

Nama : Nika Widyaningrum
NIM : 23030130003
Kelas : Pendidikan Matematika A 2023

Menggambar Plot 3D dengan EMT

Ini adalah pengenalan plot 3D di Euler. Kita membutuhkan plot 3D untuk memvisualisasikan fungsi dari dua variabel.

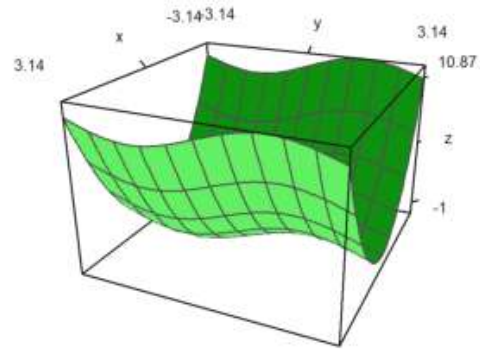
Euler menggambar fungsi tersebut menggunakan algoritma pengurutan untuk menyembunyikan bagian di latar belakang. Secara umum, Euler menggunakan proyeksi pusat. Standarnya adalah dari kuadran x-y positif menuju titik asal $x=y=z=0$, tetapi sudut $=0^\circ$ terlihat dari arah sumbu y. Sudut pandang dan ketinggian dapat diubah.

Euler dapat merencanakan

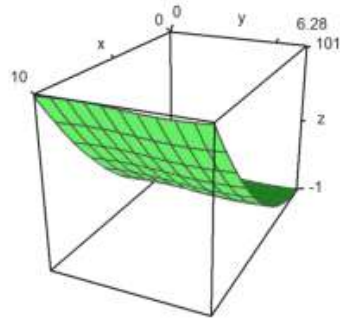
- permukaan dengan bayangan dan garis level atau rentang level,
- awan poin,
- kurva parametrik,
- permukaan implisit.

Plot 3D dari suatu fungsi menggunakan plot3d. Cara termudah adalah dengan memplot ekspresi dalam x dan y. Parameter r mengatur kisaran plot di sekitar (0,0).

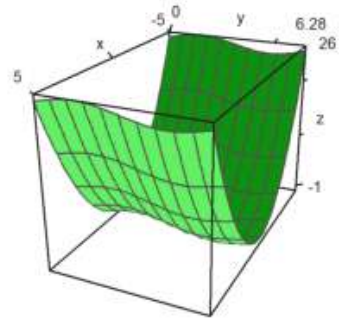
```
> aspect(1.5); plot3d("x^2+sin(y)",r=pi):
```



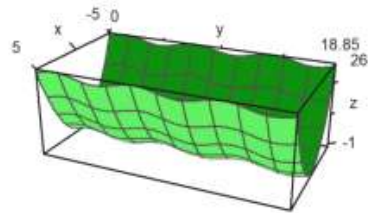
```
>aspect(1.5); plot3d("x^2+sin(y)",0,10,0,2*pi):
```



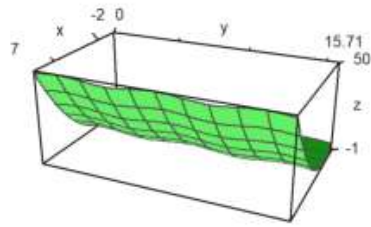
```
>aspect(1.5); plot3d("x^2+sin(y)",-5,5,0,2*pi):
```



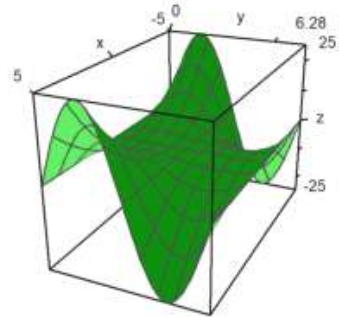
```
>aspect(1.5); plot3d("x^2+sin(y)",-5,5,0,6*pi):
```



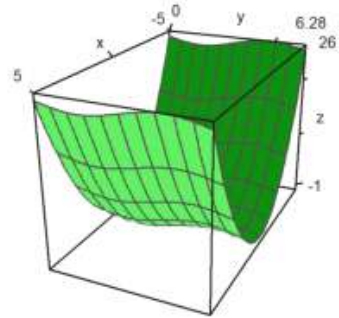
```
>aspect(1.5); plot3d("x^2+sin(y)",-2,7,0,5*pi):
```



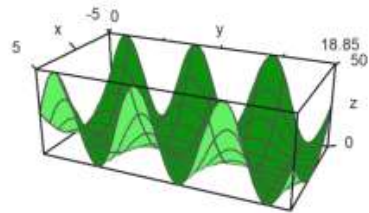
```
>aspect(1.5); plot3d("x^2*sin(y)",-5,5,0,2*pi):
```



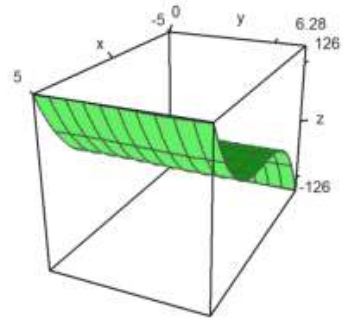
```
>aspect(1.5); plot3d("x^2-sin(y)",-5,5,0,2*pi):
```

```
>aspect(1.5); plot3d("x^2*(1+sin(y))",-5,5,0,6*pi):
```



```
>aspect(1.5); plot3d("x^3+sin(y)",-5,5,0,2*pi):
```



Fungsi dua Variabel

Untuk grafik fungsi, gunakan

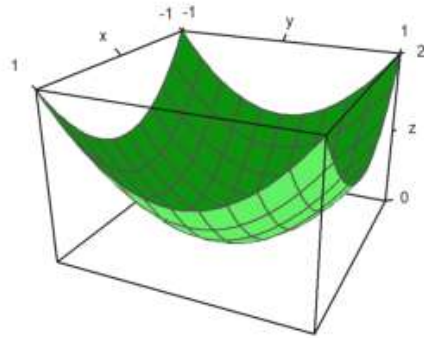
- ekspresi sederhana dalam x dan y ,
- nama fungsi dari dua variabel
- atau matriks data.

Standarnya adalah kisi kawat yang diisi dengan warna berbeda di kedua sisi. Perhatikan bahwa jumlah default interval grid adalah 10, tetapi plot menggunakan jumlah default 40×40 persegi panjang untuk membangun permukaan. Ini bisa diubah.

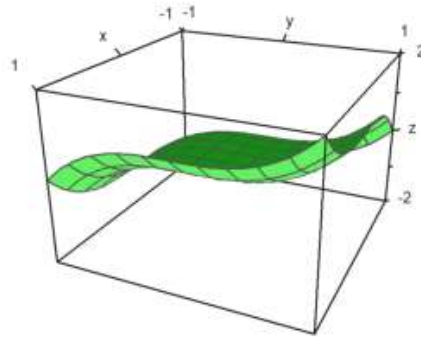
- $n=40$, $n=[40,40]$: jumlah garis kisi di setiap arah
- $grid=10$, $grid=[10,10]$: jumlah garis grid di setiap arah.

Kami menggunakan default $n=40$ dan $grid=10$.

```
>plot3d("x^2+y^2"):
```



```
>plot3d("x^3+y^3"):
```

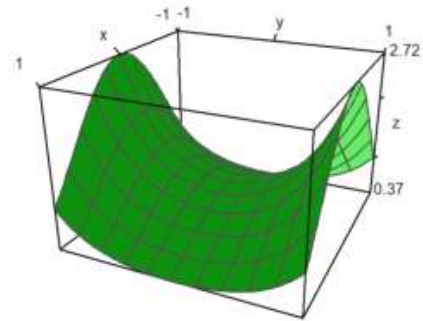


Interaksi pengguna dimungkinkan dengan >parameter pengguna. Pengguna dapat menekan tombol berikut.

- kiri, kanan, atas, bawah: putar sudut pandang
- +,-: memperbesar atau memperkecil
- a: menghasilkan anaglyph (lihat di bawah)
- l: beralih memutar sumber cahaya (lihat di bawah)
- spasi: reset ke default
- kembali: akhiri interaksi

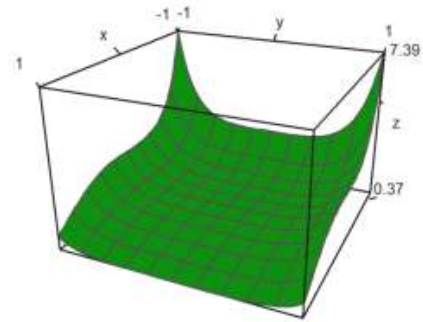
```
>plot3d("exp(-x^2+y^2)",>user, ...  
> title="Turn with the vector keys (press return to finish)":
```

Turn with the vector keys (press return to finish)



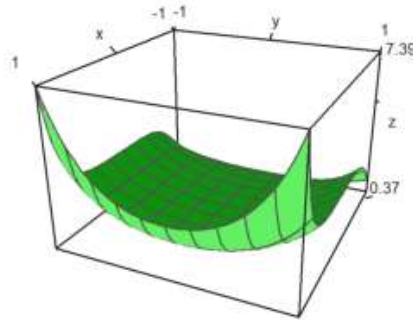
```
>plot3d("exp(-x^3+y^4)",>user, ...  
> title="Turn with the vector keys (press return to finish)":
```

Turn with the vector keys (press return to finish)



```
>plot3d("exp(x^5+y^2)",>user, ...  
> title="Turn with the vector keys (press return to finish)":
```


Turn with the vector keys (press return to finish)



Rentang plot untuk fungsi dapat ditentukan dengan

- a,b: rentang-x
- c,d: rentang-y
- r: persegi simetris di sekitar (0,0).
- n: jumlah subinterval untuk plot.

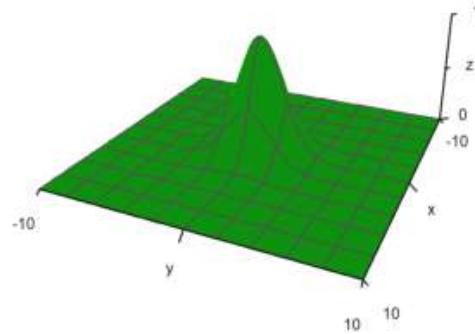
Ada beberapa parameter untuk menskalakan fungsi atau mengubah tampilan grafik.

fscale: skala ke nilai fungsi (defaultnya adalah <fscale).

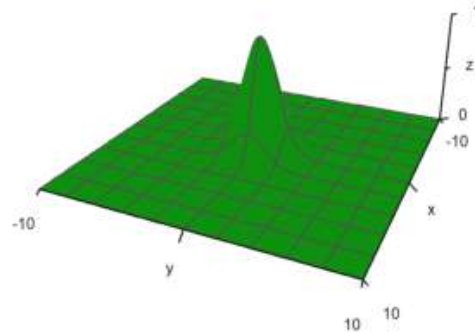
skala: angka atau vektor 1x2 untuk skala ke arah x dan y.

bingkai: jenis bingkai (default 1).

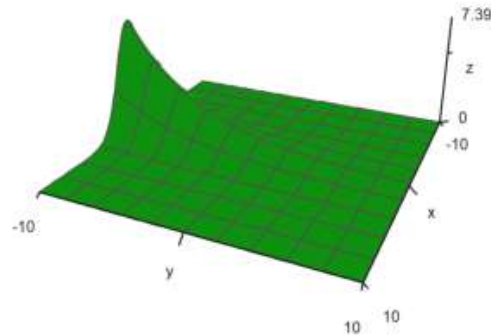
```
>plot3d("exp(-(x^2+y^2)/5)",r=10,n=80,fscale=4,scale=1.2,frame=3):
```



```
>plot3d("exp(-(x^2+y^2)/3)",r=10,n=80,fscale=4,scale=1.2,frame=3):
```



```
>plot3d("exp(-(x^2+y)/5)",r=10,n=80,fscale=4,scale=1.2,frame=3):
```



Tampilan dapat diubah dengan berbagai cara.

- jarak: jarak pandang ke plot.
- zoom: nilai zoom.
- sudut: sudut terhadap sumbu y negatif dalam radian.
- tinggi: ketinggian tampilan dalam radian.

Nilai default dapat diperiksa atau diubah dengan fungsi `view()`. Ini mengembalikan parameter dalam urutan di atas.

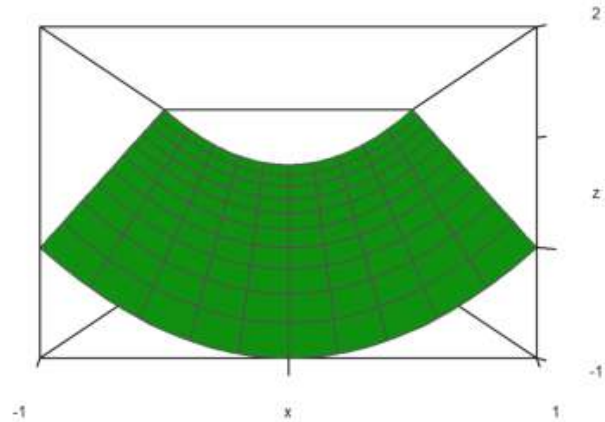
```
>view
```

```
[5, 2.6, 2, 0.4]
```

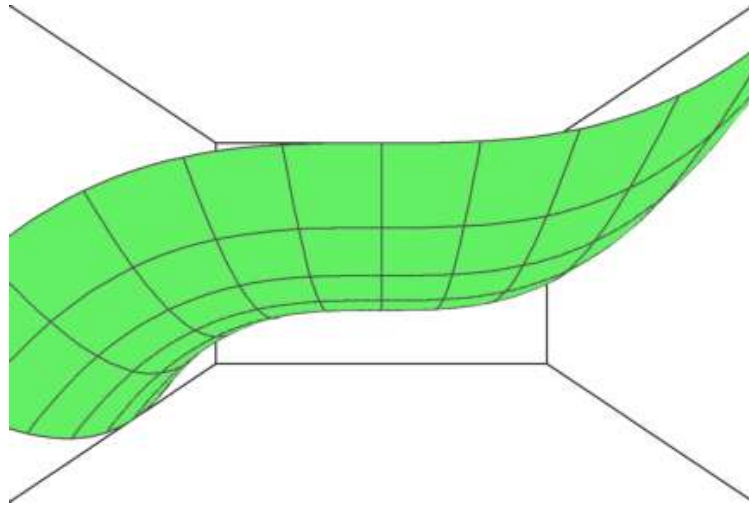
Jarak yang lebih dekat membutuhkan lebih sedikit zoom. Efeknya lebih seperti lensa sudut lebar.

Dalam contoh berikut, sudut=0 dan tinggi=0 terlihat dari sumbu y negatif. Label sumbu untuk y disembunyikan dalam kasus ini.

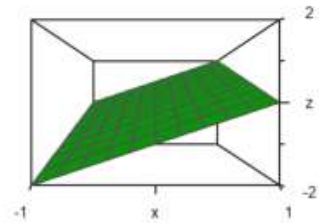
```
>plot3d("x^2+y",distance=3,zoom=2,angle=0,height=0):
```



```
>plot3d("x^3+y^2",distance=2,zoom=2,angle=0,height=0):
```

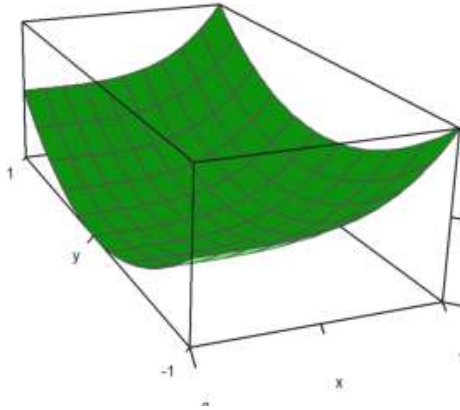


```
>plot3d("x+y",distance=3,zoom=1,angle=0,height=0):
```

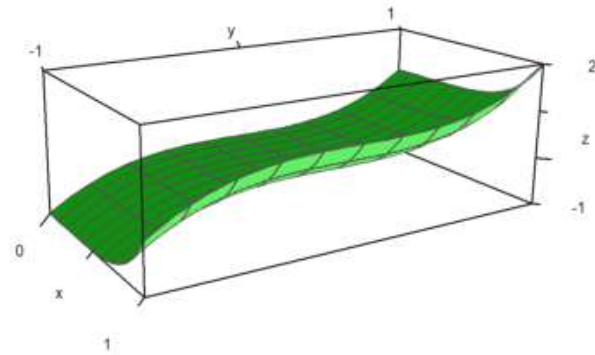


Plot terlihat selalu ke pusat kubus plot. Anda dapat memindahkan pusat dengan parameter tengah.

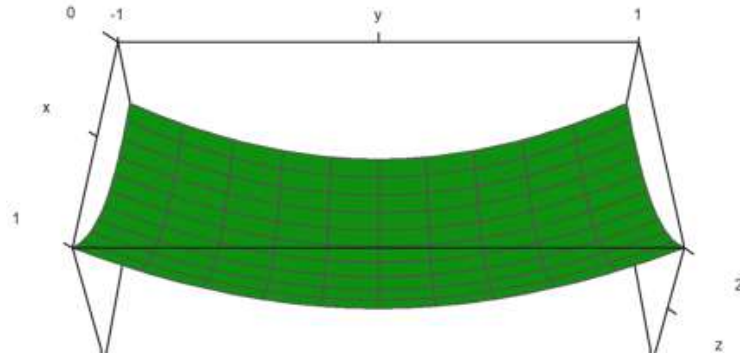
```
>plot3d("x^4+y^2",a=0,b=1,c=-1,d=1,angle=-20°,height=20°, ...  
> center=[0.4,0,0],zoom=5):
```



```
>plot3d("x^6+y^3",a=0,b=1,c=-1,d=1,angle=60°,height=15°, ...  
> center=[0.6,0,0],zoom=4):
```

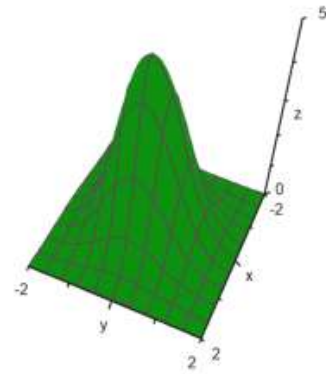
```
>plot3d("x^3+y^2",a=0,b=1,c=-1,d=1,angle=90°,height=45°, ...  
> center=[0.5,0,0],zoom=5):
```



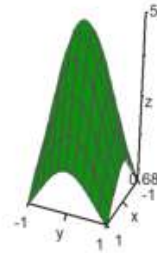
Plot diskalakan agar sesuai dengan kubus satuan untuk dilihat. Jadi tidak perlu mengubah jarak atau zoom tergantung pada ukuran plot. Namun, label mengacu pada ukuran sebenarnya.

Jika Anda mematakannya dengan `scale=false`, Anda harus berhati-hati, agar plot tetap pas dengan jendela plot, dengan mengubah jarak pandang atau zoom, dan memindahkan bagian tengahnya.

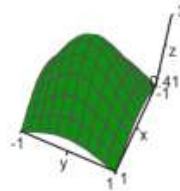
```
>plot3d("5*exp(-x^2-y^2)",r=2,<fscale,<scale,distance=13,height=50°, ...
> center=[0,0,-2],frame=3):
```



```
>plot3d("5*exp(-x^2-y^2)",r=1,<fscale,<scale,distance=15,height=30°, ...  
> center=[0,0,-2],frame=3):
```

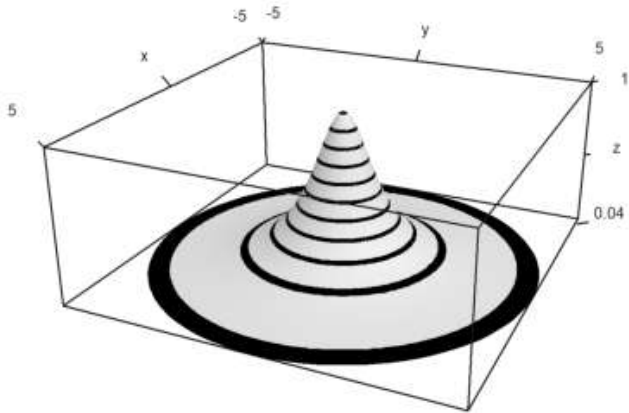


```
>plot3d("3*exp(-x^2-y^2)",r=1,<fscale,<scale,distance=15,height=70°, ...  
> center=[0,0,2],frame=3):
```

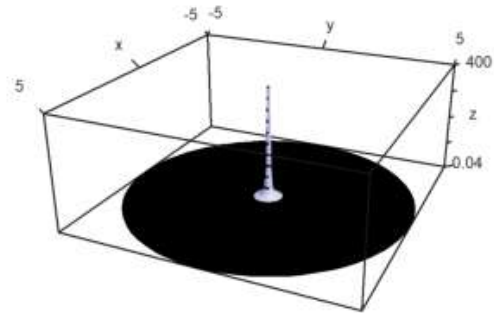


Sebuah plot kutub juga tersedia. Parameter `polar=true` menggambar plot polar. Fungsi tersebut harus tetap merupakan fungsi dari x dan y . Parameter `fscale` menskalakan fungsi dengan skala sendiri. Jika tidak, fungsi diskalakan agar sesuai dengan kubus.

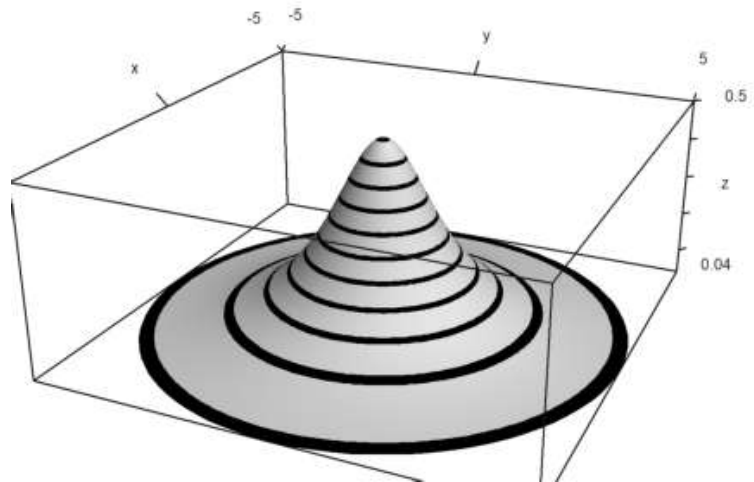
```
>plot3d("1/(x^2+y^2+1)",r=5,>polar, ...  
>fscale=2,>hue,n=100,zoom=4,>contour,color=gray):
```



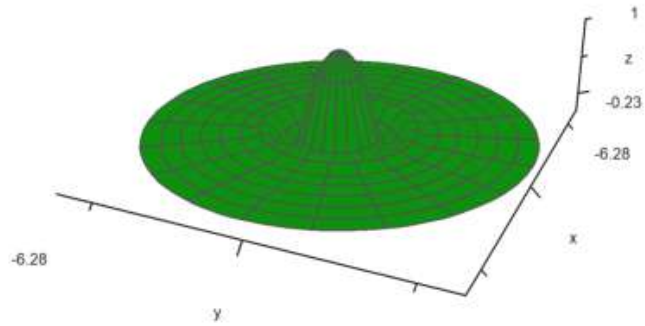
```
>plot3d("1/(x^2+y^2)",r=5,>polar, ...  
>fscale=2,>hue,n=100,zoom=3,>contour,color=blue):
```



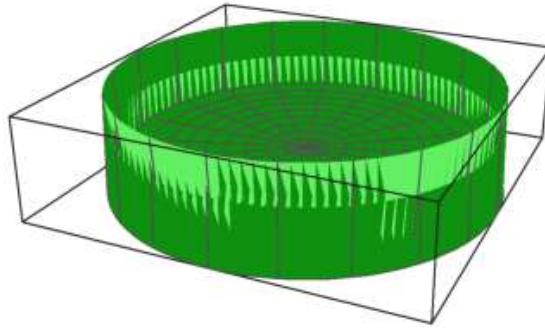
```
>plot3d("1/(x^2+y^2+2)",r=5,>polar, ...  
>fscale=2,>hue,n=100,zoom=5,>contour,color=black):
```



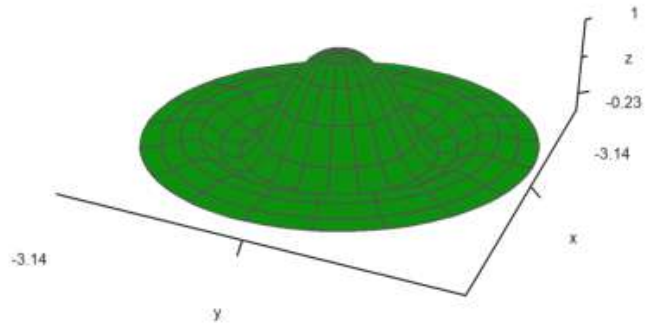
```
>function f(r) := exp(-r/2)*cos(r); ...  
>plot3d("f(x^2+y^2)",>polar,scale=[1,1,0.4],r=2pi,frame=3,zoom=4):
```

```
>function f(r) := exp(r/2)*cos(r); ...  
>plot3d("f(x^2+y^2)",>polar,scale=[1,1,0.4],r=2pi,frame=2,zoom=4):
```



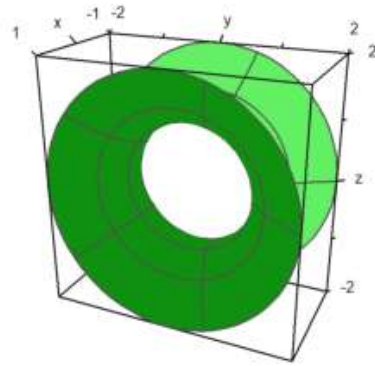
```
>function f(r) := exp(-r/2)*cos(r); ...  
>plot3d("f(x^2+y^2)",>polar,scale=[1,1,0.4],r=pi,frame=3,zoom=4):
```



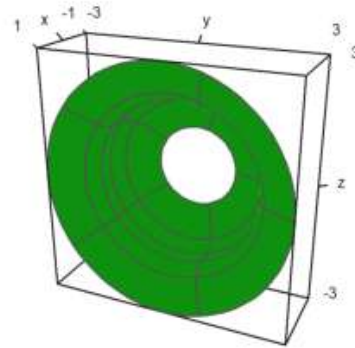
Rotasi parameter memutar fungsi dalam x di sekitar sumbu x.

