$ und $z$ jeweils zwischen $a$ und $b$ ); es können auch ein beliebiger Quader <code>box{&lt;a<sub>x</sub>,a<sub>y</sub>,a<sub>z</sub>&gt;,&lt;b<sub>x</sub>,b<sub>y</sub>,b<sub>z</sub>&gt;}</code> oder eine Kugel angegeben werden, um den dargestellten Ausschnitt zu begrenzen
<code>accuracy 0.001</code>	Genauigkeit der Berechnung der Fläche (kleinere Werte führen zu einer genaueren und langsameren Darstellung)
<code>max_gradient 4</code>	maximaler Anstieg (große Werte → langsame Berechnung)
<code>open</code>	die Ränder bleiben offen; ohne diesen Befehl werden die Seitenflächen der Containerbox mit dargestellt
<code>}</code>	

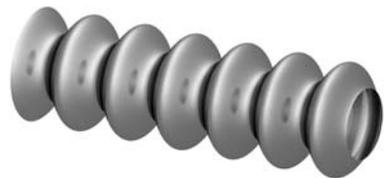
Der Wert `max_gradient` beschreibt den maximalen Anstieg der Funktion  $f(x, y, z)$ ; dieser errechnet sich aus den Ableitungen der Funktion nach  $x$ ,  $y$  und  $z$ . Setzt man `max_gradient` zu klein, werden nicht alle Bereiche der Fläche korrekt dargestellt; große Werte führen zu langen Renderzeiten. Um optimale Werte für `max_gradient` und `accuracy` zu finden ist etwas Experimentieren notwendig. Die angegebenen Werte haben sich (nach vielen Versuchen) als sinnvolle Ausgangswerte für viele Anwendungen erwiesen.

**Beispiel 1:** Die Fläche mit der impliziten Gleichung

$$\frac{\sin(10x)}{8} + \sqrt{z^2 + y^2} - 0,5 = 0$$

kann durch die Eingabe folgender Zeilen in POV-Ray dargestellt werden:<sup>1</sup>

```
isosurface {
  function { sin(10*x)/8 + sqrt(z*z+y*y) - 0.5 }
  contained_by { box { -2, 2 } }
  accuracy 0.005   max_gradient 10
  open   texture{ silbergrau }
}
```



**Beispiel 2:** Durch die Gleichung

$$-3x^2 + 2y^2 - 3z^2 - 0,25 = 0$$

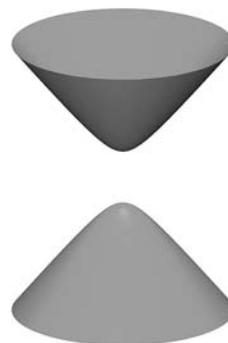
wird ein zweischaliges Hyperboloid<sup>2</sup> beschrieben.

<sup>1</sup>Alle in dieser Anleitung gegebenen Beispiele können durch Einsetzen der Anweisungen in die Vorlagdatei `vorlage.pov` direkt gerendert werden.

<sup>2</sup>Hyperboloide sind Flächen 2. Ordnung (Quadriken). Für diese Flächen gibt es in POV-Ray auch den speziellen Befehl `quadric` (siehe POV-Ray-Hilfe), der meist schneller zu denselben Ergebnissen führt, aber nicht so universell ist wie `isosurface`.

In POV-Ray entsteht es durch die folgende Eingabe:

```
isosurface {
  function { - 3*x*x + 2*y*y - 3*z*z - 0.25 }
  contained_by { box { -1.8, 1.8 } }
  accuracy 0.01
  max_gradient 20
  open
  texture { rot_matt }
  no_shadow
}
```



## Flächendarstellung durch Parametergleichungen: parametric

Durch Parametergleichungen gegebene Flächen stellt POV-Ray durch folgende Eingabe dar:

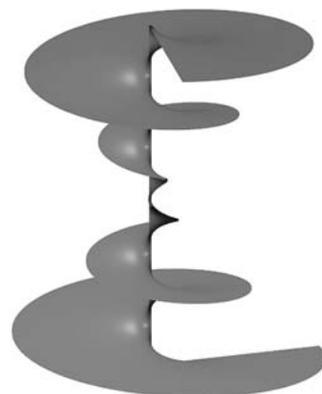
<code>parametric {</code>	
<code>function { x(u,v) }</code>	Funktionen, durch welche die Fläche gegeben ist:
<code>function { y(u,v) }</code>	Koordinaten $x$ , $y$ und $z$ in Abhängigkeit von den
<code>function { z(u,v) }</code>	Parametern $u$ und $v$
<code>&lt;u<sub>1</sub>,v<sub>1</sub>&gt;, &lt;u<sub>2</sub>,v<sub>2</sub>&gt;</code>	Definitionsbereiche für die Parameter $u$ und $v$
<code>contained_by {box{a,b}}</code>	Begrenzungsquader (siehe <code>isosurface</code> )
<code>accuracy 0.001 max_gradient 4</code>	Genauigkeit der Berechnung, maximaler Anstieg
<code>precompute 18 x,y,z</code>	Vorausberechnung von Intervallen (siehe unten)
<code>}</code>	

**Beispiel:** Die konische Spirale mit der Parameterdarstellung

$$\begin{aligned} x &= u \cdot v \cdot \sin(15v) \\ y &= v && \text{(mit } u \in [0; 1], v \in [-1; 1] \text{)} \\ z &= u \cdot v \cdot \cos(15v) \end{aligned}$$

lässt sich durch die folgende Eingabe darstellen:

```
parametric {
  function {u*v*sin(15*v)}
  function {v}
  function {u*v*cos(15*v)}
  <0,-1>,<1,1>
  contained_by { box { -1, 1 } }
  max_gradient 10 accuracy 0.0005
  precompute 18 x,y,z
  texture{blau_matt} no_shadow
}
```



Die Anweisung `precompute 18 x,y,z` bewirkt eine Unterteilung der Fläche in  $2^{18}$  Teilflächen. Für jede dieser Teilflächen werden die Intervalle vorausberechnet, in denen die Koordinaten  $x$ ,  $y$  und  $z$  liegen. Dies erfordert eine gewisse Vorausrechnungszeit (parsing) vor dem Rendervorgang, verkürzt diesen dann jedoch erheblich.

Geeignete Werte für `accuracy` und `max_gradient` müssen wie bei dem Befehl `isosurface` durch Versuche gefunden werden. Bei zu geringen Werten für `max_gradient` werden u. U. nicht alle Bereiche der Fläche dargestellt, große Werte verlängern erheblich den Rendervorgang. Flächen werden korrekt berechnet, wenn der angegebene Wert für `max_gradient` mindestens so groß ist wie das Maximum aller 6 partiellen Ableitungen der Koordinatenfunktionen nach den Parametern  $u$  und  $v$ , also  $\frac{\partial x}{\partial u}$ ,  $\frac{\partial x}{\partial v}$ ,  $\frac{\partial y}{\partial u}$ ,  $\frac{\partial y}{\partial v}$ ,  $\frac{\partial z}{\partial u}$  und  $\frac{\partial z}{\partial v}$ , über dem gesamten Definitionsbereich von  $u$  und  $v$ .