- rotate=1: Menggunakan sumbu x
- rotate=2: Menggunakan sumbu z

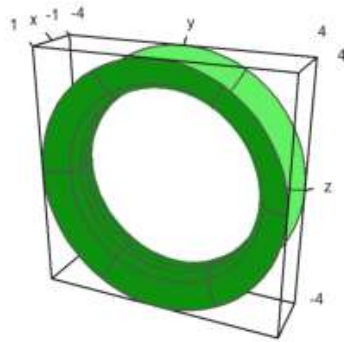
```
>plot3d("x^2+1",a=-1,b=1,rotate=true,grid=5):
```



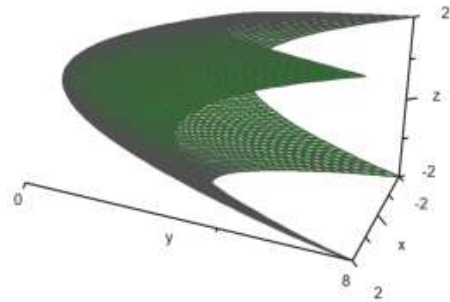
```
>plot3d("x^3+2",a=-1,b=1,rotate=true,grid=5):
```



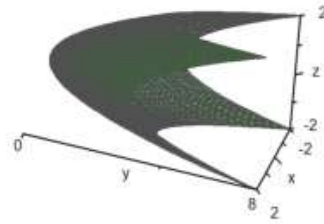
```
>plot3d("x^4+3",a=-1,b=1,rotate=true,grid=6):
```



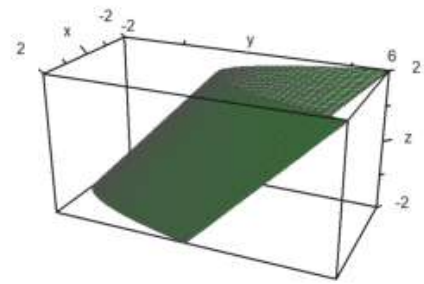
```
>plot3d("x", "x^2+y^2", "y", r=2, zoom=3.5, frame=3):
```



```
>plot3d("x", "x^2+y^2", "y", r=2, zoom=2.5, frame=3):
```



```
>plot3d("x", "x^2+y", "y", r=2, zoom=3, frame=-3):
```

Plot Kontur

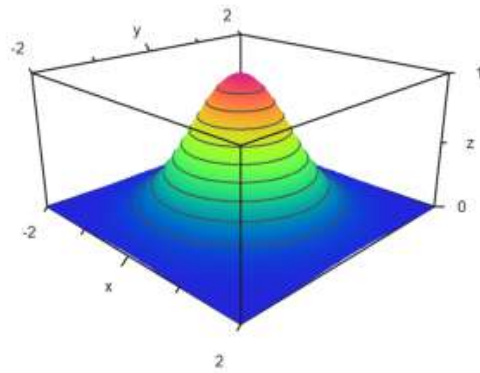
Untuk plot, Euler menambahkan garis grid. Sebagai gantinya dimungkinkan untuk menggunakan garis level dan rona satu warna atau rona berwarna spektral. Euler dapat menggambar tinggi fungsi pada plot dengan bayangan. Di semua plot 3D, Euler dapat menghasilkan anaglyph merah/sian.

- > hue: Menyalakan bayangan cahaya alih-alih kabel.
- > kontur: Memplot garis kontur otomatis pada plot.
- level=... (atau level): Sebuah vektor nilai untuk garis kontur.

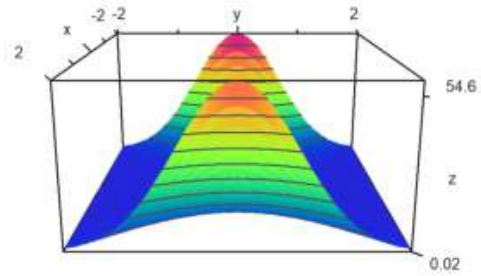
Standarnya adalah level="auto", yang menghitung beberapa garis level secara otomatis. Seperti yang Anda lihat di plot, level sebenarnya adalah rentang level.

Gaya default dapat diubah. Untuk plot kontur berikut, kami menggunakan grid yang lebih halus untuk 100x100 poin, skala fungsi dan plot, dan menggunakan sudut pandang yang berbeda.

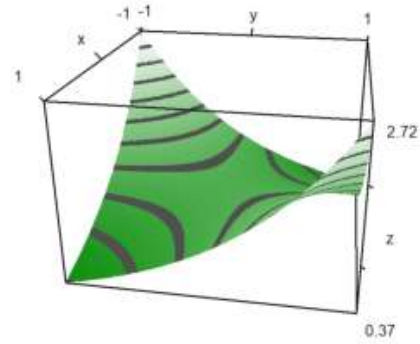
```
>plot3d("exp(-x^2-y^2)",r=2,n=100,level="thin", ...  
> >contour,>spectral,fscale=1,scale=1.1,angle=45°,height=20°):
```



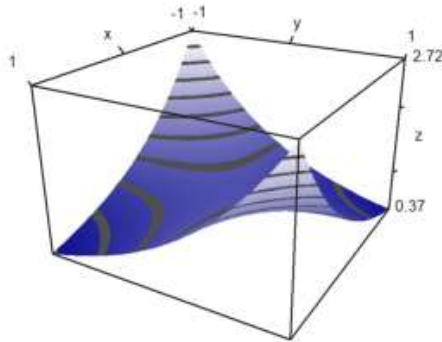
```
>plot3d("exp(x^2-y^2)",r=2,n=100,level="thin", ...  
> >contour,>spectral,fscale=1,scale=1.1,angle=90°,height=15°):
```



```
>plot3d("exp(x*y)",angle=100°,>contour,color=green):
```



```
>plot3d("exp(x*y)",angle=120°,>contour,color=blue):
```



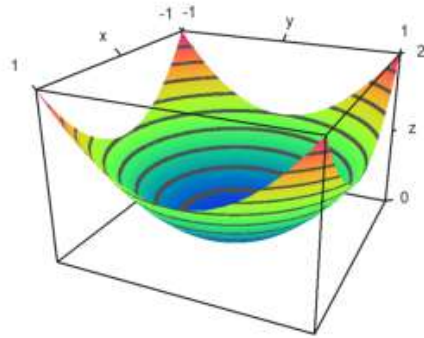
Bayangan default menggunakan warna abu-abu. Tetapi rentang warna spektral juga tersedia.

-> spektral: Menggunakan skema spektral default

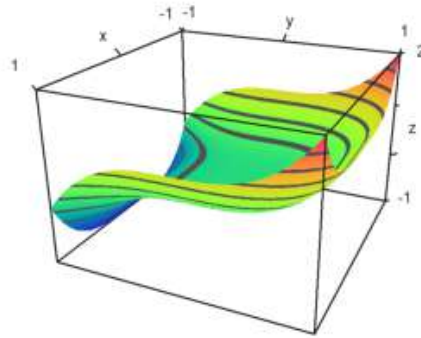
- color=...: Menggunakan warna khusus atau skema spektral

Untuk plot berikut, kami menggunakan skema spektral default dan menambah jumlah titik untuk mendapatkan tampilan yang sangat halus.

```
>plot3d("x^2+y^2",>spektral,>contour,n=100):
```

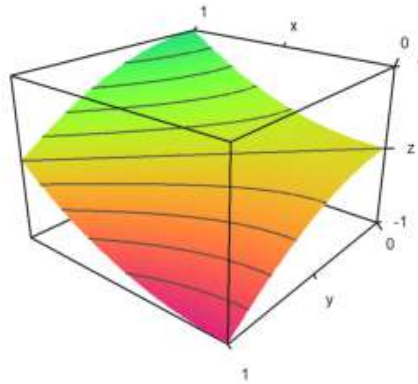


```
>plot3d("x^4+y^3",>spectral,>contour,n=100):
```

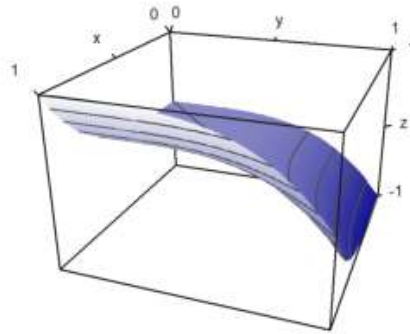


Alih-alih garis level otomatis, kita juga dapat mengatur nilai garis level. Ini akan menghasilkan garis level tipis alih-alih rentang level.

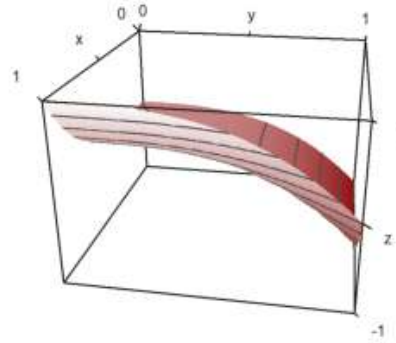
```
>plot3d("x^2-y^2",0,1,0,1,angle=220°,level=-1:0.2:1,color=redgreen):
```

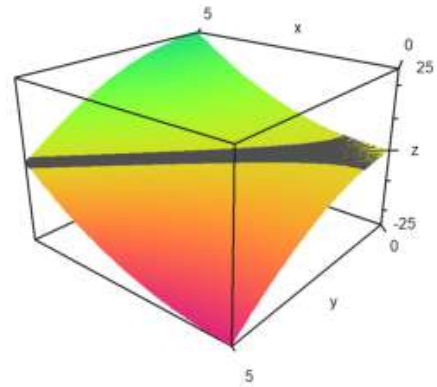
```
>plot3d("x^4-y^4",0,1,0,1,angle=110°,level=-1:0.2:1,color=blue):
```



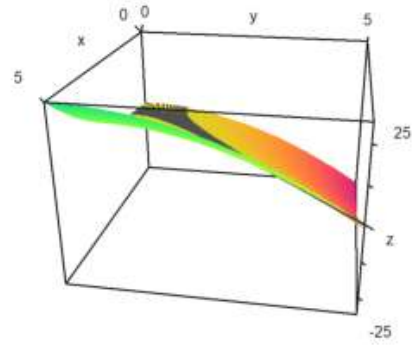
```
>plot3d("x^4-y^3",0,1,0,1,angle=100°,level=-1:0.2:1,color=red):
```



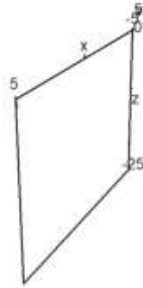
```
>plot3d("x^2-y^2",0,5,0,5,angle=220°,level=-1:0.1:1,color=redgreen):
```



```
>plot3d("x^2-y^2",0,5,0,5,angle=100°,level=-1:0.1:1,color=redgreen):
```



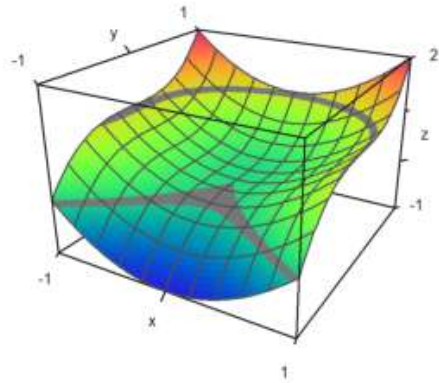
```
>plot3d("x^2-y^2",-5,5,5,5,level=-1:0.2:1,color=redgreen):
```



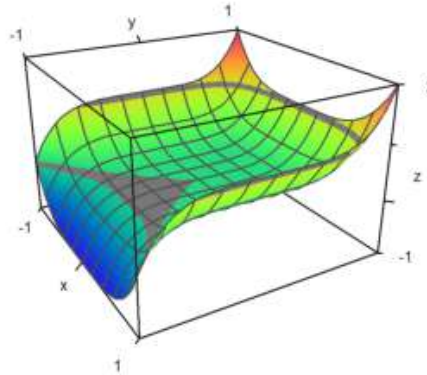
Dalam plot berikut, kami menggunakan dua pita level yang sangat luas dari -0,1 hingga 1, dan dari 0,9 hingga 1. Ini dimasukkan sebagai matriks dengan batas level sebagai kolom.

Selain itu, kami melapisi kisi dengan 10 interval di setiap arah.

```
>plot3d("x^2+y^3",level=[-0.1,0.9;0,1], ...  
> >spectral,angle=30°,grid=10,contourcolor=gray):
```



```
>plot3d("x^4+y^5",level=[-0.1,0.9;0,1], ...  
> >spectral,angle=60°,grid=10,contourcolor=gray):
```

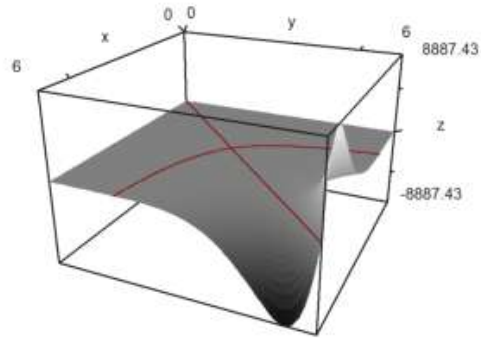


Dalam contoh berikut, kami memplot himpunan, di mana

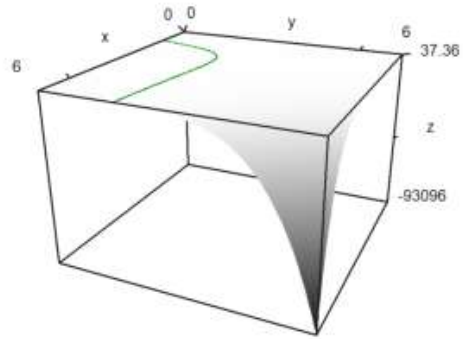
$$f(x, y) = x^y - y^x = 0$$

Kami menggunakan satu garis tipis untuk garis level.

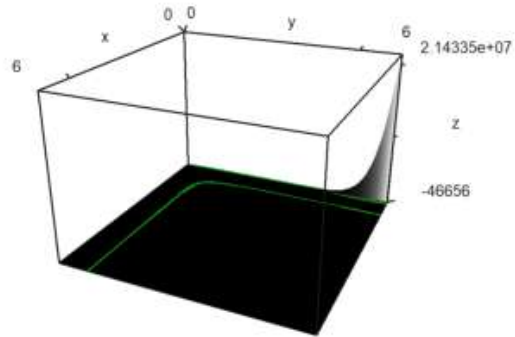
```
>plot3d("x^y-y^x",level=0,a=0,b=6,c=0,d=6,contourcolor=red,n=100):
```

```
>plot3d("x^2y-2y^x",level=0,a=0,b=6,c=0,d=6,contourcolor=green,n=100):
```

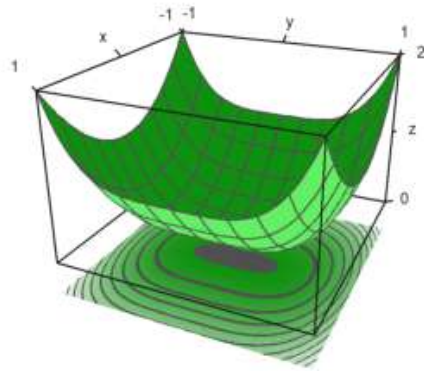


```
>plot3d("x^2-y-y^2",level=0,a=0,b=6,c=0,d=6,contourcolor=green,n=100):
```

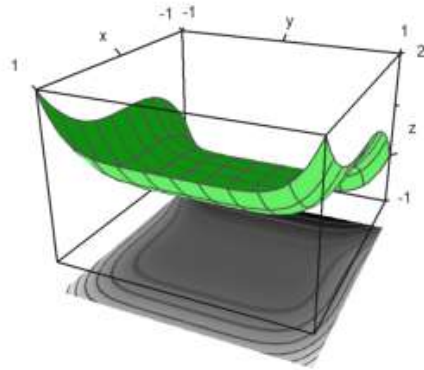


Dimungkinkan untuk menunjukkan bidang kontur di bawah plot. Warna dan jarak ke plot dapat ditentukan.

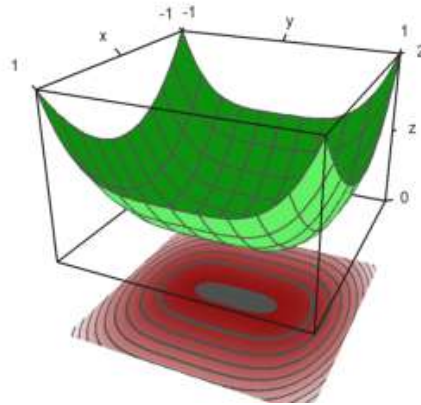
```
>plot3d("x^2+y^4",>cp,cpcolor=green,cpdelta=0.2):
```



```
>plot3d("x^3+y^6",>cp,cpcolor=black,cpdelta=0.2):
```

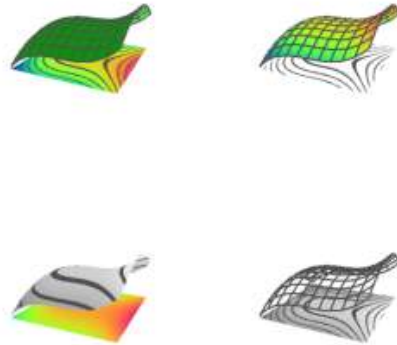


```
>plot3d("x^2+y^4",>cp,cpcolor=red,cpdelta=0.5):
```

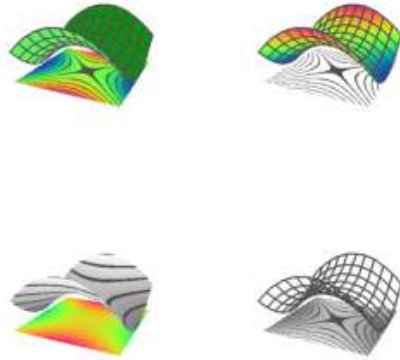


Berikut adalah beberapa gaya lagi. Kami selalu mematikan frame, dan menggunakan berbagai skema warna untuk plot dan grid.

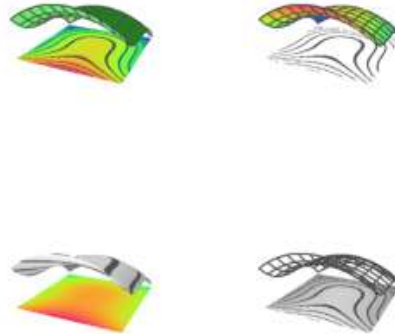
```
>figure(2,2); ...  
>expr="y^3-x^2"; ...  
>figure(1); ...  
> plot3d(expr,<frame,>cp,cpcolor=spectral); ...  
>figure(2); ...  
> plot3d(expr,<frame,>spectral,grid=10,cp=2); ...  
>figure(3); ...  
> plot3d(expr,<frame,>contour,color=gray,nc=5,cp=3,cpcolor=greenred); ...  
>figure(4); ...  
> plot3d(expr,<frame,>hue,grid=10,>transparent,>cp,cpcolor=gray); ...  
>figure(0):
```



```
>figure(2,2); ...  
>expr="x^2 - y^2"; ...  
>figure(1); ...  
> plot3d(expr,<frame,>cp,cpcolor=spectral); ...  
>figure(2); ...  
> plot3d(expr,<frame,>spectral,grid=10,cp=2); ...  
>figure(3); ...  
> plot3d(expr,<frame,>contour,color=gray,nc=5,cp=3,cpcolor=greenred); ...  
>figure(4); ...  
> plot3d(expr,<frame,>hue,grid=10,>transparent,>cp,cpcolor=gray); ...  
>figure(0):
```



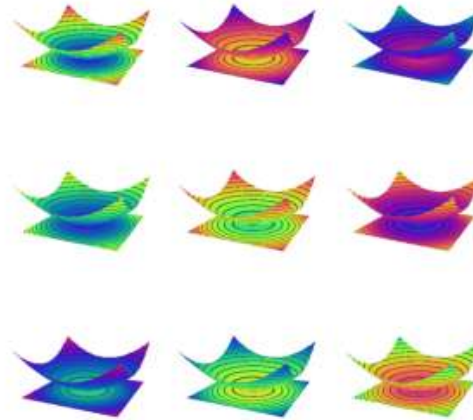
```
>figure(2,2); ...  
>expr="x^3 - y^2"; ...  
>figure(1); ...  
> plot3d(expr,<frame,>cp,cpcolor=spectral); ...  
>figure(2); ...  
> plot3d(expr,<frame,>spectral,grid=10,cp=2); ...  
>figure(3); ...  
> plot3d(expr,<frame,>contour,color=gray,nc=5,cp=3,cpcolor=greenred); ...  
>figure(4); ...  
> plot3d(expr,<frame,>hue,grid=10,>transparent,>cp,cpcolor=gray); ...  
>figure(0):
```

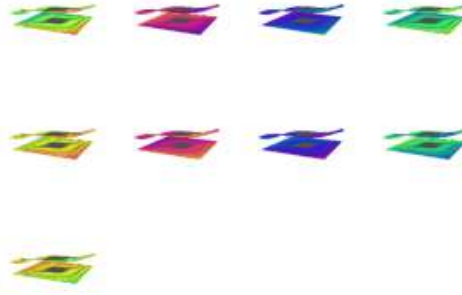
Ada beberapa skema spektral lainnya, bernomor dari 1 hingga 9. Tetapi Anda juga dapat menggunakan warna=nilai, di mana nilai

- spektral: untuk rentang dari biru ke merah
- putih: untuk rentang yang lebih redup
- kuningbiru, ungu hijau, birukuning, hijaumerah
- birukuning, hijau ungu, kuning biru, merah hijau

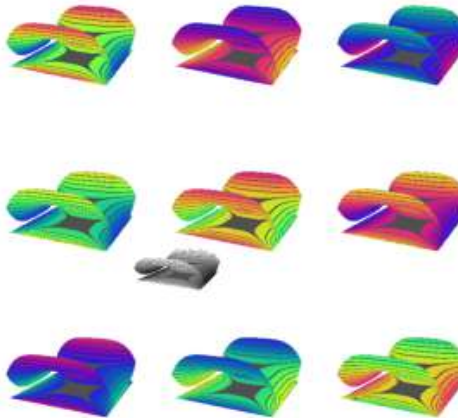
```
>figure(3,3); ...
>for i=1:9; ...
> figure(i); plot3d("x^2+y^2",spectral=i,>contour,>cp,<frame,zoom=4); ...
>end; ...
>figure(0):
```



```
>figure(4,4); ...  
>for i=1:9; ...  
> figure(i); plot3d("x^3+y^3",spectral=i,>contour,>cp,<frame,zoom=4); ...  
>end; ...  
>figure(0):
```



```
>figure(3,3); ...  
>for i=1:10; ...  
> figure(i); plot3d("x^4-y^4",spectral=i,>contour,>cp,<frame,zoom=4); ...  
>end; ...  
>figure(0):
```



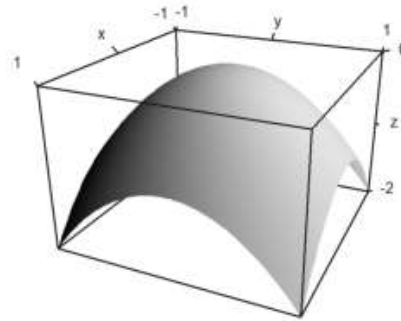
Sumber cahaya dapat diubah dengan l dan tombol kursor selama interaksi pengguna. Itu juga dapat diatur dengan parameter.

- cahaya: arah untuk cahaya
- amb: cahaya sekitar antara 0 dan 1

Perhatikan bahwa program tidak membuat perbedaan antara sisi plot. Tidak ada bayangan. Untuk ini, Anda perlu Povray.

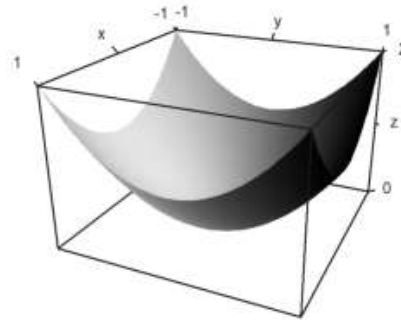
```
>plot3d("-x^2-y^2", ...  
> hue=true,light=[0,1,1],amb=0,user=true, ...  
> title="Press l and cursor keys (return to exit)":
```

Press l and cursor keys (return to exit)



```
>plot3d("x^2+y^2", ...  
> hue=true,light=[0,1,1],amb=0,user=true, ...  
> title="Press l and cursor keys (return to exit)":
```

Press I and cursor keys (return to exit)



Parameter warna mengubah warna permukaan. Warna garis level juga dapat diubah.

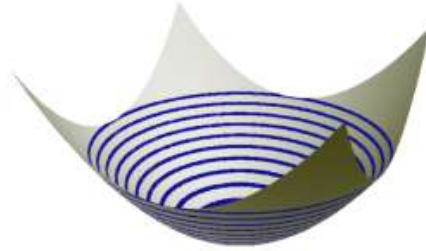
```
>plot3d("-x^2-y^2",color=rgb(0.2,0.2,0),hue=true,frame=false, ...  
> zoom=3,contourcolor=red,level=-2:0.1:1,d1=0.01):
```



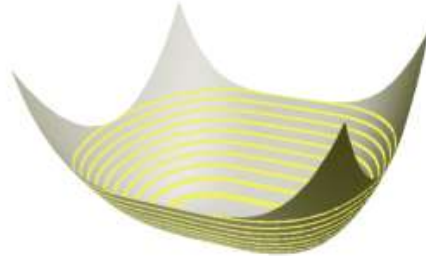
```
>plot3d("x^2-y^2",color=rgb(0.2,0.2,0),hue=true,frame=false, ...  
> zoom=3,contourcolor=green,level=-2:0.1:1,d1=0.01):
```



```
>plot3d("x^2+y^2",color=rgb(0.2,0.2,0),hue=true,frame=false, ...  
> zoom=3,contourcolor=blue,level=-2:0.1:1,d1=0.01):
```

```
>plot3d("x^2+y^4",color=rgb(0.2,0.2,0),hue=true,frame=false, ...  
> zoom=3,contourcolor=yellow,level=-2:0.1:1,d1=0.01):
```

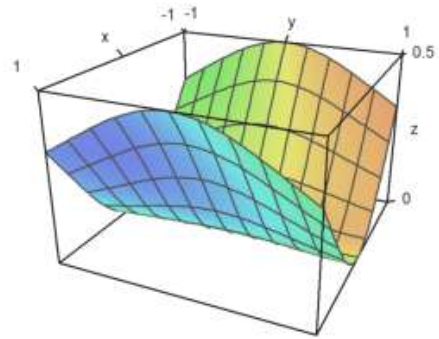


```
>plot3d("x^2+y^3",color=rgb(0.2,0.2,0),hue=true,frame=false, ...  
> zoom=2,contourcolor=green,level=-2:0.1:1,d1=0.01):
```

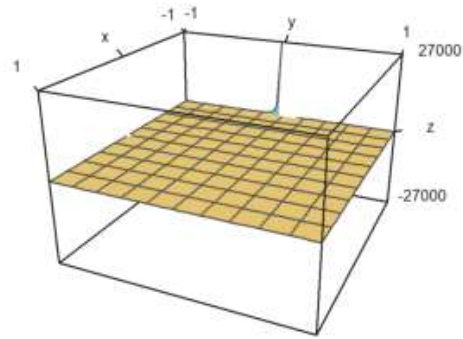


Warna 0 memberikan efek pelangi khusus.

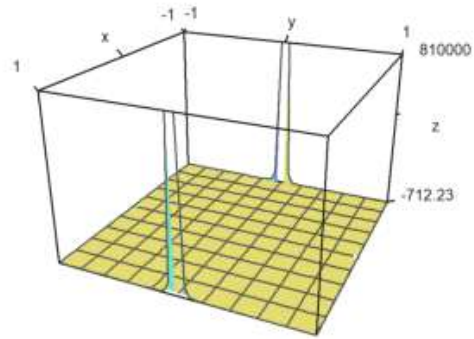
```
>plot3d("x^2/(x^2+y^2+1)",color=0,hue=true,grid=10):
```



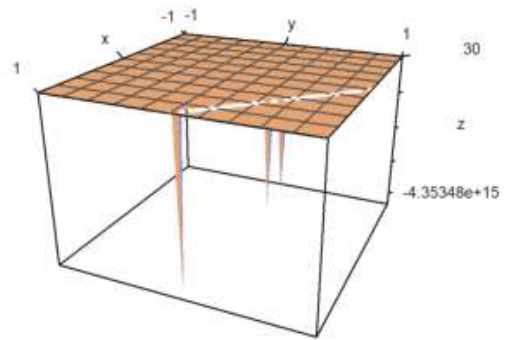
```
>plot3d("x^3/(x^3+y^3+1)",color=0,hue=true,grid=10):
```



```
>plot3d("x^4/(x^4+y^4-1)",color=0,hue=true,grid=10):
```

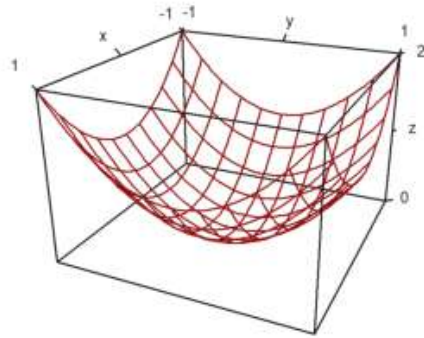


```
>plot3d("x/(x+y-1)",color=0,hue=true,grid=10):
```

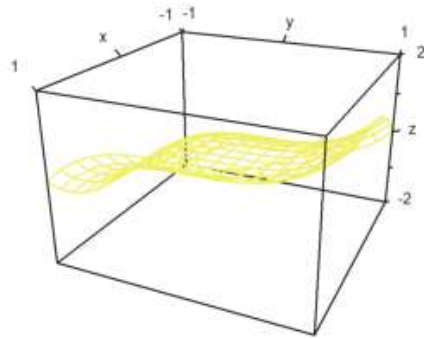


Permukaannya juga bisa transparan.

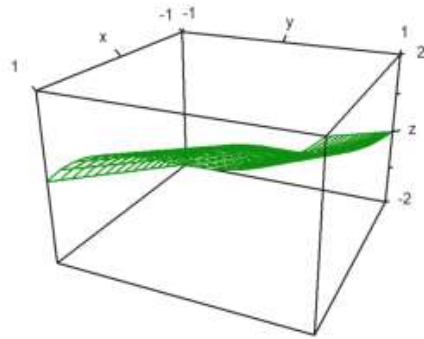
```
>plot3d("x^2+y^2",>transparent,grid=10,wirecolor=red):
```



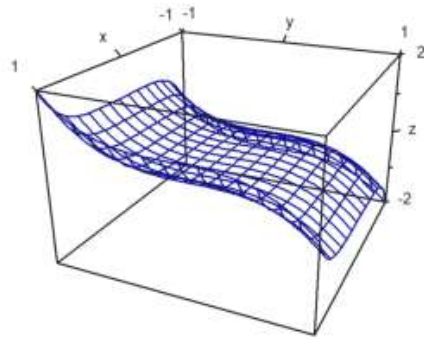
```
>plot3d("x^3+y^3",>transparent,grid=15,wirecolor=yellow):
```

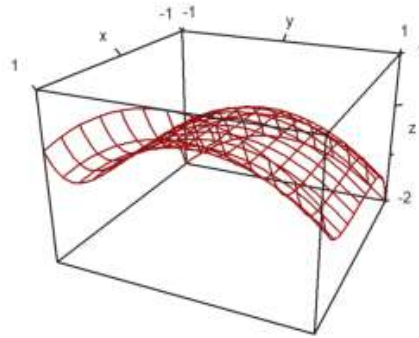
```
>plot3d("x+y^3",>transparent,grid=15,wirecolor=green):
```



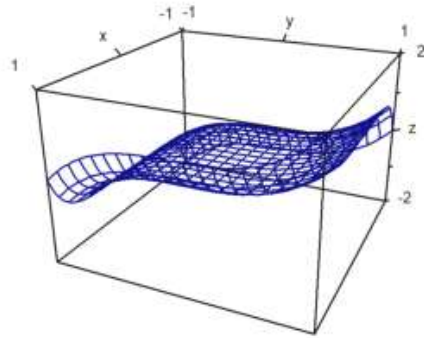
```
>plot3d("x^5-y^3",>transparent,grid=15,wirecolor=blue):
```



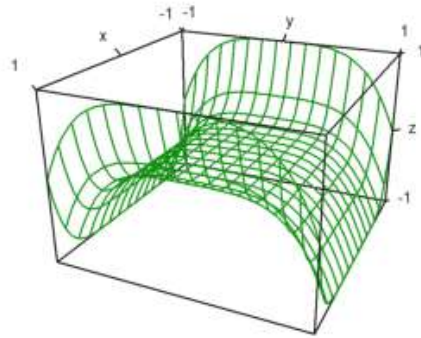
```
>plot3d("x^5-y^2",>transparent,grid=13,wirecolor=red):
```



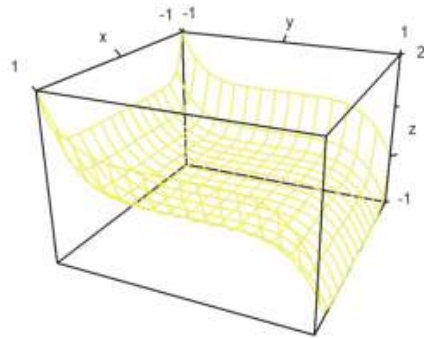
```
>plot3d("x^7+y^3",>transparent,grid=20,wirecolor=blue):
```



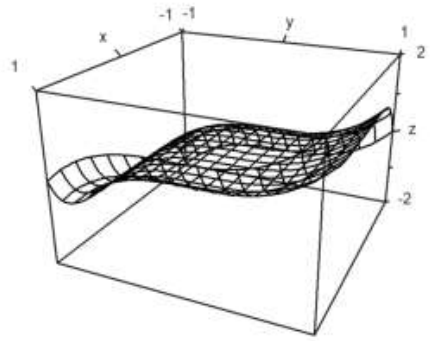
```
>plot3d("x^8-y^4",>transparent,grid=15,wirecolor=green):
```



```
>plot3d("x^10-y^5",>transparent,grid=15,wirecolor=yellow):
```



```
>plot3d("x^7+y^3",>transparent,grid=15,wirecolor=black):
```



Plot Implisit

Ada juga plot implisit dalam tiga dimensi. Euler menghasilkan pemotongan melalui objek. Fitur `plot3d` termasuk plot implisit. Plot-plot ini menunjukkan himpunan nol dari suatu fungsi dalam tiga variabel. Solusi dari

$$f(x, y, z) = 0$$

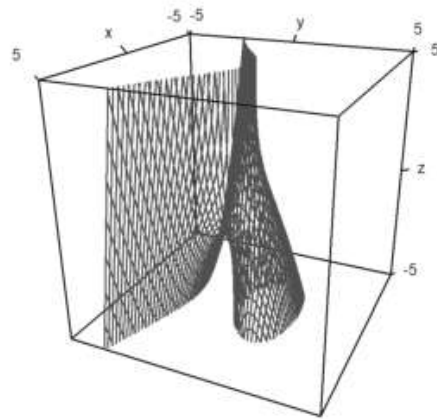
dapat divisualisasikan dalam potongan sejajar dengan bidang x - y -, x - z - dan y - z .

- `implicit=1`: potong sejajar dengan bidang y - z
- `implicit=2`: potong sejajar dengan bidang x - z
- `implicit=4`: potong sejajar dengan bidang x - y

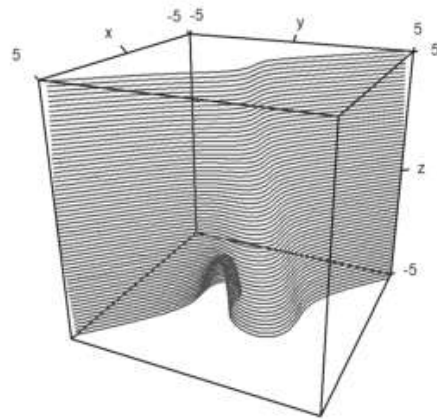
Tambahkan nilai-nilai ini, jika Anda suka. Dalam contoh kita plot

$$M = \{(x, y, z) : x^2 + y^3 + zy = 1\}$$

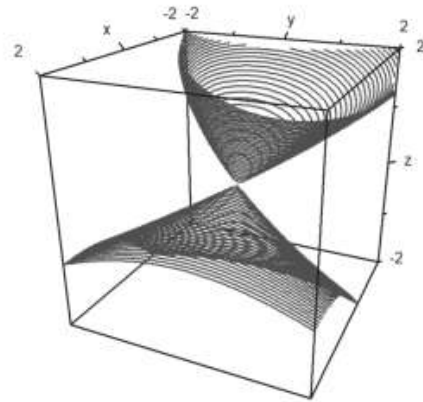
```
>plot3d("x^2+y^3+z*y-1",r=5,implicit=3):
```



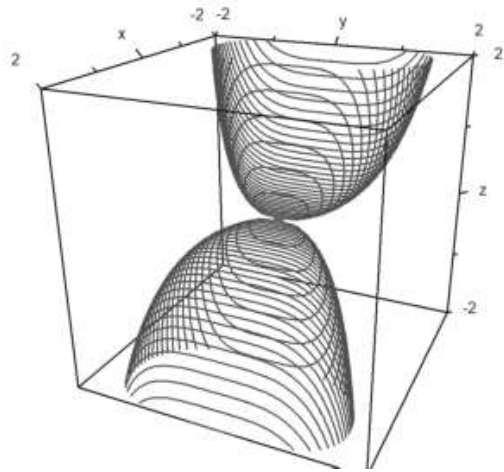
```
>plot3d("x^3+y^3+z*y-2",r=5,implicit=4):
```



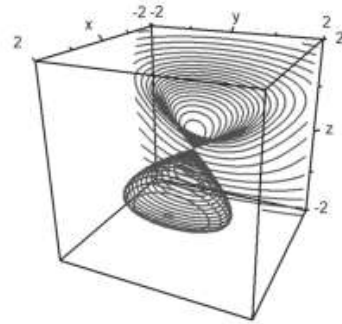
```
>plot3d("x^2+y^2+4*x*z+z^3",>implicit,r=2,zoom=2.5):
```



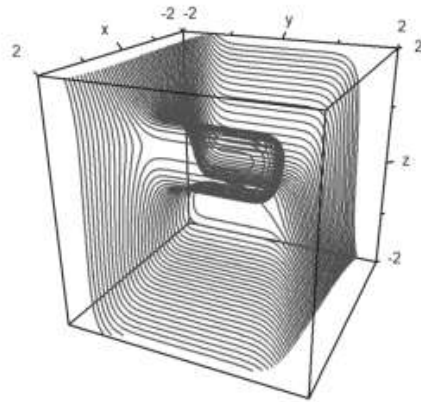
```
>plot3d("x^4+y^4+5*x*z+z^2",>implicit,r=2,zoom=3):
```



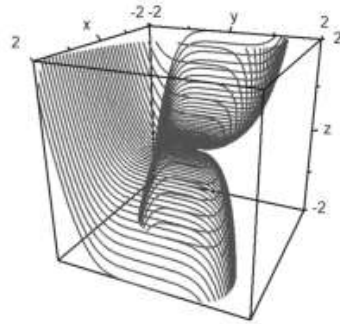
```
>plot3d("x^7+y^2+3*x*z+z^4",>implicit,r=2,zoom=2):
```



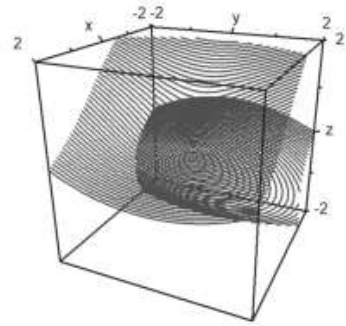
```
>plot3d("x^5+y^9-3*x*z+z^5",>implicit,r=2,zoom=2.5):
```



```
>plot3d("x^4+y^5+5*x*z+z^2",>implicit,r=2,zoom=2):
```



```
>plot3d("x^2-y^2+4*x*z+z^3",>implicit,r=2,zoom=2):
```

Merencanakan Data 3D

Sama seperti `plot2d`, `plot3d` menerima data. Untuk objek 3D, Anda perlu menyediakan matriks nilai x -, y - dan z , atau tiga fungsi atau ekspresi $f_x(x,y)$, $f_y(x,y)$, $f_z(x,y)$.

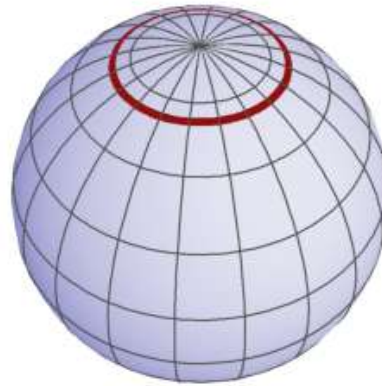
$$\gamma(t, s) = (x(t, s), y(t, s), z(t, s))$$

Karena x,y,z adalah matriks, kita asumsikan bahwa (t,s) melalui sebuah kotak persegi. Hasilnya, Anda dapat memplot gambar persegi panjang di ruang angkasa.

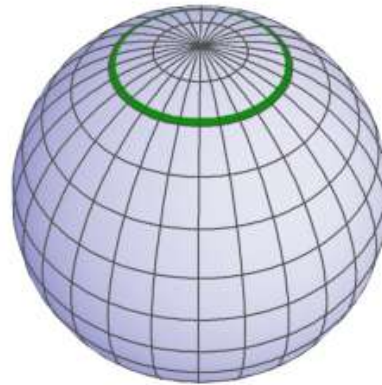
Anda dapat menggunakan bahasa matriks Euler untuk menghasilkan koordinat secara efektif.

Dalam contoh berikut, kami menggunakan vektor nilai t dan vektor kolom nilai s untuk membuat parameter permukaan bola. Dalam gambar kita dapat menandai daerah, dalam kasus kita daerah kutub.

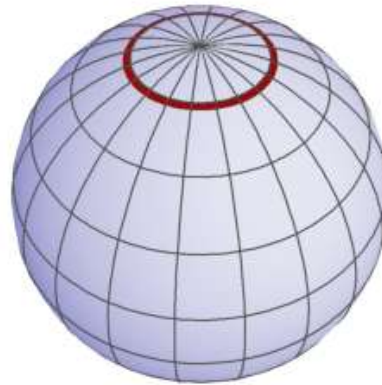
```
>t=linspace(0,2pi,180); s=linspace(-pi/2,pi/2,90)'; ...
>x=cos(s)*cos(t); y=cos(s)*sin(t); z=sin(s); ...
>plot3d(x,y,z,>hue, ...
>color=blue,<frame,grid=[10,20], ...
>values=s,contourcolor=red,level=[90°-24°;90°-22°], ...
>scale=1.4,height=50°):
```



```
>t=linspace(0,2pi,180); s=linspace(-pi/2,pi/2,90)'; ...  
>x=cos(s)*cos(t); y=cos(s)*sin(t); z=sin(s); ...  
>plot3d(x,y,z,>hue, ...  
>color=blue,<frame,grid=[15,30], ...  
>values=s,contourcolor=green,level=[90°-24°;90°-22°], ...  
>scale=1.4,height=50°):
```

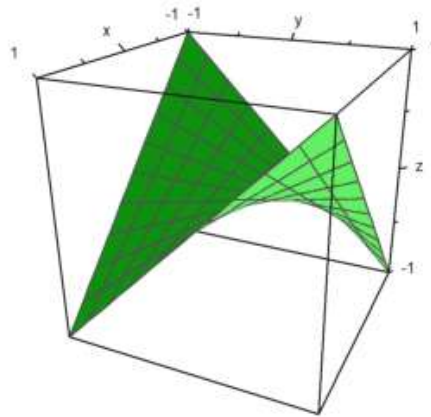


```
>t=linspace(0,2pi,180); s=linspace(-pi/2,pi/2,90)'; ...  
>x=cos(s)*cos(t); y=cos(s)*sin(t); z=sin(s); ...  
>plot3d(x,y,z,>hue, ...  
>color=blue,<frame,grid=[10,20], ...  
>values=s,contourcolor=red,level=[90°-20°;90°-18°], ...  
>scale=1.4,height=50°):
```

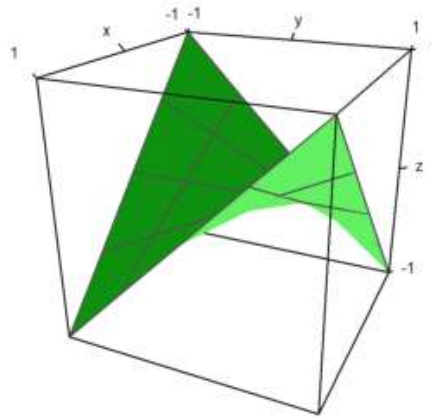


Berikut adalah contoh, yang merupakan grafik fungsi.

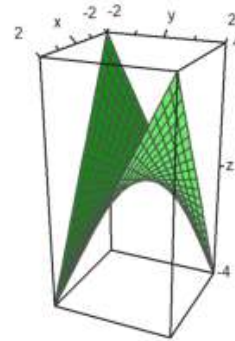
```
>t=-1:0.1:1; s=(-1:0.1:1)'; plot3d(t,s,t*s,grid=10):
```



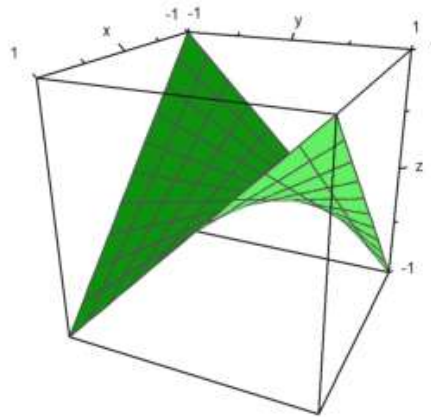
```
>t=-1:0.5:1; s=(-1:0.5:1)'; plot3d(t,s,t*s,grid=10):
```



```
>t=-2:0.2:2; s=(-2:0.2:2)'; plot3d(t,s,t*s,grid=15):
```



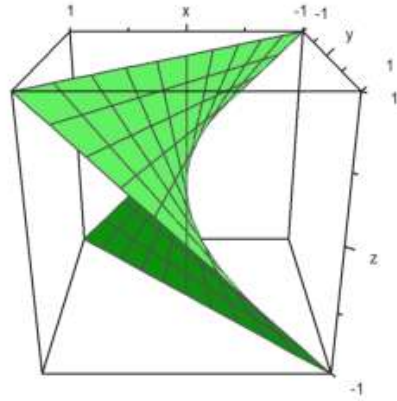
```
>t=-1:0.1:1; s=(-1:0.1:1)'; plot3d(t,s,t*s,grid=11):
```

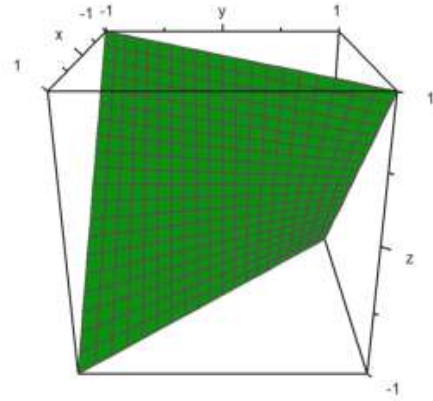
Namun, kita bisa membuat segala macam permukaan. Berikut adalah permukaan yang sama dengan fungsi

$$x = yz$$

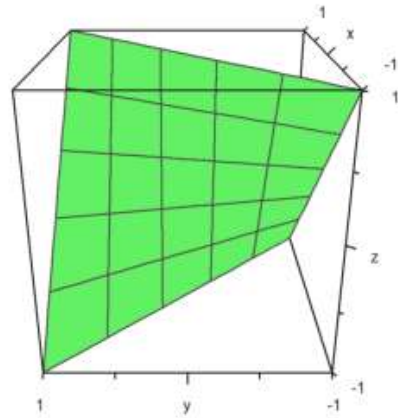
```
>plot3d(t*s,t,s,angle=180°,grid=10):
```



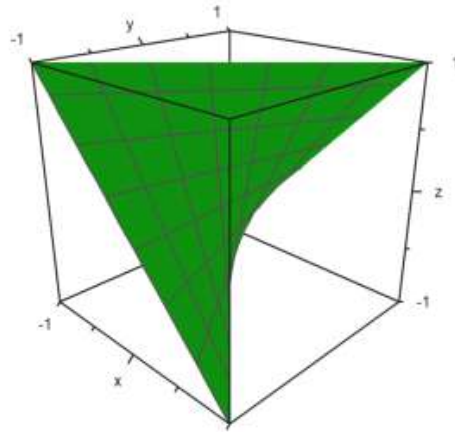
```
>plot3d(t*s,t,s,angle=90°,grid=16):
```



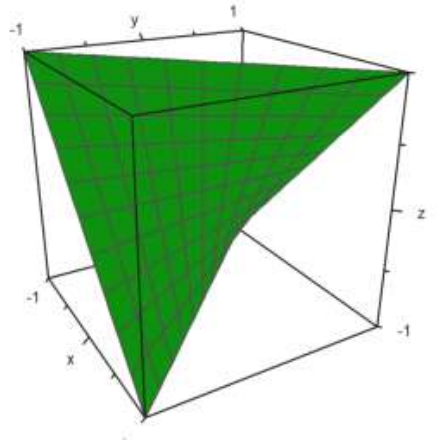
```
>plot3d(t*s,t,s,angle=270°,grid=5):
```



```
>plot3d(t*s,t,s,angle=45°,grid=7):
```



```
>plot3d(t*s,t,s,angle=60°,grid=10):
```



Dengan lebih banyak usaha, kami dapat menghasilkan banyak permukaan.

Dalam contoh berikut, kita membuat tampilan bayangan dari bola yang terdistorsi. Koordinat biasa untuk bola adalah

$$\gamma(t, s) = (\cos(t) \cos(s), \sin(t) \sin(s), \cos(s))$$

dengan

$$0 \leq t \leq 2\pi, \quad -\frac{\pi}{2} \leq s \leq \frac{\pi}{2}.$$

Kami mendistorsi ini dengan sebuah faktor

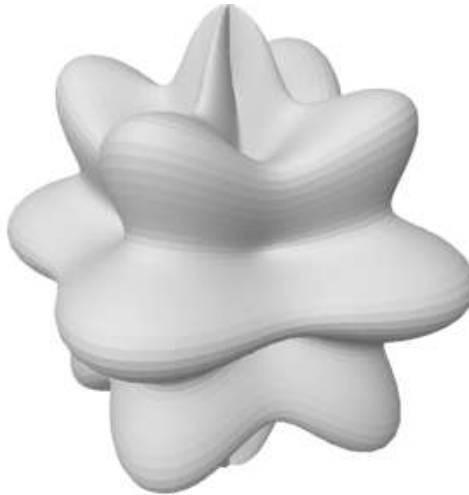
$$d(t, s) = \frac{\cos(4t) + \cos(8s)}{4}.$$

```
>t=linspace(0,2pi,320); s=linspace(-pi/2,pi/2,160)'; ...  
>d=1+0.2*(cos(4*t)+cos(8*s)); ...  
>plot3d(cos(t)*cos(s)*d,sin(t)*cos(s)*d,sin(s)*d,hue=1, ...  
> light=[1,0,1],frame=0,zoom=5):
```



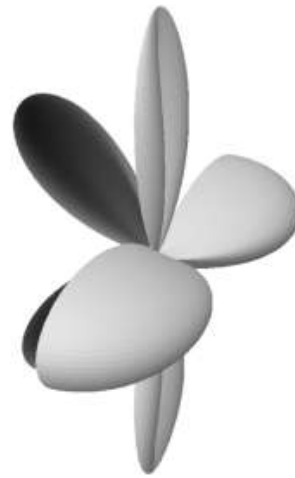
```
>t=linspace(0,2pi,270); s=linspace(-pi/2,pi/2,80)'; ...  
>d=1+0.2*(cos(4*t)+cos(8*s)); ...  
>plot3d(cos(t)*cos(s)*d,sin(t)*cos(s)*d,sin(s)*d,hue=1, ...
```

```
> light=[1,0,1],frame=0,zoom=5):
```

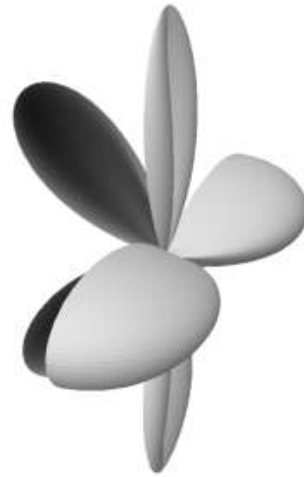


Dalam percobaan saya ini ternyata dengan fungsi tersebut dapat membentuk sebuah bunga. berikut beberapa bentuk contohnya

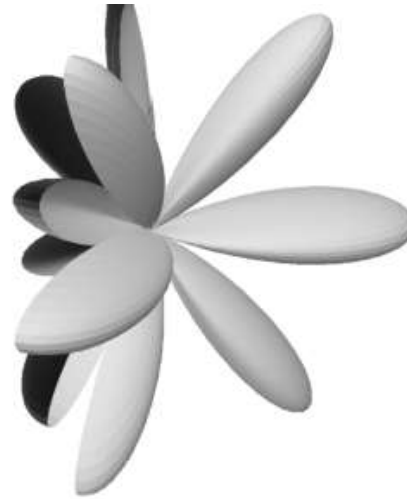
```
>t=linspace(0,pi,320); s=linspace(-pi/2,pi/2,150)'; ...  
>d=1+5*(cos(2*t)+cos(4*s)); ...  
>plot3d(cos(t)*cos(s)*d,sin(t)*cos(s)*d,sin(s)*d,hue=1, ...  
> light=[1,0,1],frame=0,zoom=5):
```

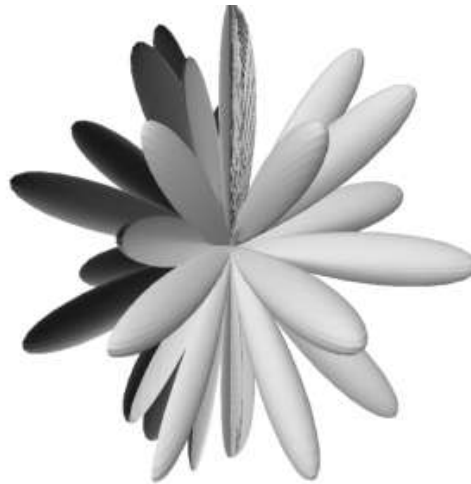
```
>t=linspace(0,pi,320); s=linspace(-pi/2,pi/2,150)'; ...  
>d=1+7*(cos(2*t)+cos(4*s)); ...  
>plot3d(cos(t)*cos(s)*d,sin(t)*cos(s)*d,sin(s)*d,hue=1, ...  
> light=[1,0,1],frame=0,zoom=5):
```



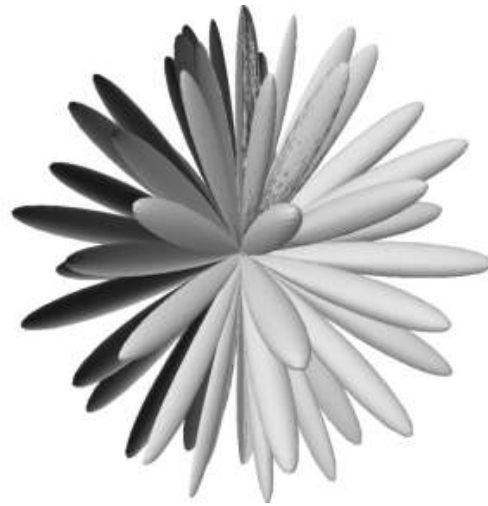
```
>t=linspace(0,pi,260); s=linspace(-pi/2,pi/2,120)'; ...  
>d=1+10*(cos(3*t)+cos(7*s)); ...  
>plot3d(cos(t)*cos(s)*d,sin(t)*cos(s)*d,sin(s)*d,hue=1, ...  
> light=[1,0,1],frame=0,zoom=5):
```



```
>t=linspace(0,pi,260); s=linspace(-pi/2,pi/2,120)'; ...  
>d=1+15*(cos(7*t)+cos(8*s)); ...  
>plot3d(cos(t)*cos(s)*d,sin(t)*cos(s)*d,sin(s)*d,hue=1, ...  
> light=[1,0,1],frame=0,zoom=5):
```



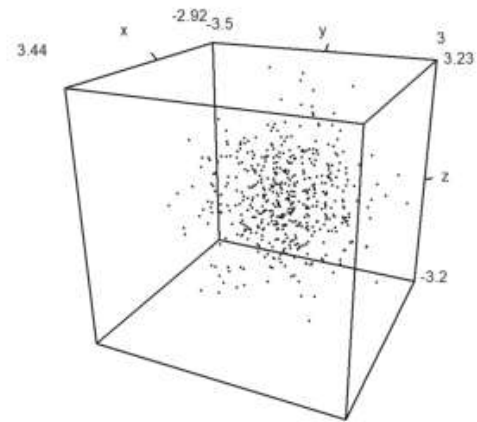
```
>t=linspace(0,pi,270); s=linspace(-pi/2,pi/2,180)'; ...  
>d=1+15*(cos(10*t)+cos(12*s)); ...  
>plot3d(cos(t)*cos(s)*d,sin(t)*cos(s)*d,sin(s)*d,hue=1, ...  
> light=[1,0,1],frame=0,zoom=5):
```



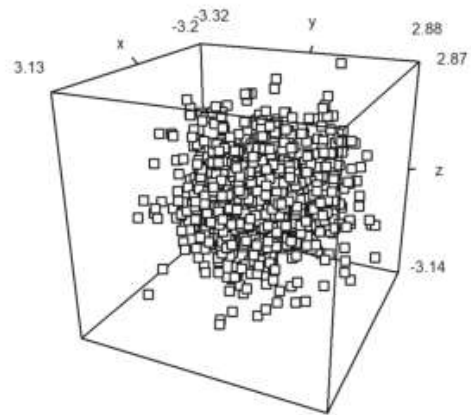
Tentu saja, titik cloud juga dimungkinkan. Untuk memplot data titik dalam ruang, kita membutuhkan tiga vektor untuk koordinat titik-titik tersebut.

Gayanya sama seperti di plot2d dengan `points=true`;

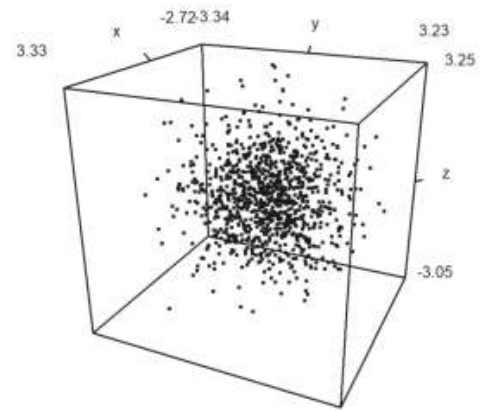
```
>n=500; ...  
> plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style="."):
```



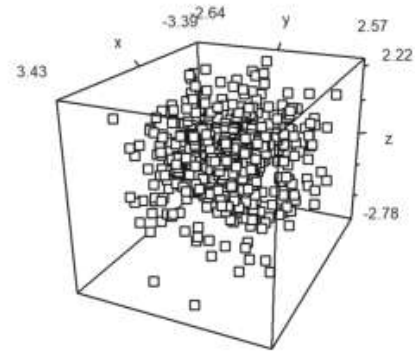
```
>n=1000; ...  
> plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style="--"):
```



```
>n=1000; ...  
> plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style=".."):
```

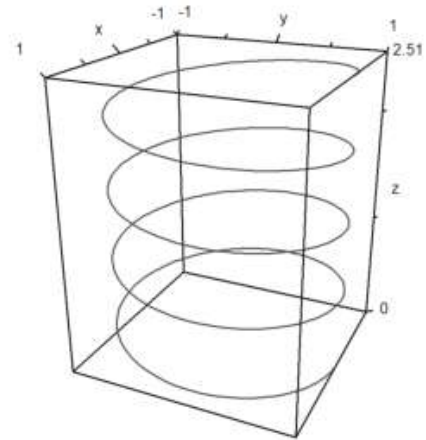


```
>n=500; ...  
> plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style="**"):
```

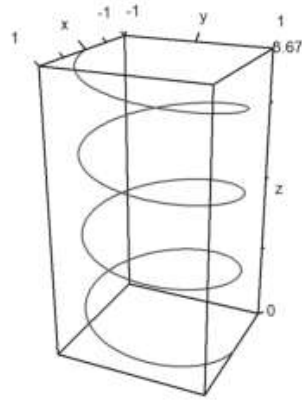



Dimungkinkan juga untuk memplot kurva dalam 3D. Dalam hal ini, lebih mudah untuk menghitung titik-titik kurva. Untuk kurva di pesawat kami menggunakan urutan koordinat dan parameter `wire=true`.

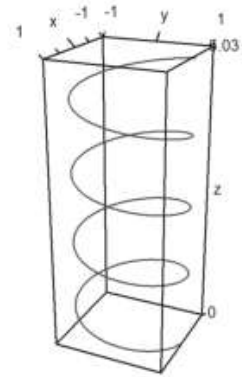
```
>t=linspace(0,8pi,500); ...  
>plot3d(sin(t),cos(t),t/10,>wire,zoom=3):
```



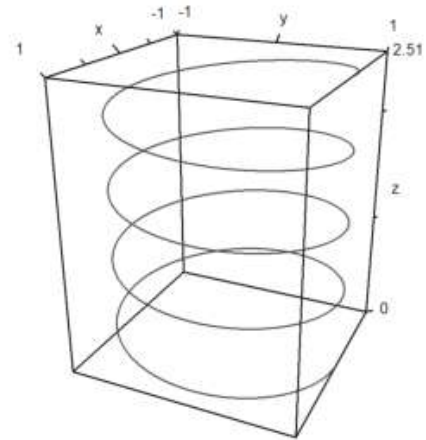
```
>t=linspace(0,7pi,1000); ...  
>plot3d(sin(t),cos(t),t/6,>wire,zoom=3):
```



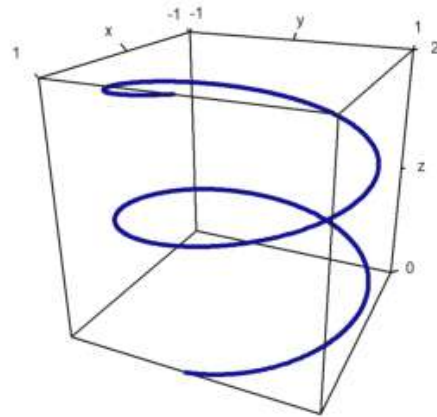
```
>t=linspace(0,8*pi,1000); ...  
>plot3d(sin(t),cos(t),t/5,>wire,zoom=3):
```



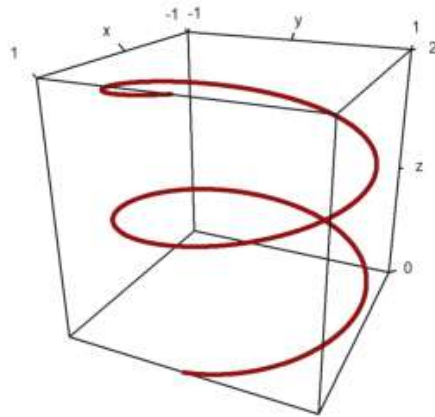
```
>t=linspace(0,8*pi,3000); ...  
>plot3d(sin(t),cos(t),t/10,>wire,zoom=3):
```



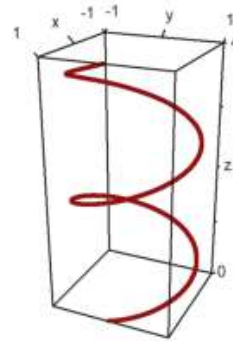
```
>t=linspace(0,4pi,1000); plot3d(cos(t),sin(t),t/2pi,>wire, ...  
>linewidth=3,wirecolor=blue):
```



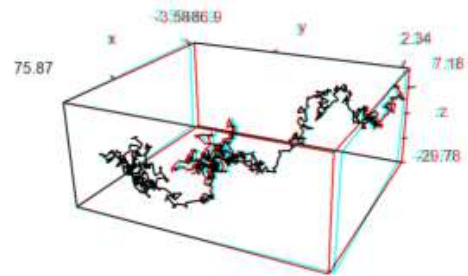
```
>t=linspace(0,4pi,4000); plot3d(cos(t),sin(t),t/2pi,>wire, ...  
>linewidth=3,wirecolor=red):
```



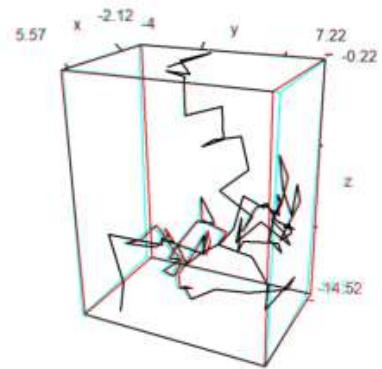
```
>t=linspace(0,4pi,4000); plot3d(cos(t),sin(t),t/pi,>wire, ...  
>linewidth=3,wirecolor=red):
```



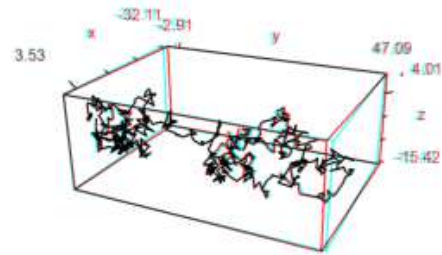
```
>X=cumsum(normal(3,1000)); ...  
> plot3d(X[1],X[2],X[3],>anaglyph,>wire):
```

```
>X=cumsum(normal(5,100)); ...  
> plot3d(X[1],X[2],X[3],>anaglyph,>wire):
```

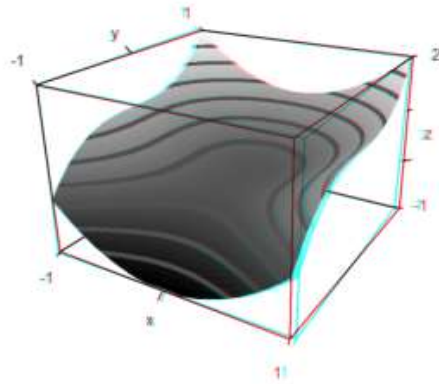


```
>X=cumsum(normal(6,600)); ...  
> plot3d(X[1],X[2],X[3],>anaglyph,>wire):
```

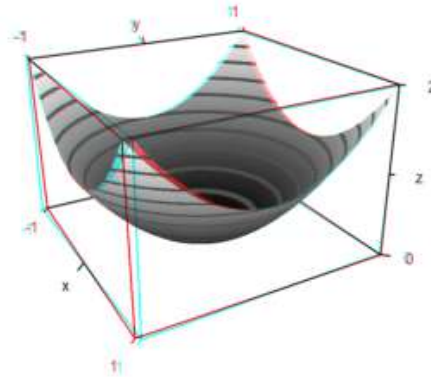


EMT juga dapat memplot dalam mode anaglyph. Untuk melihat plot seperti itu, Anda memerlukan kacamata merah/sian.

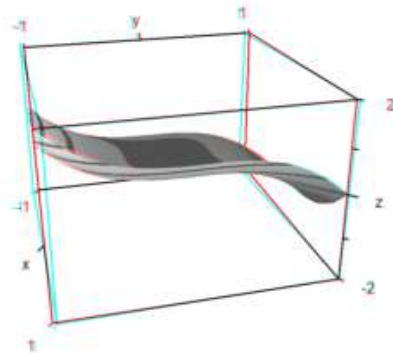
```
> plot3d("x^2+y^3",>anaglyph,>contour,angle=30°):
```



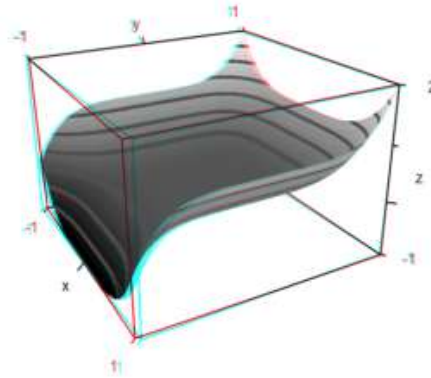
```
>plot3d("x^2+y^2",>anaglyph,>contour,angle=60°):
```



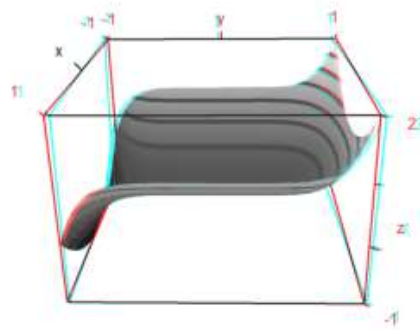
```
> plot3d("x^3-y^3",>anaglyph,>contour,angle=75°):
```



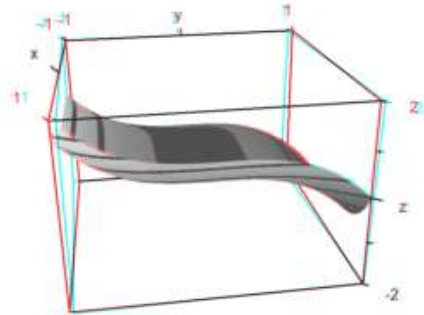
```
> plot3d("x^4+y^5",>anaglyph,>contour,angle=60°):
```



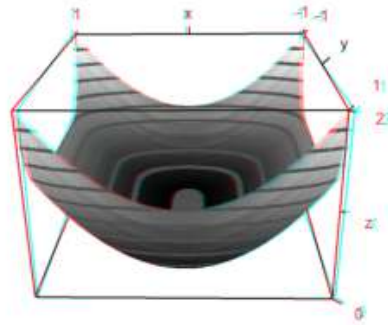
```
> plot3d("x^2+y^7",>anaglyph,>contour,angle=90°):
```



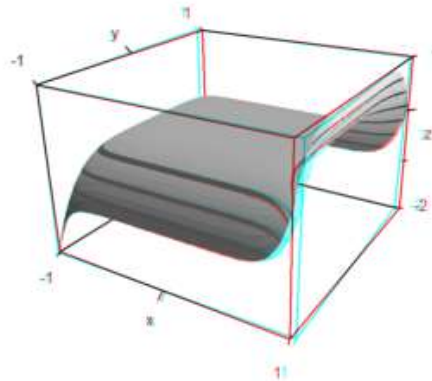
```
> plot3d("x^5-y^3",>anaglyph,>contour,angle=80°):
```

```
> plot3d("x^2+y^8",>anaglyph,>contour,angle=180):
```

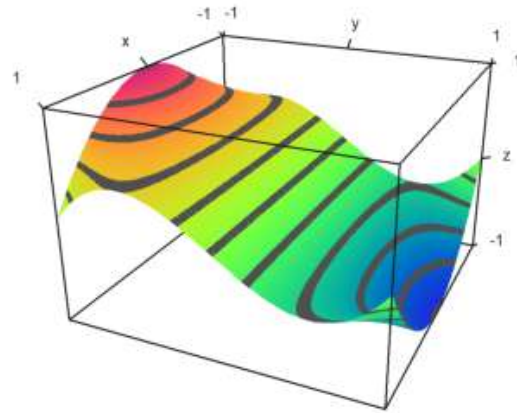


```
> plot3d("x^9-y^6",>anaglyph,>contour,angle=30°):
```

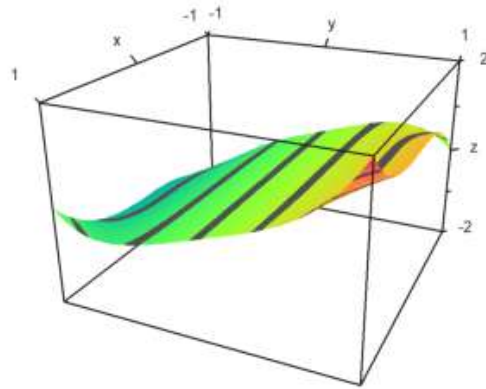


Seringkali, skema warna spektral digunakan untuk plot. Ini menekankan ketinggian fungsi.

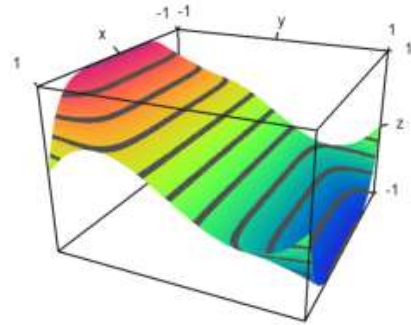
```
>plot3d("x^2*y^3-y",>spectral,>contour,zoom=3.2):
```



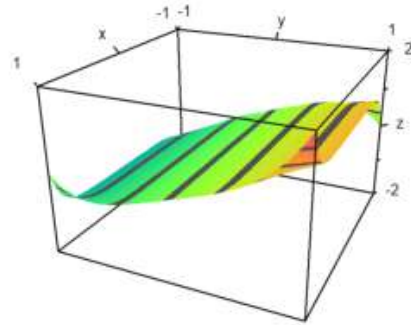
```
>plot3d("x^3*y^4+y",>spectral,>contour,zoom=3):
```



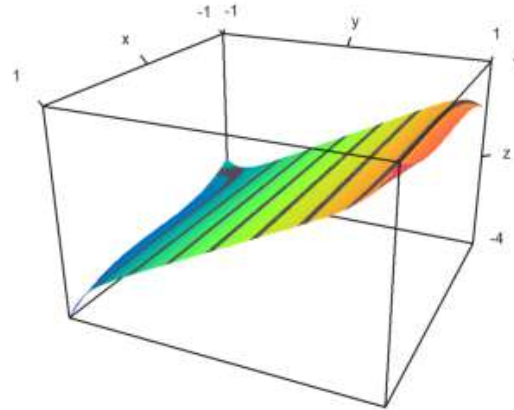
```
>plot3d("x^6*y^3-y",>spectral,>contour, zoom=2.5):
```



```
>plot3d("x^9*y^4+y",>spectral,>contour,zoom=2.5):
```



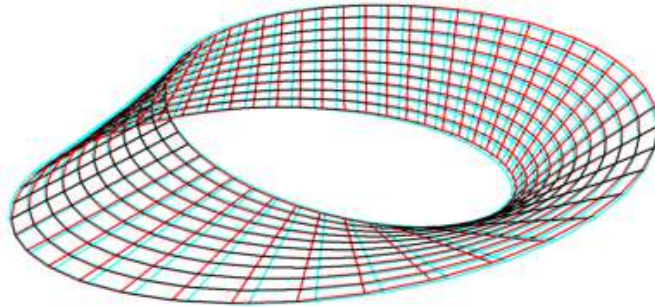
```
>plot3d("x^3*y^9+3y",>spectral,>contour,zoom=3.2):
```



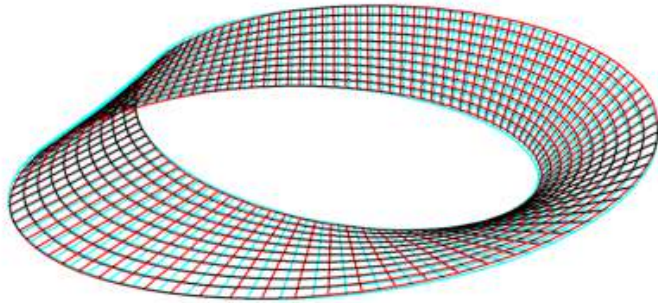
Euler juga dapat memplot permukaan berparameter, ketika parameternya adalah nilai x -, y -, dan z dari gambar kotak persegi panjang dalam ruang.

Untuk demo berikut, kami mengatur parameter u - dan v -, dan menghasilkan koordinat ruang dari ini.

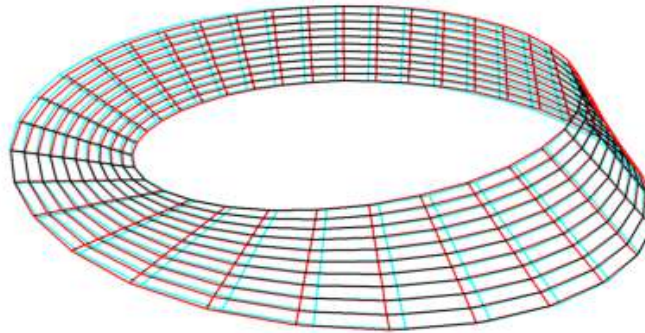
```
>u=linspace(-1,1,10); v=linspace(0,2*pi,50)'; ...
>X=(3+u*cos(v/2))*cos(v); Y=(3+u*cos(v/2))*sin(v); Z=u*sin(v/2); ...
>plot3d(X,Y,Z,>anaglyph,<frame,>wire,scale=2.3):
```

```
>u=linspace(-1,1,10); v=linspace(0,2*pi,100)'; ...  
>X=(4+u*cos(v/2))*cos(v); Y=(4+u*cos(v/2))*sin(v); Z=u*sin(v/2); ...  
>plot3d(X,Y,Z,>anaglyph,<frame,>wire,scale=2.3):
```

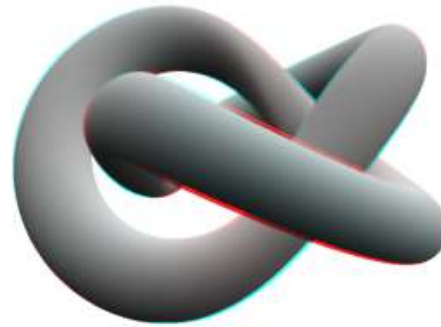


```
>u=linspace(-1,1,10); v=linspace(0,4*pi,70)'; ...  
>X=(5-u*cos(v/2))*cos(v); Y=(5-u*cos(v/2))*sin(v); Z=u*sin(v/2); ...  
>plot3d(X,Y,Z,>anaglyph,<frame,>wire,scale=2.3):
```

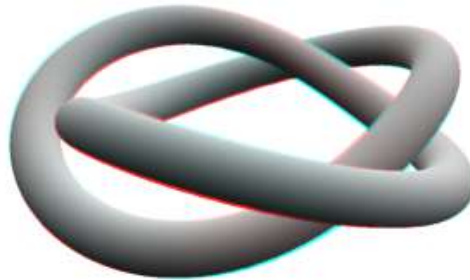


Berikut adalah contoh yang lebih rumit, yang megah dengan kacamata merah/sian.

```
>u=linspace(-pi,pi,160); v=linspace(-pi,pi,400)'; ...  
>x=(4*(1+.25*sin(3*v))+cos(u))*cos(2*v); ...  
>y=(4*(1+.25*sin(3*v))+cos(u))*sin(2*v); ...  
> z=sin(u)+2*cos(3*v); ...  
>plot3d(x,y,z,frame=0,scale=1.5,hue=1,light=[1,0,-1],zoom=2.8,>anaglyph):
```



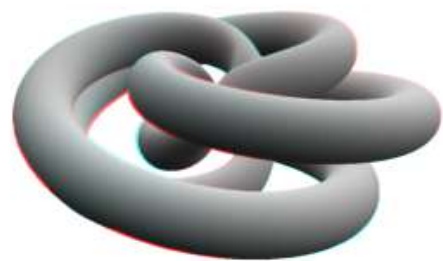
```
>u:=linspace(-pi,pi,160); v:=linspace(-pi,pi,500)'; ...  
>x:=(4*(2+.25*sin(3*v))+cos(u))*cos(2*v); ...  
>y:=(4*(2+.25*sin(3*v))+cos(u))*sin(2*v); ...  
> z=sin(u)+2*cos(3*v); ...  
>plot3d(x,y,z,frame=0,scale=1.5,hue=1,light=[1,0,-1],zoom=3,>anaglyph):
```



```
>u:=linspace(-pi,pi,170); v:=linspace(-pi,pi,700)'; ...  
>x:=(5*(1+.25*sin(3*v))+cos(u))*cos(2*v); ...  
>y:=(5*(1+.25*sin(3*v))+cos(u))*sin(2*v); ...  
> z=sin(u)+2*cos(3*v); ...  
>plot3d(x,y,z,frame=0,scale=1.5,hue=1,light=[1,0,-1],zoom=2,>anaglyph):
```



```
>u:=linspace(-pi,pi,160); v:=linspace(-pi,pi,400)'; ...  
>x:=(4*(1+.75*sin(3*v))+cos(u))*cos(4*v); ...  
>y:=(4*(1+.75*sin(3*v))+cos(u))*sin(4*v); ...  
> z=sin(u)+2*cos(3*v); ...  
>plot3d(x,y,z,frame=0,scale=1.5,hue=1,light=[1,0,-1],zoom=2.8,>anaglyph):
```



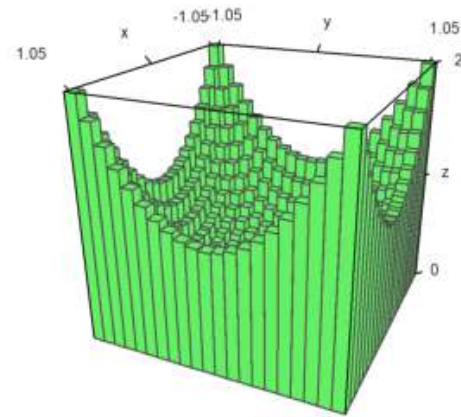
Plot bar juga dimungkinkan. Untuk ini, kita harus menyediakan

- x: vektor baris dengan $n+1$ elemen
- y: vektor kolom dengan $n+1$ elemen
- z: matriks nilai $n \times n$.

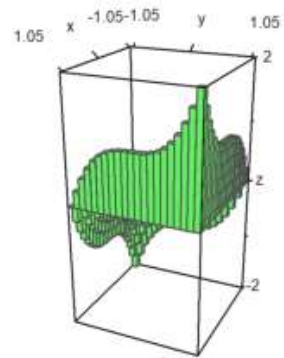
z bisa lebih besar, tetapi hanya nilai $n \times n$ yang akan digunakan.

Dalam contoh, pertama-tama kita menghitung nilainya. Kemudian kita sesuaikan x dan y, sehingga vektor berpusat pada nilai yang digunakan.

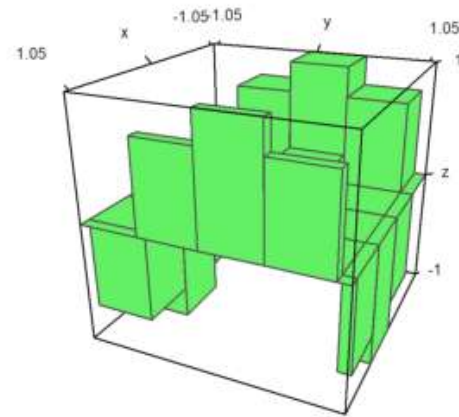
```
>x=-1:0.1:1; y=x'; z=x^2+y^2; ...  
>xa=(x|1.1)-0.05; ya=(y_1.1)-0.05; ...  
>plot3d(xa,ya,z,bar=true):
```

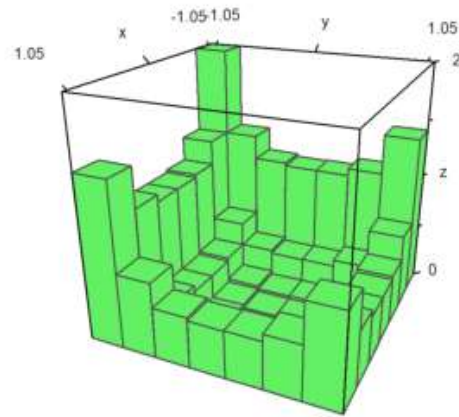
```
>x=-1:0.1:1; y=x'; z=x^3+y^3; ...  
>xa=(x|1.1)-0.05; ya=(y_1.1)-0.05; ...  
>plot3d(xa,ya,z,bar=true):
```



```
>x=-1:0.5:1; y=x'; z=x^2-y^2; ...  
>xa=(x|1.1)-0.05; ya=(y|1.1)-0.05; ...  
>plot3d(xa,ya,z,bar=true):
```

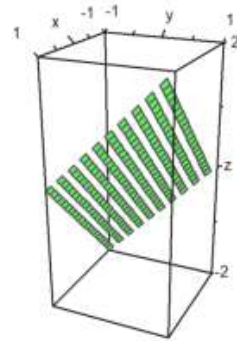


```
>x=-1:0.3:1; y=x'; z=x^4+y^4; ...  
>xa=(x|1.1)-0.05; ya=(y_1.1)-0.05; ...  
>plot3d(xa,ya,z,bar=true):
```

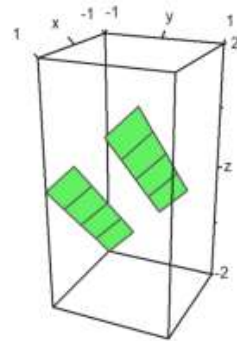


Dimungkinkan untuk membagi plot permukaan menjadi dua atau lebih bagian.

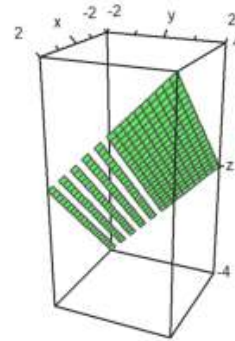
```
>x=-1:0.1:1; y=x'; z=x+y; d=zeros(size(x)); ...  
>plot3d(x,y,z,disconnect=2:2:20):
```



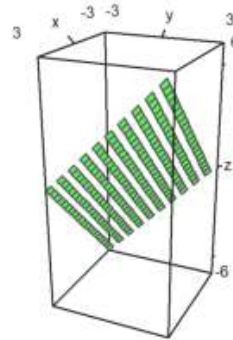
```
>x=-1:0.5:1; y=x'; z=x+y; d=zeros(size(x)); ...  
>plot3d(x,y,z,disconnect=2:2:20):
```



```
>x=-2:0.2:2; y=x'; z=x+y; d=zeros(size(x)); ...  
>plot3d(x,y,z,disconnect=2:2:10):
```

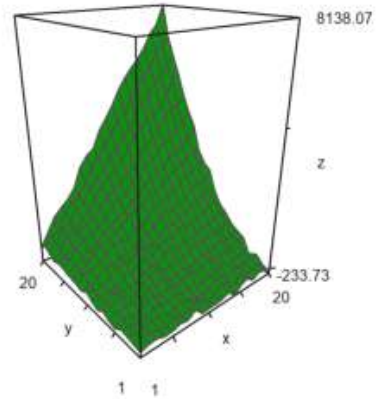


```
>x=-3:0.3:3; y=x'; z=x+y; d=zeros(size(x)); ...  
>plot3d(x,y,z,disconnect=2:2:30):
```

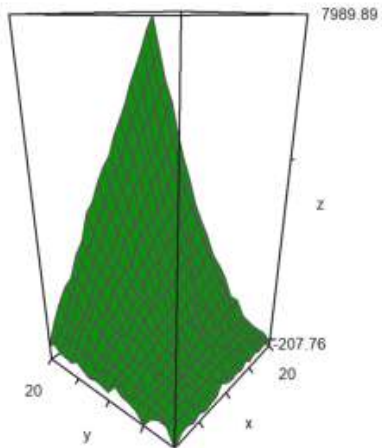


Jika memuat atau menghasilkan matriks data M dari file dan perlu memplotnya dalam 3D, Anda dapat menskalakan matriks ke $[-1,1]$ dengan `scale(M)`, atau menskalakan matriks dengan `>zscale`. Ini dapat dikombinasikan dengan faktor penskalaan individu yang diterapkan sebagai tambahan.

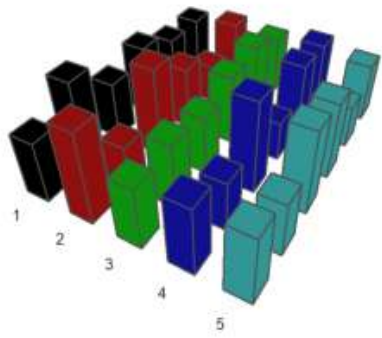
```
>i=1:20; j=i'; ...  
>plot3d(i*j^2+100*normal(20,20),>zscale,scale=[1,1,1.5],angle=-40°,zoom=1.8):
```

```
>i=1:20; j=i'; ...  
>plot3d(i*j^2+100*normal(20,20),>zscale,scale=[1,1,2],angle=-50°,zoom=1.8):
```

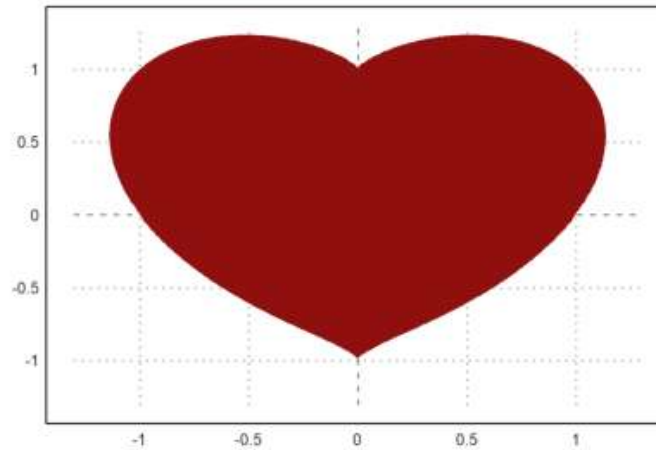


```
>Z=intrandom(5,100,6); v=zeros(5,6); ...  
>loop 1 to 5; v[#]=getmultiplicities(1:6,Z[#]); end; ...  
>columnplot3d(v',scols=1:5,ccols=[1:5]):
```



Permukaan Benda Putar

```
>plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1.3, ...  
>style="#",color=red,<outline, ...  
>level=[-2;0],n=100):
```



```
>ekspresi &= (x^2+y^2-1)^3-x^2*y^3; $ekspresi
```

$$(y^2 + x^2 - 1)^3 - x^2 y^3$$

Kami ingin memutar kurva jantung di sekitar sumbu y. Berikut adalah ungkapan, yang mendefinisikan hati:

$$f(x, y) = (x^2 + y^2 - 1)^3 - x^2 \cdot y^3.$$

Selanjutnya kita atur

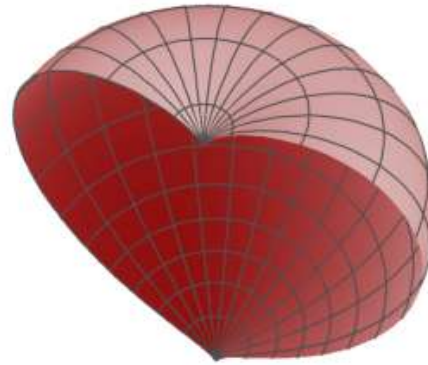
$$x = r \cdot \cos(a), \quad y = r \cdot \sin(a).$$

```
>function fr(r,a) &= ekspresi with [x=r*cos(a),y=r*sin(a)] | trigreduce; $fr(r,a)
```

$$(r^2 - 1)^3 + \frac{(\sin(5a) - \sin(3a) - 2 \sin a) r^5}{16}$$

Hal ini memungkinkan untuk mendefinisikan fungsi numerik, yang memecahkan r, jika a diberikan. Dengan fungsi itu kita dapat memplot jantung yang diputar sebagai permukaan parametrik.

```
>function map f(a) := bisect("fr",0,2;a); ...  
>t=linspace(-pi/2,pi/2,100); r=f(t); ...  
>s=linspace(pi,2pi,100)'; ...  
>plot3d(r*cos(t)*sin(s),r*cos(t)*cos(s),r*sin(t), ...  
>>hue,<frame,color=red,zoom=4,amb=0,max=0.7,grid=12,height=50°):
```

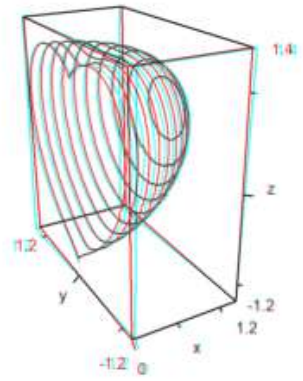


Berikut ini adalah plot 3D dari gambar di atas yang diputar di sekitar sumbu z. Kami mendefinisikan fungsi, yang menggambarkan objek.

```
>function f(x,y,z) ...
```

```
    r=x^2+y^2;  
    return (r+z^2-1)^3-r*z^3;  
endfunction
```

```
>plot3d("f(x,y,z)", ...  
>xmin=0,xmax=1.2,ymin=-1.2,ymax=1.2,zmin=-1.2,zmax=1.4, ...  
>implicit=1,angle=-30°,zoom=2.5,n=[10,60,60],>anaglyph):
```



Plot 3D Khusus

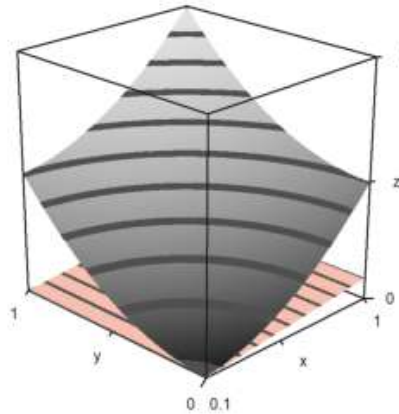
Fungsi `plot3d` bagus untuk dimiliki, tetapi tidak memenuhi semua kebutuhan. Selain rutinitas yang lebih mendasar, dimungkinkan untuk mendapatkan plot berbingkai dari objek apa pun yang Anda sukai.

Meskipun Euler bukan program 3D, ia dapat menggabungkan beberapa objek dasar. Kami mencoba memvisualisasikan paraboloid dan garis singgungnya.

```
>function myplot ...  
  
    y=0:0.01:1; x=(0.1:0.01:1)';  
    plot3d(x,y,0.2*(x-0.1)/2,<scale,<frame,>hue, ..  
        hues=0.5,>contour,color=orange);  
    h=holding(1);  
    plot3d(x,y,(x^2+y^2)/2,<scale,<frame,>contour,>hue);  
    holding(h);  
endfunction
```

Sekarang `framedplot()` menyediakan frame, dan mengatur tampilan.

```
>framedplot("myplot",[0.1,1,0,1,0,1],angle=-45°, ...  
> center=[0,0,-0.7],zoom=6):
```

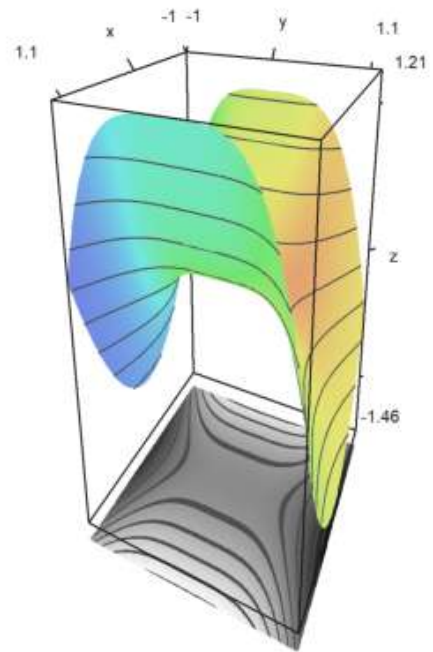



Dengan cara yang sama, Anda dapat memplot bidang kontur secara manual. Perhatikan bahwa `plot3d()` menyetel jendela ke `fullwindow()` secara default, tetapi `plotcontourplane()` mengasumsikan itu.

```
>x=-1:0.02:1.1; y=x'; z=x^2-y^4;
>function myplot (x,y,z) ...
```

```
    zoom(2);
    wi=fullwindow();
    plotcontourplane(x,y,z,level="auto",<scale);
    plot3d(x,y,z,>hue,<scale,>add,color=white,level="thin");
    window(wi);
    reset();
endfunction
```

```
>myplot(x,y,z):
```

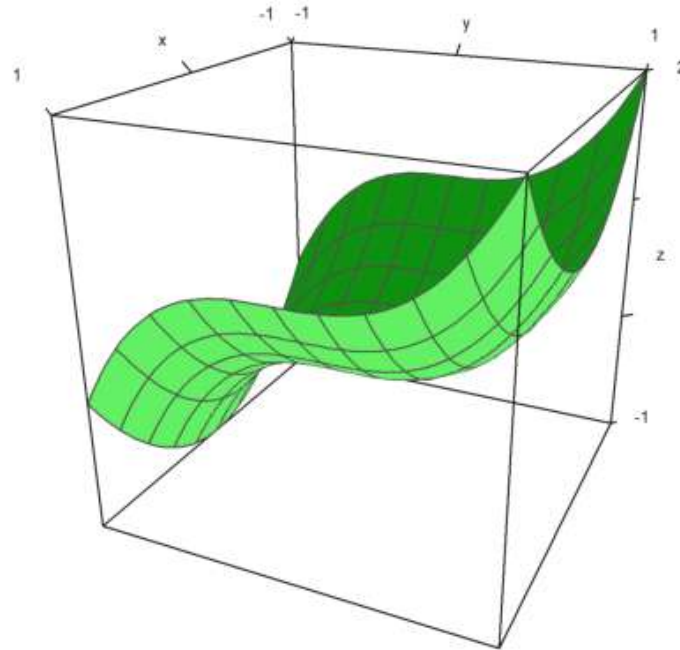


Euler dapat menggunakan frame untuk menghitung animasi terlebih dahulu.

Salah satu fungsi yang memanfaatkan teknik ini adalah `rotate`. Itu dapat mengubah sudut pandang dan menggambar ulang plot 3D. Fungsi memanggil `addpage()` untuk setiap plot baru. Akhirnya itu menjiwai plot.

Silakan pelajari sumber rotasi untuk melihat lebih detail.

```
>function testplot () := plot3d("x^2+y^3"); ...  
>rotate("testplot"); testplot():
```



*Menggambar Povray

Dengan bantuan file Euler povray.e, Euler dapat menghasilkan file Povray. Hasilnya sangat bagus untuk dilihat.

Anda perlu menginstal Povray (32bit atau 64bit) dari <http://www.povray.org/>, dan meletakkan sub-direktori "bin" dari Povray ke jalur lingkungan, atau mengatur variabel "defaultpovray" dengan path lengkap yang menunjuk ke "pvengine.exe".

Antarmuka Povray dari Euler menghasilkan file Povray di direktori home pengguna, dan memanggil Povray untuk mengurai file-file ini. Nama file default adalah `current.pov`, dan direktori default adalah `eulerhome()`, biasanya `c:\Users\Username\Euler`. Povray menghasilkan file PNG, yang dapat dimuat oleh Euler ke dalam buku catatan. Untuk membersihkan file-file ini, gunakan `povclear()`.

Fungsi `pov3d` memiliki semangat yang sama dengan `plot3d`. Ini dapat menghasilkan grafik fungsi $f(x,y)$, atau permukaan dengan koordinat X,Y,Z dalam matriks, termasuk garis level opsional. Fungsi ini memulai raytracer secara otomatis, dan memuat adegan ke dalam notebook Euler.

Selain `pov3d()`, ada banyak fungsi yang menghasilkan objek Povray. Fungsi-fungsi ini mengembalikan string, yang berisi kode Povray untuk objek. Untuk menggunakan fungsi ini, mulai file Povray dengan `povstart()`. Kemudian gunakan `writeln(...)` untuk menulis objek ke file adegan. Terakhir, akhiri file dengan `povend()`. Secara default, raytracer akan dimulai, dan PNG akan dimasukkan ke dalam notebook Euler.

Fungsi objek memiliki parameter yang disebut "look", yang membutuhkan string dengan kode Povray untuk tekstur dan hasil akhir objek. Fungsi `povlook()` dapat digunakan untuk menghasilkan string ini. Ini memiliki parameter untuk warna, transparansi, Phong Shading dll.

Perhatikan bahwa alam semesta Povray memiliki sistem koordinat lain. Antarmuka ini menerjemahkan semua koordinat ke sistem Povray. Jadi Anda dapat terus berpikir dalam sistem koordinat Euler dengan z menunjuk vertikal ke atas, dan x,y,z sumbu dalam arti tangan kanan. Anda perlu memuat file povray.

```
>load povray;
```

Pastikan, direktori bin Povray ada di jalurnya. Jika tidak, edit variabel berikut sehingga berisi path ke povray yang dapat dieksekusi.

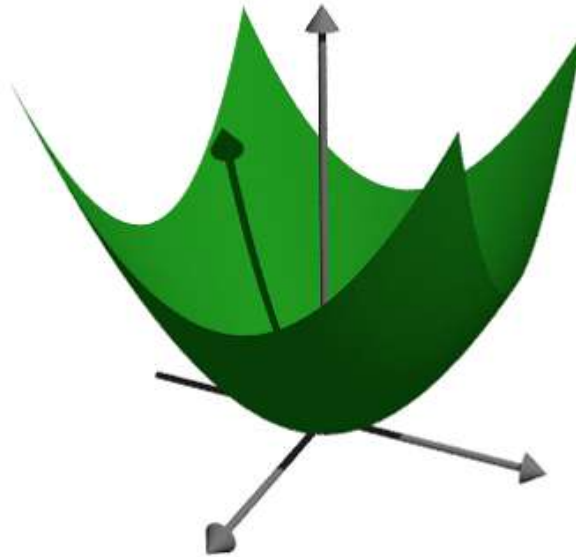
```
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

```
C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe
```

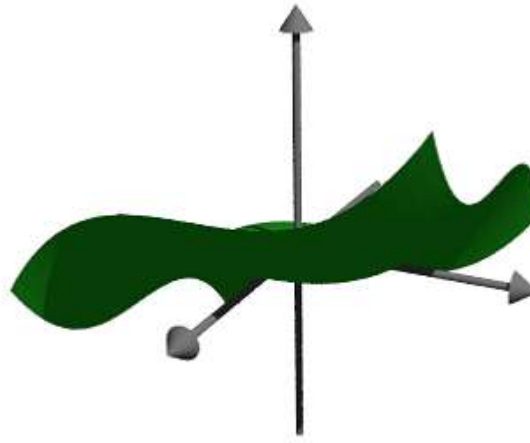
Untuk kesan pertama, kami memplot fungsi sederhana. Perintah berikut menghasilkan file povray di direktori pengguna Anda, dan menjalankan Povray untuk ray tracing file ini.

Jika Anda memulai perintah berikut, GUI Povray akan terbuka, menjalankan file, dan menutup secara otomatis. Karena alasan keamanan, Anda akan ditanya, apakah Anda ingin mengizinkan file exe untuk dijalankan. Anda dapat menekan batal untuk menghentikan pertanyaan lebih lanjut. Anda mungkin harus menekan OK di jendela Povray untuk mengakui dialog awal Povray.

```
>pov3d("x^2+y^2",zoom=3);
```

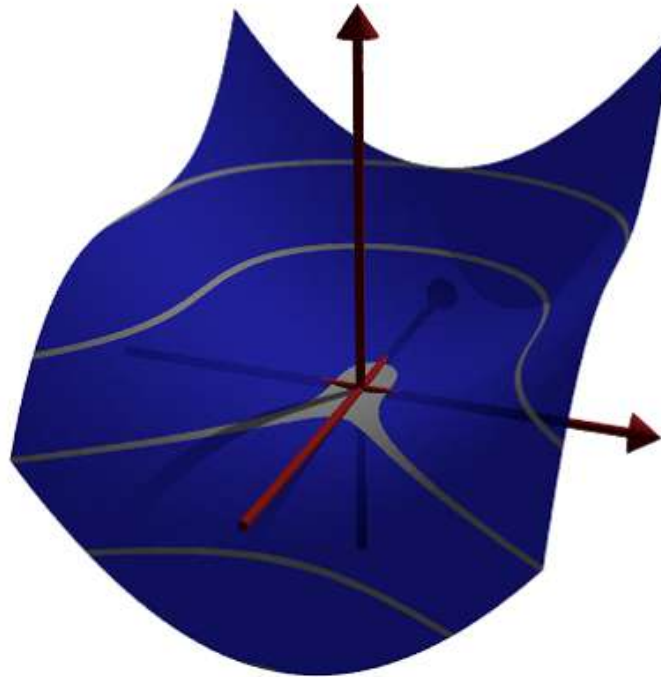


```
>pov3d("x^3+y^3",zoom=3);
```

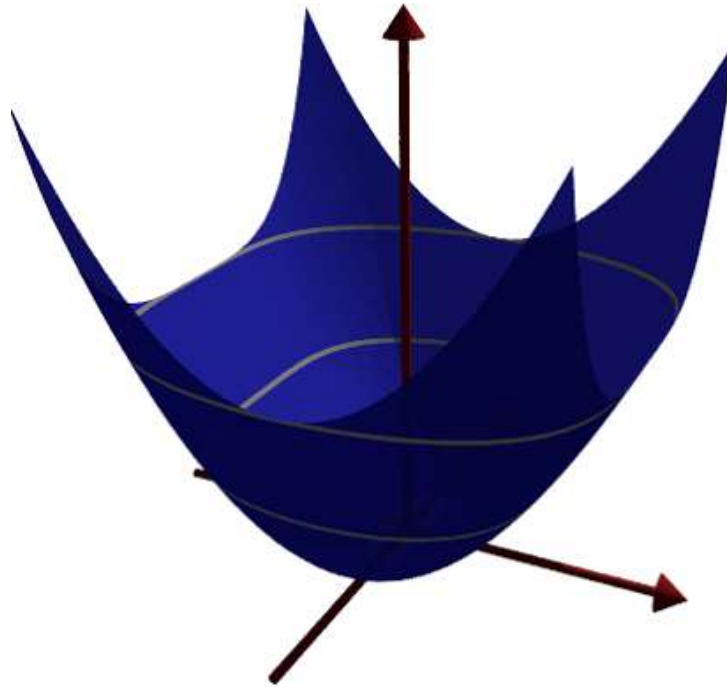


Kita dapat membuat fungsi menjadi transparan dan menambahkan hasil akhir lainnya. Kami juga dapat menambahkan garis level ke plot fungsi.

```
> pov3d("x^2+y^3",axiscolor=red,angle=20°, ...  
> look=povlook(blue,0.2),level=-1:0.5:1,zoom=3.8);
```

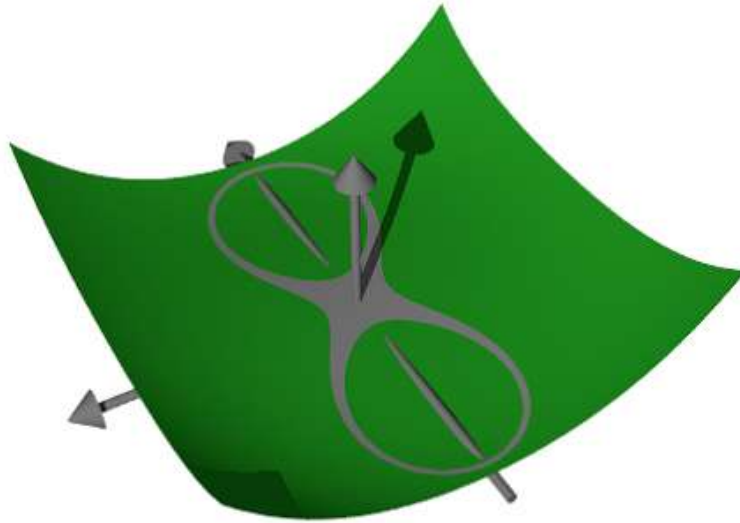
```
>pov3d("x^2+y^4",axiscolor=red,angle=25°, ...  
> look=povlook(blue,0.2),level=-1:0.5:1,zoom=3.8);
```



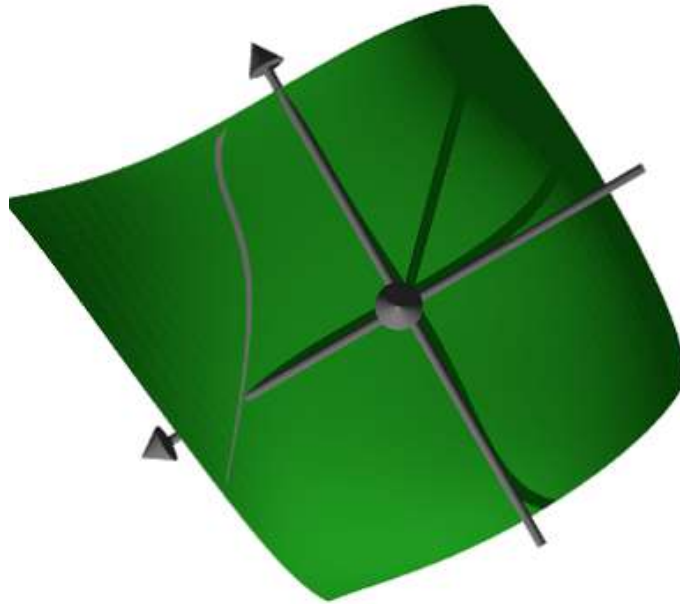
Terkadang perlu untuk mencegah penskalaan fungsi, dan menskalakan fungsi dengan tangan.

Kami memplot himpunan titik di bidang kompleks, di mana produk dari jarak ke 1 dan -1 sama dengan 1.

```
>pov3d("((x-1)^2+y^2)*((x+1)^2+y^2)/40",r=1.5, ...  
> angle=-120°,level=1/40,dlevel=0.005,light=[-1,1,1],height=45°,n=50, ...  
> <fscale,zoom=3.8);
```



```
>pov3d("((x-1)^3+y^3)*((x+1)^2+y^2)/40",r=1.5, ...  
> angle=-120°,level=1/40,dlevel=0.005,light=[-1,1,1],height=90°,n=50, ...  
> <fscale,zoom=3.8);
```



Merencanakan dengan Koordinat

Alih-alih fungsi, kita dapat memplot dengan koordinat. Seperti pada `plot3d`, kita membutuhkan tiga matriks untuk mendefinisikan objek.

Dalam contoh kita memutar fungsi di sekitar sumbu z.

```
>function f(x) := x^3-x+1; ...  
>x=-1:0.01:1; t=linspace(0,2pi,8)'; ...  
>Z=x; X=cos(t)*f(x); Y=sin(t)*f(x); ...  
>pov3d(X,Y,Z,angle=40°,height=20°,axis=0,zoom=4,light=[10,-5,5]);
```



```
>function f(x) := x^4+x+1; ...  
>x=-1:0.01:1; t=linspace(0,2pi,8)'; ...  
>Z=x; X=cos(t)*f(x); Y=sin(t)*f(x); ...  
>pov3d(X,Y,Z,angle=40°,height=20°,axis=0,zoom=4,light=[10,-5,5]);
```



Dalam contoh berikut, kami memplot gelombang teredam. Kami menghasilkan gelombang dengan bahasa matriks Euler.

Kami juga menunjukkan, bagaimana objek tambahan dapat ditambahkan ke adegan pov3d. Untuk pembuatan objek, lihat contoh berikut. Perhatikan bahwa plot3d menskalakan plot, sehingga cocok dengan kubus satuan.

```
>r=linspace(0,1,80); phi=linspace(0,2pi,80)'; ...  
>x=r*cos(phi); y=r*sin(phi); z=exp(-5*r)*cos(8*pi*r)/3; ...  
>pov3d(x,y,z,zoom=5,axis=0,add=povsphere([0,0,0.5],0.1,povlook(green)), ...  
> w=500,h=300);
```



```
>r=linspace(0,1,80); phi=linspace(0,pi,80)'; ...  
>x=r*cos(phi); y=r*sin(phi); z=exp(-5*r)*cos(8*pi*r)/3; ...  
>pov3d(x,y,z,zoom=5,axis=0,add=povsphere([0,0,0.5],0.1,povlook(green)), ...  
> w=500,h=300);
```




Dengan metode bayangan canggih dari Povray, sangat sedikit titik yang dapat menghasilkan permukaan yang sangat halus. Hanya di perbatasan dan dalam bayang-bayang triknya mungkin menjadi jelas.

Untuk ini, kita perlu menambahkan vektor normal di setiap titik matriks.

```
>Z &= x^2*y^3
```

```
2 3  
x y
```

Persamaan permukaannya adalah $[x,y,Z]$. Kami menghitung dua turunan ke x dan y ini dan mengambil produk silang sebagai normal.

```
>dx &= diff([x,y,Z],x); dy &= diff([x,y,Z],y);
```

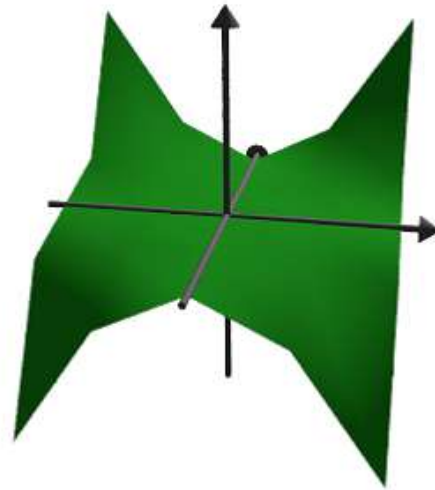
Kami mendefinisikan normal sebagai produk silang dari turunan ini, dan mendefinisikan fungsi koordinat.

```
>N &= crossproduct(dx,dy); NX &= N[1]; NY &= N[2]; NZ &= N[3]; N,
```

$$[- 2 x y^3, - 3 x^2 y^2, 1]$$

Kami hanya menggunakan 25 poin.

```
>x=-1:0.5:1; y=x';  
>pov3d(x,y,Z(x,y),angle=10°, ...  
> xv=NX(x,y),yv=NY(x,y),zv=NZ(x,y),<shadow);
```



Berikut ini adalah simpul Trefoil yang dilakukan oleh A. Busser di Povray. Ada versi yang ditingkatkan dari ini dalam contoh.

Lihat: [Contoh\Trefoil Simpul | Simpul trefoil](#)

Untuk tampilan yang bagus dengan tidak terlalu banyak titik, kami menambahkan vektor normal di sini. Kami menggunakan Maxima untuk menghitung normal bagi kami. Pertama, ketiga fungsi koordinat sebagai ekspresi simbolik.

```
>X &= ((4+sin(3*y))+cos(x))*cos(2*y); ...  
>Y &= ((4+sin(3*y))+cos(x))*sin(2*y); ...  
>Z &= sin(x)+2*cos(3*y);
```

Kemudian kedua vektor turunan ke x dan y.

```
>dx &= diff([X,Y,Z],x); dy &= diff([X,Y,Z],y);
```

Sekarang normal, yang merupakan produk silang dari dua turunan.

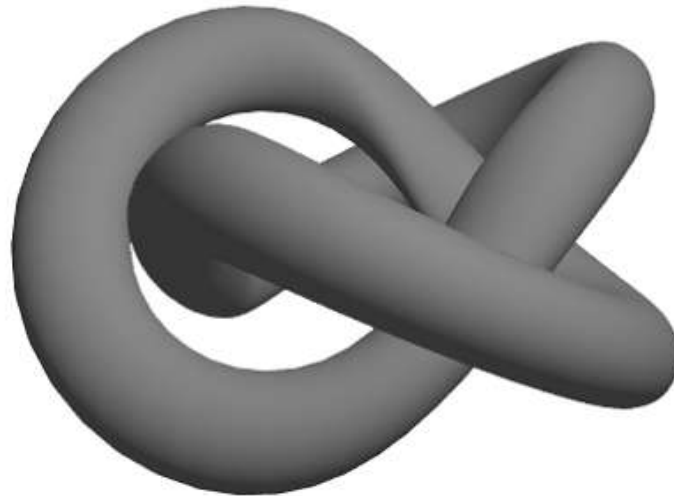
```
>dn &= crossproduct(dx,dy);
```

Kami sekarang mengevaluasi semua ini secara numerik.

```
>x:=linspace(-%pi,%pi,40); y:=linspace(-%pi,%pi,100)';
```

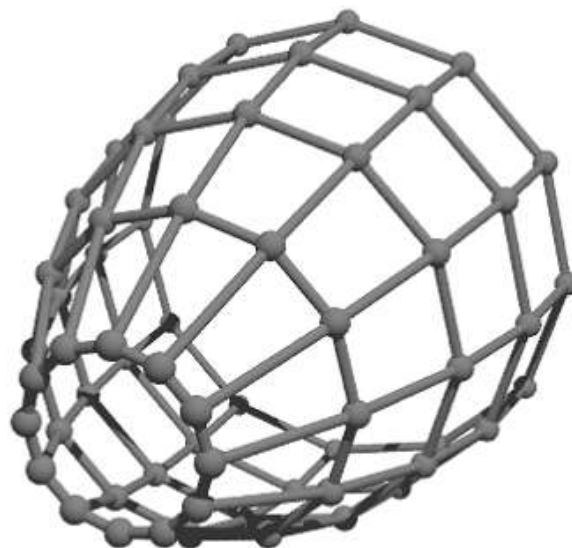
Vektor normal adalah evaluasi dari ekspresi simbolik $dn[i]$ untuk $i=1,2,3$. Sintaks untuk ini adalah `&"expression"(parameters)`. Ini adalah alternatif dari metode pada contoh sebelumnya, di mana kita mendefinisikan ekspresi simbolik NX, NY, NZ terlebih dahulu.

```
>pov3d(X(x,y),Y(x,y),Z(x,y),axis=0,zoom=5,w=450,h=350, ...  
> <shadow,look=povlook(gray), ...  
> xv=&"dn[1]"(x,y), yv=&"dn[2]"(x,y), zv=&"dn[3]"(x,y));
```



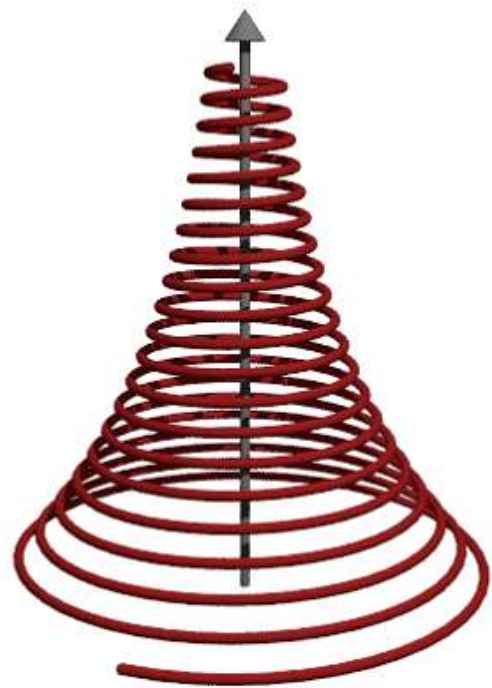
Kami juga dapat menghasilkan grid dalam 3D.

```
>povstart(zoom=4); ...  
>x=-1:0.5:1; r=1-(x+1)^2/6; ...  
>t=(0°:30°:360°)'; y=r*cos(t); z=r*sin(t); ...  
>writeln(povgrid(x,y,z,d=0.02,dballs=0.05)); ...  
>povend();
```



Dengan `povgrid()`, kurva dimungkinkan.

```
>povstart(center=[0,0,1],zoom=3.6); ...  
>t=linspace(0,2,1000); r=exp(-t); ...  
>x=cos(2*pi*10*t)*r; y=sin(2*pi*10*t)*r; z=t; ...  
>writeln(povgrid(x,y,z,povlook(red))); ...  
>writeAxis(0,2,axis=3); ...  
>povend();
```



Objek Povray

Di atas, kami menggunakan pov3d untuk memplot permukaan. Antarmuka povray di Euler juga dapat menghasilkan objek Povray. Objek-objek ini disimpan sebagai string di Euler, dan perlu ditulis ke file Povray.

Kami memulai output dengan povstart().

```
>povstart(zoom=4);
```

Pertama kita mendefinisikan tiga silinder, dan menyimpannya dalam string di Euler.

Fungsi povx() dll. hanya mengembalikan vektor [1,0,0], yang dapat digunakan sebagai gantinya.

```
>c1=povcylinder(-povx,povx,1,povlook(red)); ...  
>c2=povcylinder(-povy,povy,1,povlook(green)); ...  
>c3=povcylinder(-povz,povz,1,povlook(blue)); ...
```

String berisi kode Povray, yang tidak perlu kita pahami pada saat itu.

```
>c1
```

```
cylinder { <-1,0,0>, <1,0,0>, 1  
  texture { pigment { color rgb <0.564706,0.0627451,0.0627451> } }  
  finish { ambient 0.2 }  
}
```

Seperti yang Anda lihat, kami menambahkan tekstur ke objek dalam tiga warna berbeda.

Itu dilakukan oleh `povlook()`, yang mengembalikan string dengan kode Povray yang relevan. Kita dapat menggunakan warna Euler default, atau menentukan warna kita sendiri. Kami juga dapat menambahkan transparansi, atau mengubah cahaya sekitar.

```
>povlook(rgb(0.1,0.2,0.3),0.1,0.5)
```

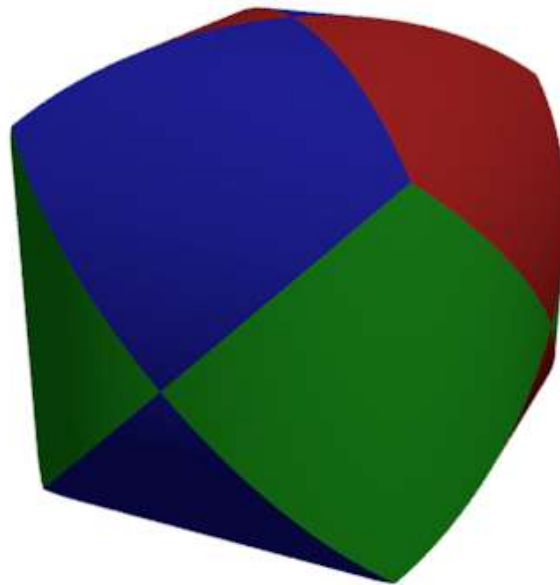
```
texture { pigment { color rgbf <0.101961,0.2,0.301961,0.1> } }  
finish { ambient 0.5 }
```

Sekarang kita mendefinisikan objek persimpangan, dan menulis hasilnya ke file.

```
>writeln(povintersection([c1,c2,c3]));
```

Persimpangan tiga silinder sulit untuk divisualisasikan, jika Anda belum pernah melihatnya sebelumnya.

```
>poend;
```



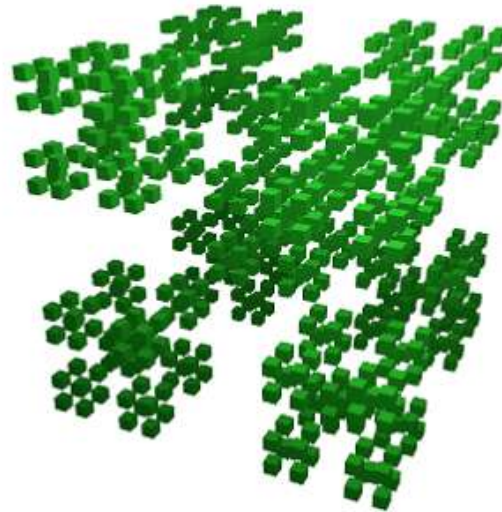
Fungsi berikut menghasilkan fraktal secara rekursif.

Fungsi pertama menunjukkan, bagaimana Euler menangani objek Povray sederhana. Fungsi `povbox()` mengembalikan string, yang berisi koordinat kotak, tekstur, dan hasil akhir.

```
>function onebox(x,y,z,d) := povbox([x,y,z],[x+d,y+d,z+d],povlook());  
>function fractal (x,y,z,h,n) ...
```

```
    if n==1 then writeln(onebox(x,y,z,h));  
    else  
        h=h/3;  
        fractal(x,y,z,h,n-1);  
        fractal(x+2*h,y,z,h,n-1);  
        fractal(x,y+2*h,z,h,n-1);  
        fractal(x,y,z+2*h,h,n-1);  
        fractal(x+2*h,y+2*h,z,h,n-1);  
        fractal(x+2*h,y,z+2*h,h,n-1);  
        fractal(x,y+2*h,z+2*h,h,n-1);  
        fractal(x+2*h,y+2*h,z+2*h,h,n-1);  
        fractal(x+h,y+h,z+h,h,n-1);  
    endif;  
endfunction
```

```
>povstart (fade=10,<shadow);  
>fractal(-1,-1,-1,2,4);  
>povend();
```



Perbedaan memungkinkan memotong satu objek dari yang lain. Seperti persimpangan, ada bagian dari objek CSG Povray.

```
>povstart(light=[5,-5,5],fade=10);
```

Untuk demonstrasi ini, kami mendefinisikan objek di Povray, alih-alih menggunakan string di Euler. Definisi ditulis ke file segera.

Koordinat kotak -1 berarti [-1,-1,-1].

```
>povdefine("mycube",povbox(-1,1));
```

Kita dapat menggunakan objek ini di povobject(), yang mengembalikan string seperti biasa.

```
>c1=povobject("mycube",povlook(red));
```

Kami menghasilkan kubus kedua, dan memutar dan menskalakannya sedikit.

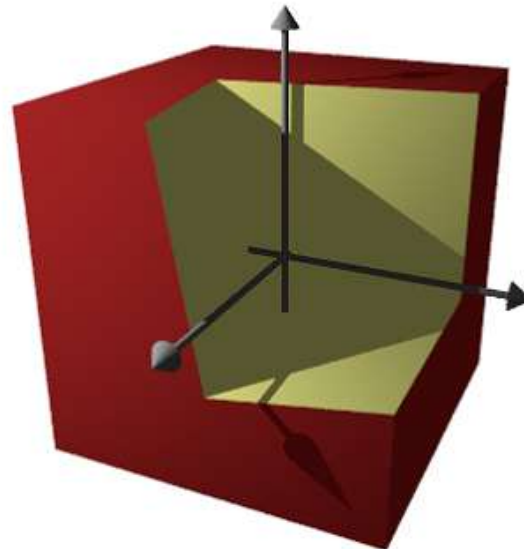
```
>c2=povobject("mycube",povlook(yellow),translate=[1,1,1], ...  
> rotate=xrotate(10°)+yrotate(10°), scale=1.2);
```

Kemudian kita ambil selisih kedua benda tersebut.

```
>writeln(povdifference(c1,c2));
```

Sekarang tambahkan tiga sumbu.

```
>writeAxis(-1.2,1.2,axis=1); ...  
>writeAxis(-1.2,1.2,axis=2); ...  
>writeAxis(-1.2,1.2,axis=4); ...  
>povend();
```



Fungsi Implisit

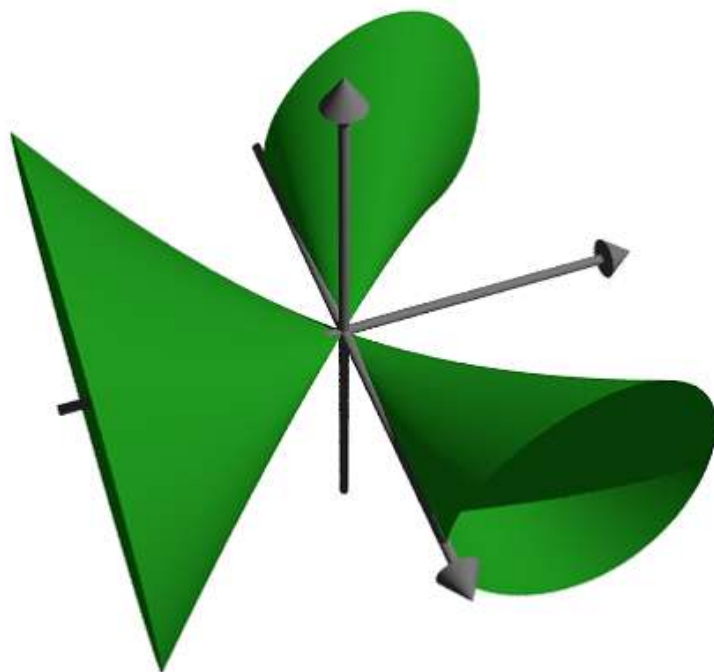
Povray dapat memplot himpunan di mana $f(x,y,z)=0$, seperti parameter implisit di plot3d. Namun, hasilnya terlihat jauh lebih baik.

Sintaks untuk fungsinya sedikit berbeda. Anda tidak dapat menggunakan output dari ekspresi Maxima atau Euler.

```
>povstart(angle=70°,height=50°,zoom=4);
```

Buat permukaan implisit. Perhatikan sintaks yang berbeda dalam ekspresi.

```
>writeln(povsurface("pow(x,2)*y-pow(y,3)-pow(z,2)",povlook(green))); ...  
>writeAxes(); ...  
>povend();
```

Objek Jala

Dalam contoh ini, kami menunjukkan cara membuat objek mesh, dan menggambarinya dengan informasi tambahan.

Kami ingin memaksimalkan xy di bawah kondisi $x+y=1$ dan menunjukkan sentuhan tangensial dari garis level.

```
>povstart (angle=-10°,center=[0.5,0.5,0.5],zoom=7);
```

Kami tidak dapat menyimpan objek dalam string seperti sebelumnya, karena terlalu besar. Jadi kita mendefinisikan objek dalam file Povray menggunakan declare. Fungsi `povtriangle()` melakukan ini secara otomatis. Itu dapat menerima vektor normal seperti `pov3d()`.

Berikut ini mendefinisikan objek mesh, dan langsung menulisnya ke dalam file.

```
>x=0:0.02:1; y=x'; z=x*y; vx=-y; vy=-x; vz=1;  
>mesh=povtriangles(x,y,z,"",vx,vy,vz);
```

Sekarang kita mendefinisikan dua cakram, yang akan berpotongan dengan permukaan.

```
>c1=povdisc([0.5,0.5,0],[1,1,0],2); ...  
>l1=povdisc([0,0,1/4],[0,0,1],2);
```

Tulis permukaan dikurangi dua cakram.

```
>writeln(povdifference(mesh,povunion([c1,l1]),povlook(green)));
```

Tulis dua persimpangan.

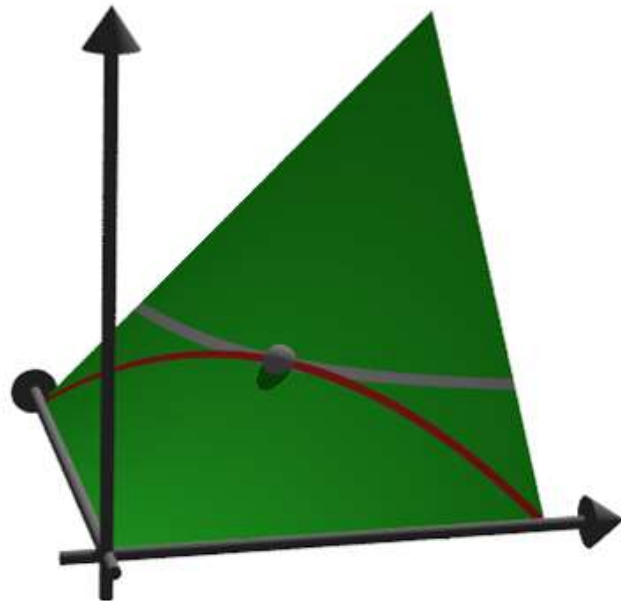
```
>writeln(povintersection([mesh,c1],povlook(red)); ...  
>writeln(povintersection([mesh,l1],povlook(gray)));
```

Tulis titik maksimum.

```
>writeln(povpoint([1/2,1/2,1/4],povlook(gray),size=2*defaultpointsize));
```

Tambahkan sumbu dan selesaikan.

```
>writeAxes(0,1,0,1,0,1,d=0.015); ...  
>povend();
```



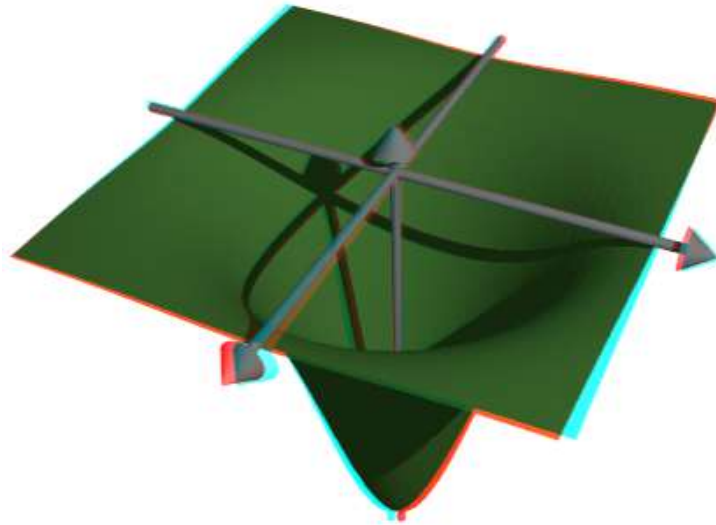
Anaglyph di Povray

Untuk menghasilkan anaglyph untuk kacamata merah/sian, Povray harus berjalan dua kali dari posisi kamera yang berbeda. Ini menghasilkan dua file Povray dan dua file PNG, yang dimuat dengan fungsi `loadanaglyph()`.

Tentu saja, Anda memerlukan kacamata merah/sian untuk melihat contoh berikut dengan benar.

Fungsi `pov3d()` memiliki sakelar sederhana untuk menghasilkan anaglyphs.

```
>pov3d("-exp(-x^2-y^2)/2",r=2,height=45°,>anaglyph, ...  
> center=[0,0,0.5],zoom=3.5);
```



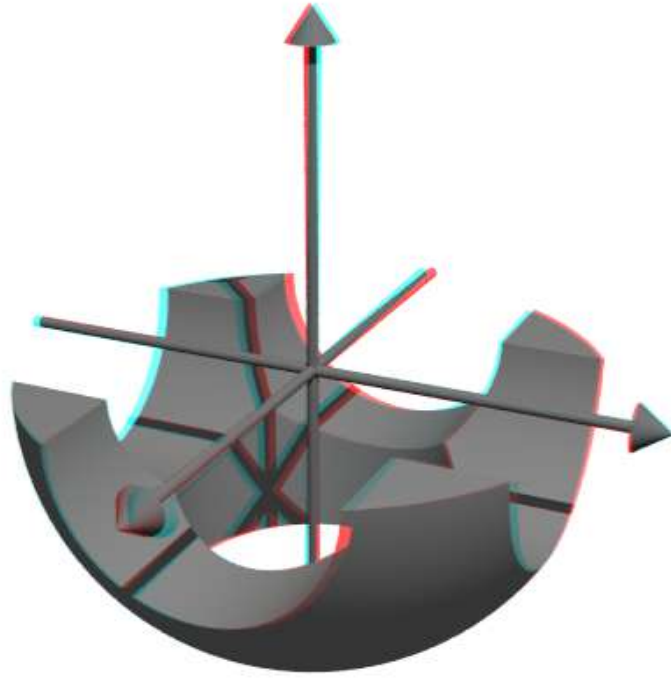
Jika Anda membuat adegan dengan objek, Anda perlu menempatkan generasi adegan ke dalam fungsi, dan menjalankannya dua kali dengan nilai yang berbeda untuk parameter `anaglyph`.

```
>function myscene ...
```

```
s=povsphere(povc,1);  
c1=povcylinder(-povz,povz,0.5);  
clx=povobject(c1,rotate=xrotate(90°));  
cly=povobject(c1,rotate=yrotate(90°));  
c=povbox([-1,-1,0],1);  
un=povunion([c1,clx,cly,c]);  
obj=povdifference(s,un,povlook(red));  
writeln(obj);  
writeAxes();  
endfunction
```

Fungsi `povanaglyph()` melakukan semua ini. Parameternya seperti di `povstart()` dan `povend()` digabungkan.

```
>povanaglyph("myscene",zoom=4.5);
```



Mendefinisikan Objek sendiri

Antarmuka povray Euler berisi banyak objek. Tapi Anda tidak terbatas pada ini. Anda dapat membuat objek sendiri, yang menggabungkan objek lain, atau objek yang sama sekali baru.

Kami mendemonstrasikan sebuah torus. Perintah Povray untuk ini adalah "torus". Jadi kami mengembalikan string dengan perintah ini dan parameternya. Perhatikan bahwa torus selalu berpusat di titik asal.

```
>function povdonat (r1,r2,look="") ...
```

```
    return "torus {"+r1+", "+r2+look+"}";  
endfunction
```

=Inilah torus pertama kami.

```
>t1=povdonat(0.8,0.2)
```

```
torus {0.8,0.2}
```

Mari kita gunakan objek ini untuk membuat torus kedua, diterjemahkan dan diputar.

```
>t2=povobject(t1,rotate=xrotate(90°),translate=[0.8,0,0])
```

```
object { torus {0.8,0.2}  
  rotate 90 *x  
  translate <0.8,0,0>  
}
```

Sekarang kita menempatkan objek-objek ini ke dalam sebuah adegan. Untuk tampilan, kami menggunakan Phong Shading.

```
>povstart(center=[0.4,0,0],angle=0°,zoom=3.8,aspect=1.5); ...  
>writeln(povobject(t1,povlook(green,phong=1))); ...  
>writeln(povobject(t2,povlook(green,phong=1))); ...
```

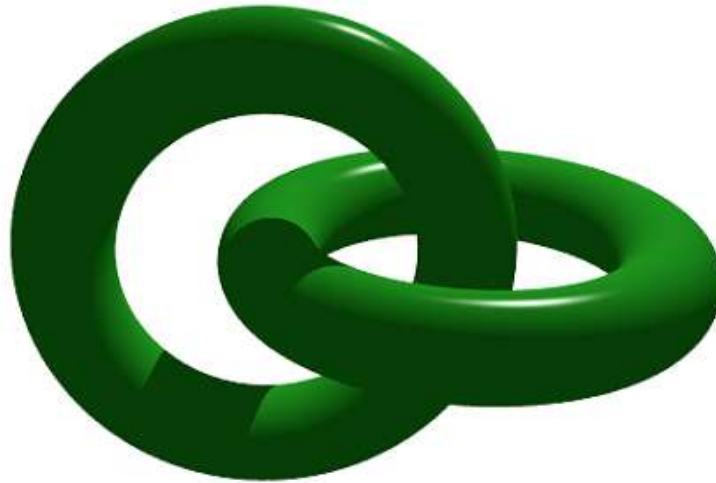
```
>povend();
```

memanggil program Povray. Namun, jika terjadi kesalahan, itu tidak menampilkan kesalahan. Karena itu Anda harus menggunakan

```
>povend(<keluar>;
```

jika ada yang tidak berhasil. Ini akan membiarkan jendela Povray terbuka.

```
>povend(h=320,w=480);
```



Berikut adalah contoh yang lebih rumit. Kami memecahkan

$$Ax \leq b, \quad x \geq 0, \quad c \cdot x \rightarrow \text{Max.}$$

dan menunjukkan titik layak dan optimal dalam plot 3D.

```
>A=[10,8,4;5,6,8;6,3,2;9,5,6];  
>b=[10,10,10,10]';  
>c=[1,1,1];
```

Pertama, mari kita periksa, apakah contoh ini memiliki solusi sama sekali.

```
>x=simplex(A,b,c,>max,>check)'
```

```
[0, 1, 0.5]
```

Ya, sudah.

Selanjutnya kita mendefinisikan dua objek. Yang pertama adalah pesawat

$$a \cdot x \leq b$$

```
>function oneplane (a,b,look="") ...
```

```
    return povplane(a,b,look)  
endfunction
```

Kemudian kita mendefinisikan persimpangan semua setengah ruang dan kubus.

```
>function adm (A, b, r, look="") ...  
  
    ol=[];  
    loop 1 to rows(A); ol=ol|oneplane(A[#],b[#]); end;  
    ol=ol|povbox([0,0,0],[r,r,r]);  
    return povintersection(ol,look);  
endfunction
```

Kita sekarang dapat merencanakan adegannya.

```
>povstart(angle=120°,center=[0.5,0.5,0.5],zoom=3.5); ...  
>writeln(adm(A,b,2,povlook(green,0.4))); ...  
>writeAxes(0,1.3,0,1.6,0,1.5); ...
```

Berikut ini adalah lingkaran di sekitar optimal.

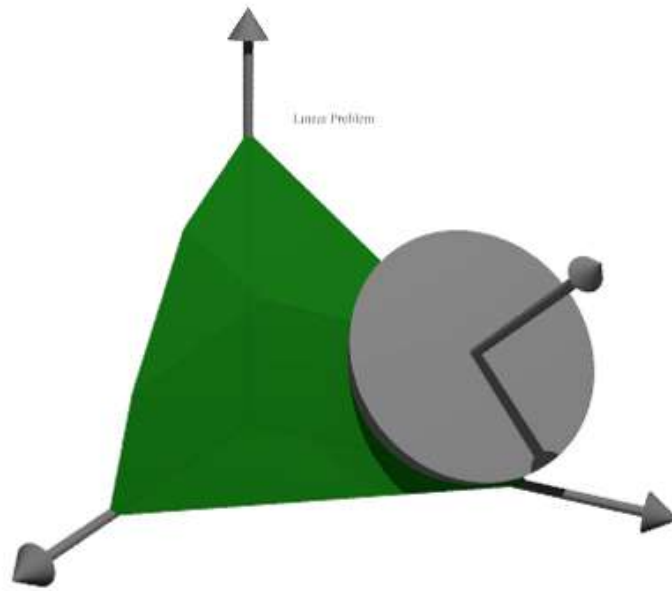
```
>writeln(povintersection([povsphere(x,0.5),povplane(c,c.x')], ...  
> povlook(red,0.9)));
```

Dan kesalahan ke arah yang optimal.

```
>writeln(povarrow(x,c*0.5,povlook(red)));
```

Kami menambahkan teks ke layar. Teks hanyalah objek 3D. Kita perlu menempatkan dan memutarinya menurut pandangan kita.

```
>writeln(povtext("Linear Problem",[0,0.2,1.3],size=0.05,rotate=125°)); ...  
>povend();
```



LATIHAN - LATIHAN SOAL

Fungsi Dua Variabel atau Lebih

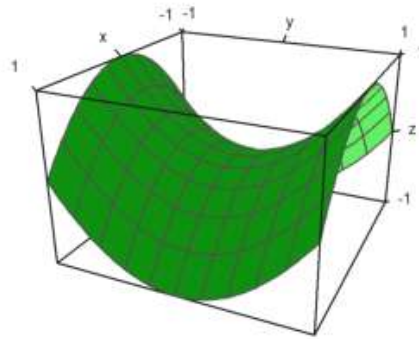
Grafik dari fungsi f dengan dua variabel yang dimaksud adalah grafik

dari persamaan $z = f(x,y)$. Biasanya grafik ini berupa permukaan dan karena setiap (x,y) di daerah asal hanya berpadanan dengan satu nilai z , maka setiap garis tegak lurus bidang- xy memotong permukaan pada paling banyak satu titik.

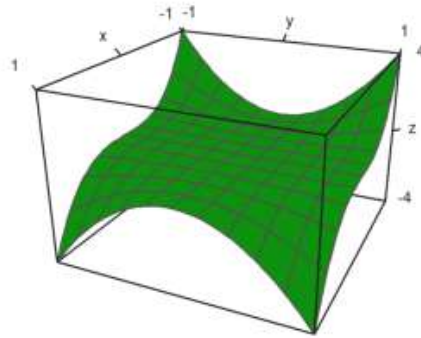
Soal :

1. Grafik merupakan sebuah paraboloida $f(x,y) = y^2 - x^2$
2. $z = -4x^3 - 3y^2$
3. $z = xy \exp(-x^2 - y^2)$
4. $z = x - 1/8x^3 - 1/3y^2$

```
>aspect(1.5); plot3d("y^2-x^2"):
```

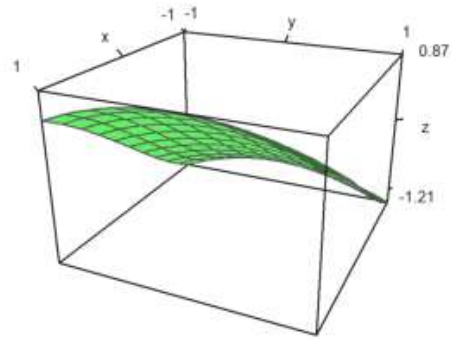
```
>aspect(1.5); plot3d("-4x^3*y^2");
```



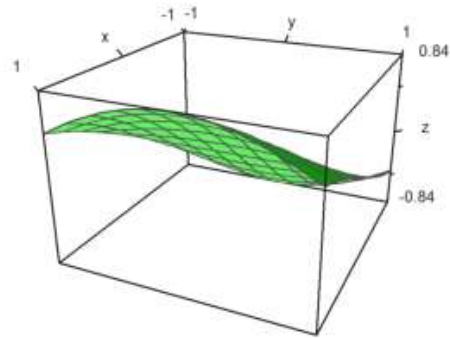
```
>plot3d("x*y*exp(-x^2-y^2)",r=2,<fscale,<scale,distance=13,height=20°, ...  
> center=[0,0,-0.2],frame=3):
```



```
>aspect(1.5); plot3d("x-1/8*x^3-1/3*y^2"):
```



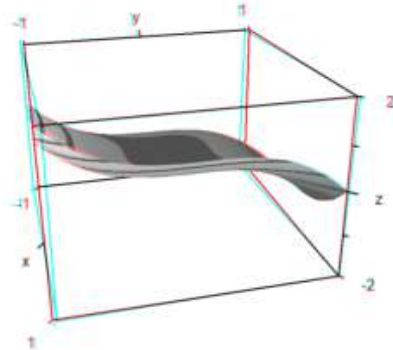
```
>plot3d("sin(x)*cos(y)":
```



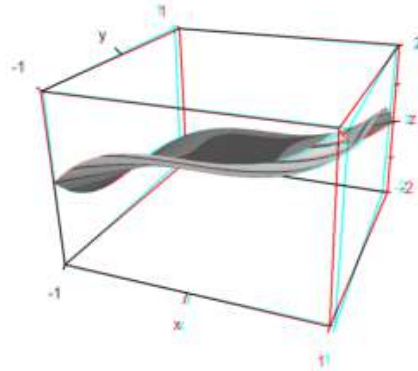
Membuat Talang (Perosotan pada kolam renang)

Dalam membuat perosotan kolam renang ini saya belum bisa membuat secara bagus, karena perosotan yang saya buat masih kurang jadi dan bentuknya pun masih dari arah kiri ke kanan, bahkan terlihat dari sisi belakang dan dari samping, belum terlihat dari depan.

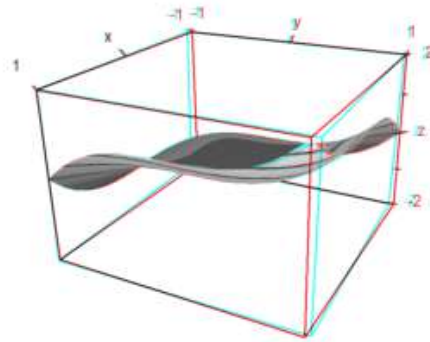
```
>plot3d("x^3-y^3",>anaglyph,>contour,angle=75°):
```



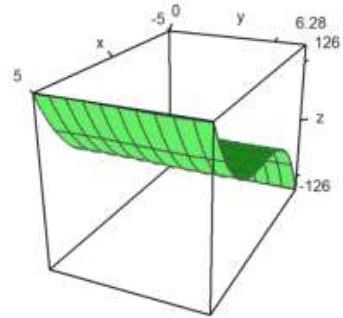
```
>plot3d("x^3-y^3",>anaglyph,>contour,angle=20°):
```



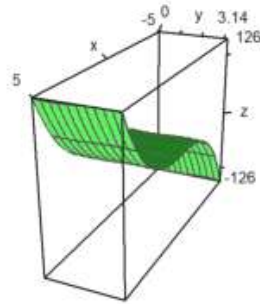
```
>plot3d("x^3+y^3",>anaglyph,>contour,angle=116°):
```



```
>aspect(1.5); plot3d("x^3+sin(y)",-5,5,0,2*pi):
```

```
>aspect(1.5); plot3d("x^3-sin(2y)",-5,5,0,pi):
```



```
>load povray;  
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

```
C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe
```

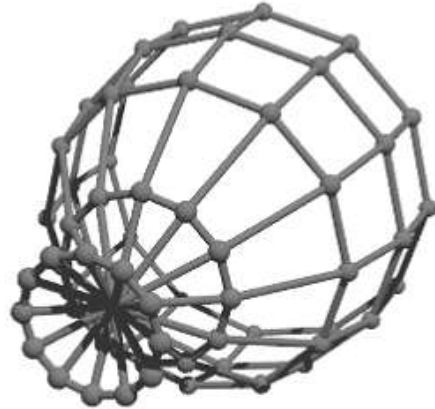
```
>povstart(center=[0,0,1],zoom=3.6); ...  
>t=linspace(0,2,1000); r=exp(-t); ...  
>x=cos(2*pi*20*t)*r; y=sin(2*pi*10*t)*r; z=t; ...  
>writeln(povgrid(x,y,z,povlook(red))); ...  
>writeAxis(0,2,axis=4); ...  
>povend();
```



```
>povstart(center=[0,0,1],zoom=3); ...  
>t=linspace(0,2,1000); r=exp(-t); ...  
>x=cos(pi*10*t)*r; y=sin(pi*10*t)*r; z=t; ...  
>writeln(povgrid(x,y,z,povlook(yellow))); ...  
>writeAxis(0,2,axis=4); ...  
>povend();
```



```
>povstart(zoom=3); ...  
>x=-1:0.5:1; r=1-(x+1)^3/6; ...  
>t=(0°:30°:360°)'; y=r*cos(t); z=r*sin(t); ...  
>writeln(povgrid(x,y,z,d=0.02,dballs=0.05)); ...  
>povend();
```

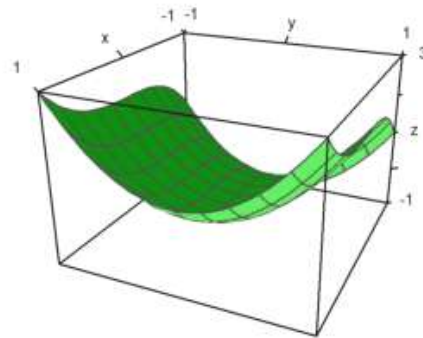


Grafik Fungsi Linear

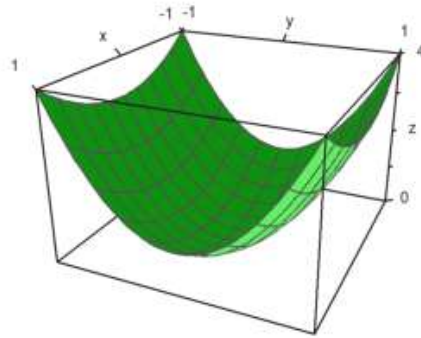
Fungsi linear dua variabel biasanya dinyatakan dalam bentuk

$$f(x, y) = ax + by + c$$

```
>plot3d("x^3+2*y^2"):
```



```
>plot3d("x^2+3*y^2"):
```

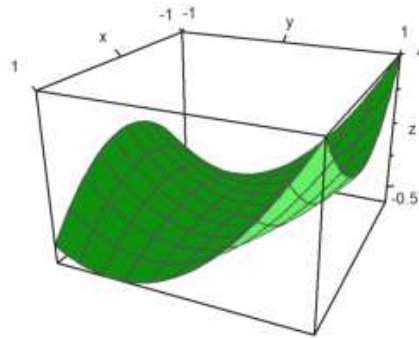


Grafik Fungsi Kuadrat

Fungsi kuadrat dua variabel biasanya dinyatakan dalam bentuk

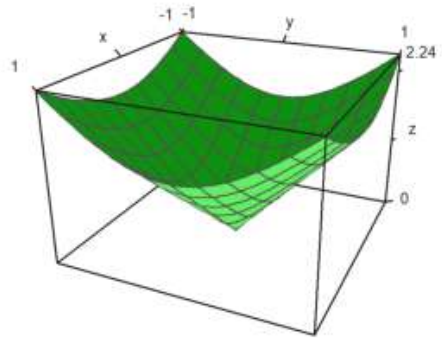
$$f(x, y) = ax^2 + by^2 + cxy + dx + ey + f$$

```
>plot3d("2*x^2*y+2*y^2"):
```



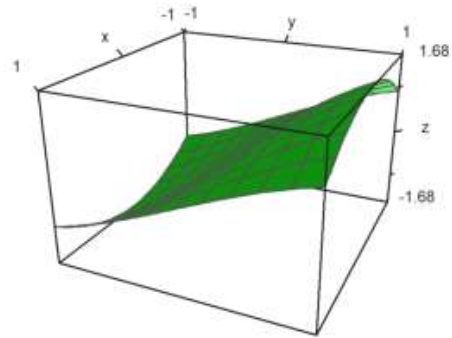
Grafik Fungsi Akar Kuadrat

```
>plot3d("sqrt(2*x^2+3*y^2)"): 
```

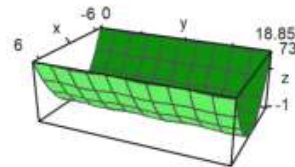



Grafik Fungsi Trigonometri

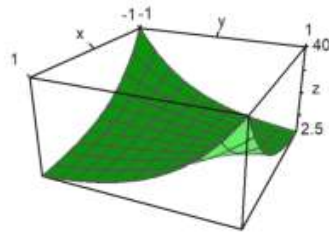
```
>plot3d("2*cos(x)*sin(y)"): 
```



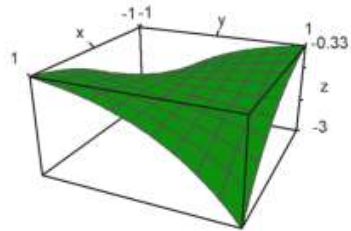
```
>aspect(2); plot3d("2*x^2+sin(y)",-6,6,0,6*pi):
```



```
>plot3d("10*2^(2*x*y)":
```



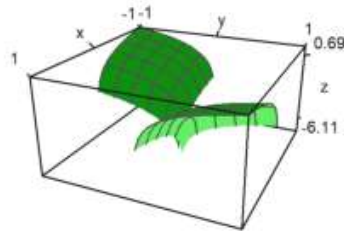
```
>plot3d("-3^(x*y)":
```



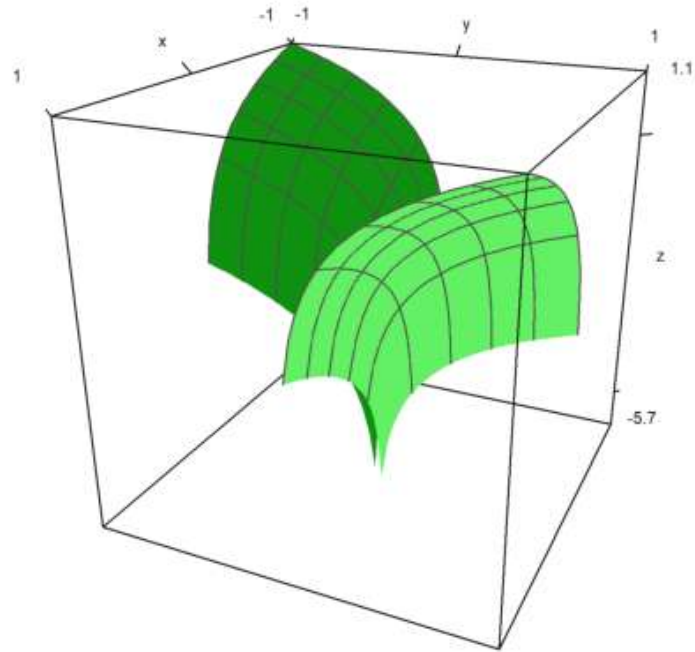
Fungsi logaritma dua variabel bisa dinyatakan sebagai

$$f(x, y) = \log_b(xy)$$

```
>plot3d("log(2*x*y)":
```



```
>plot3d("log(3*x*y)":
```



>

Menggambar Grafik Fungsi Dua Variabel yang Disimpan dalam Variabel

Ekspresi

contoh:

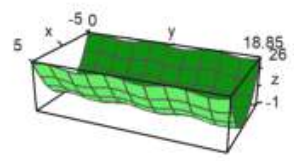
ekspresi dalam string

```
>expr := "x^2+sin(y)"
```

```
x^2+sin(y)
```

plot ekspresi

```
>plot3d(expr,-5,5,0,6*pi):
```



Menggambar Grafik Fungsi Dua Variabel yang Fungsinya Didefinisikan

sebagai Fungsi Numerik

Fungsi Numerik

Fungsi numerik adalah suatu fungsi matematika yang menghasilkan nilai numerik sebagai output-nya. Fungsi ini dapat dinyatakan dalam bentuk persamaan matematika atau algoritma komputasi.

Contoh:

Fungsi

$$f(x, y) = 5x + y$$

Misal input nilai $x=2$ dan $y=3$, maka akan dihasilkan nilai z yaitu

$$z = f(x, y) = 5(2) + 3 = 10 + 3 = 13$$

Gambar Grafik Fungsi

Fungsi satu baris numerik didefinisikan oleh " := " .

Langkah-langkah untuk memvisualisasikan grafik fungsi dua variabel yang fungsinya didefinisikan sebagai fungsi numerik dalam plot3d:

1. Buat fungsi numerik yang akan digunakan untuk memvisualisasikan data.

function f(x,y):=ax+by

dimana a dan b adalah konstanta

2. Gunakan fungsi plot3d() untuk membuat grafik tiga dimensi dari fungsi numerik.

plot3d("f"):

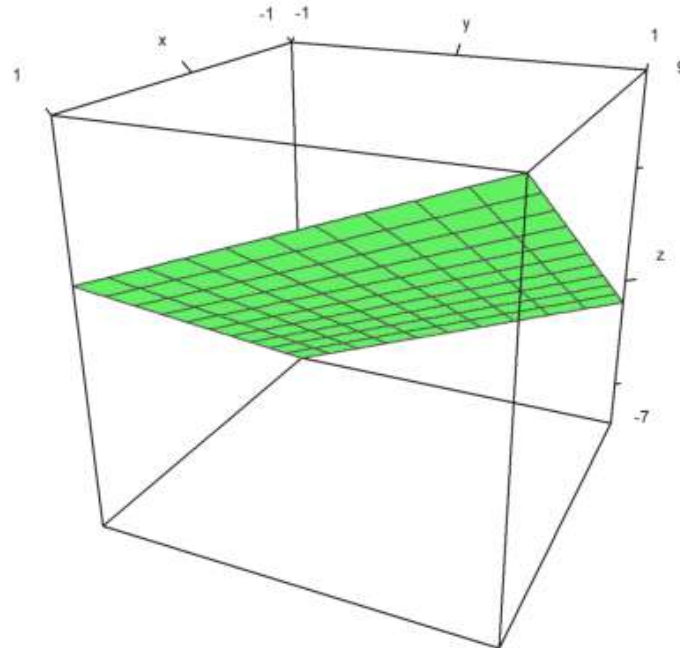
Contoh

Fungsi matematika $f(x,y)$ dapat digambarkan dalam bentuk grafik tiga dimensi menggunakan perintah plot3d. Berikut adalah contoh penggunaan perintah plot3d untuk menggambarkan fungsi tersebut:

1. Fungsi Linear Dua Variabel

$$f(x, y) = 5x + 3y + 1$$

```
>function f(x,y):= 5*x+3*y+1  
>plot3d("f"):
```

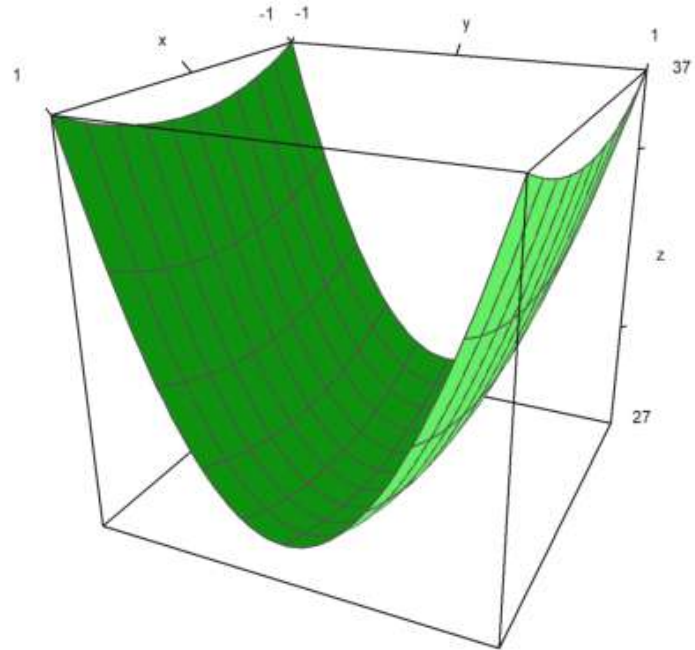


- Fungsi $f(x,y)$ didefinisikan sebagai $5x+3y+1$.
- Perintah `"plot3d('f')"` digunakan untuk memplot grafik 3D dari fungsi $f(x,y)$ menggunakan fungsi `plot3d` di EMT.
- Grafik yang dihasilkan akan menampilkan fungsi dalam tiga dimensi, dengan sumbu x dan y mewakili variabel masukan dan sumbu z mewakili nilai keluaran fungsi. Grafik akan menunjukkan bentuk fungsi dan perubahannya seiring dengan perubahan variabel masukan.

2. Fungsi Kuadrat Dua Variabel

$$f(x, y) = x^2 + 3y^2 + 27$$

```
>function f(x,y):= x^2+(3*y)^2+27  
>plot3d("f"):
```

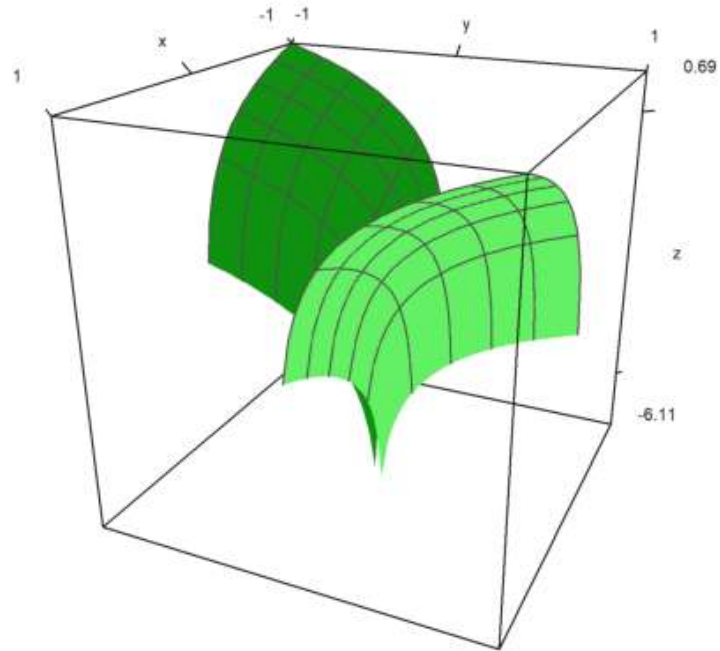


- Perintah "function f(x,y):= x^2+(3*y)^2+27" berarti mendefinisikan fungsi matematika f(x,y) sebagai x pangkat 2 ditambah 3 kali y pangkat 2 ditambah 27.
- Perintah "plot3d("f")" berarti membuat grafik tiga dimensi dari fungsi f(x,y) yang telah didefinisikan sebelumnya.

3. Fungsi Logaritma Dua Variabel

$$f(x, y) = \log(2xy)$$

```
>function f(x,y):= log((2*x)*y)
>plot3d("f"):
```

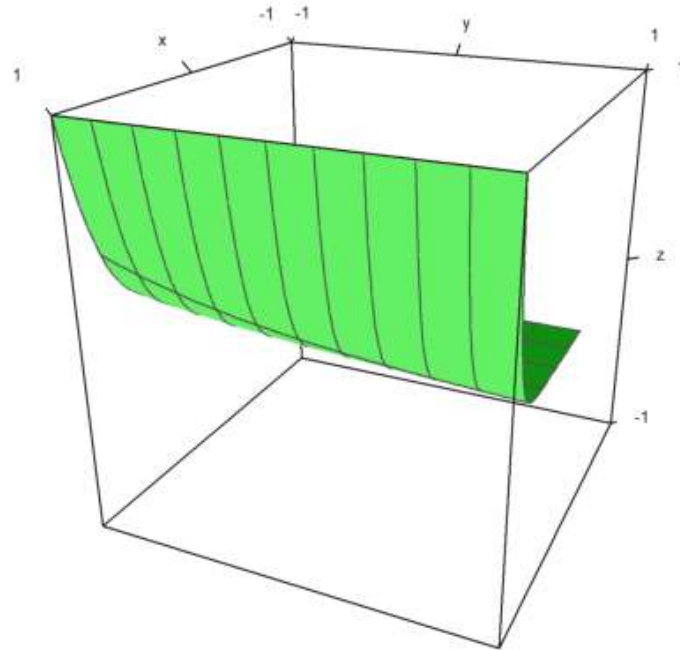


- Input yang diberikan adalah fungsi matematika dua variabel, $f(x,y)$, yang didefinisikan sebagai logaritma hasil kali $2x$ dan y .
- Perintah `"plot3d('f')"` digunakan untuk memplot grafik fungsi $f(x,y)$ dalam ruang tiga dimensi.

4. Fungsi Eksponen Dua Variabel

$$f(x, y) = x^{5y+10}$$

```
>function f(x,y):= x^(5*y+10)  
>plot3d("f"):
```

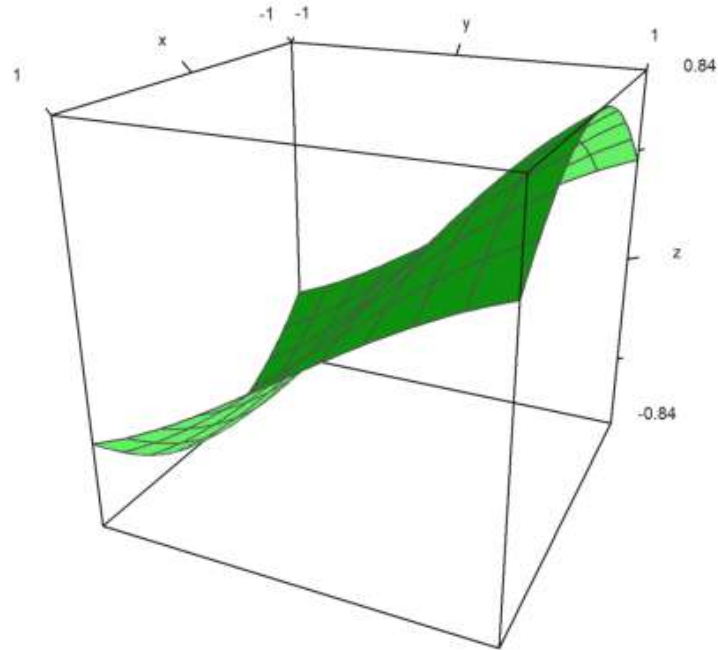


- Perintah 'fungsi $f(x,y):= x^{(5y+10)}$ ' adalah fungsi matematika dua variabel 'x' dan 'y' dan dengan rumus $x^{(5y+10)}$
- Perintah 'plot3d("f")' digunakan untuk memplot fungsi dalam ruang tiga dimensi. Plot yang dihasilkan akan menampilkan nilai fungsi sebagai permukaan pada bidang x-y, dengan tinggi permukaan mewakili nilai fungsi pada titik tersebut.

5. Fungsi Trigonometri Dua Variabel

$$f(x, y) = \cos(x) + \sin(y)$$

```
>function f(x,y):= cos(x)*sin(y)
>plot3d("f"):
```



- Perintah "function f(x,y):= cos(x)*sin(y)" adalah perintah untuk mendefinisikan fungsi matematika f(x,y) yang menghasilkan nilai cosinus dari x dikalikan dengan sinus dari y.
- Perintah "plot3d('f')" adalah perintah untuk membuat grafik tiga dimensi dari fungsi f(x,y) yang telah didefinisikan sebelumnya.

Menggambar Grafik Fungsi Dua Variabel yang Fungsinya Didefinisikan

sebagai Fungsi Simbolik

Fungsi Simbolik

Fungsi simbolik adalah fungsi yang dinyatakan dengan menggunakan simbol-simbol matematika, seperti huruf dan operasi matematika, daripada menggunakan angka konkret atau ekspresi numerik. Fungsi simbolik sering digunakan untuk menggambarkan hubungan antara variabel-variabel matematika dalam bentuk yang lebih umum dan abstrak.

Contoh fungsi simbolik yang umum adalah:

$$g(x, y) = 2x + y$$

Dalam contoh di atas, $g(x)$ adalah fungsi simbolik yang mengaitkan setiap nilai x dengan hasil dari ekspresi matematika $2x + 3$. Fungsi ini dapat digunakan untuk menghitung nilai fungsi untuk berbagai nilai x .

Perbedaan Fungsi Numerik dan Fungsi Simbolik

1. Fungsi Numerik

Fungsi numerik dinyatakan dalam bentuk yang lebih konkret menggunakan angka-angka nyata.

Contoh fungsi numerik adalah

$$g(x, y) = 2x + y + 3$$

dimana kita memberikan nilai numerik kepada "x dan y"

misalnya, $x = 5$ dan $y = 2$, maka hasilnya adalah angka konkret yaitu $g(5,2) = 15$

2. Fungsi Simbolik

Fungsi simbolik dinyatakan menggunakan simbol-simbol matematika seperti huruf (variabel) dan operasi matematika.

Contoh fungsi simbolik adalah

$$g(x, y) = 2x + y + 3$$

"g" adalah simbol fungsi

"x,y" adalah variabel,

$2x + 3$ adalah ekspresi matematika yang menggambarkan hubungan antara "x,y" dan hasil fungsi.

Gambar Grafik Fungsi

Fungsi satu baris simbolik didefinisikan oleh "g=".

Langkah-langkah untuk memvisualisasikan grafik fungsi dua variabel yang fungsinya didefinisikan sebagai fungsi simbolik dalam plot3d:

1. Buat fungsi simbolik yang akan digunakan untuk memvisualisasikan data.

```
function g(x,y)= ax+by;
```

dimana a dan b adalah konstanta

2. Gunakan fungsi plot3d() untuk membuat grafik tiga dimensi dari fungsi simbolik

```
plot3d("g"):
```

3. Menentukan rentang variabel

misal

```
plot3d("g(x,y)",-10,10,-5,5):
```

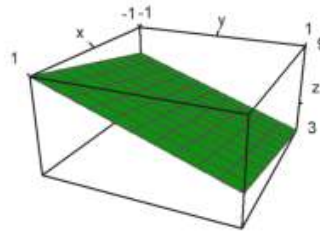
dengan batasan x dari -10 hingga 10 dan batasan y dari -5 hingga 5

Contoh

1. Fungsi Linear Dua Variabel

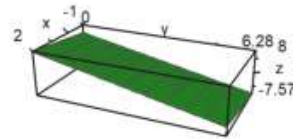
$$g(x, y) = x - 2y + 6$$

```
>function g(x,y)&= x-2*y+6;  
>plot3d("g(x,y)":
```



- Fungsi $g(x,y)$ adalah fungsi matematika yang mengambil dua variabel, x dan y , dan menghasilkan sebuah nilai berdasarkan rumus $x - 2y + 6$.
- Perintah "plot3d" digunakan untuk menghasilkan grafik tiga dimensi dari fungsi tersebut.

```
>plot3d("g(x,y)",-1,2,0,2*pi):
```

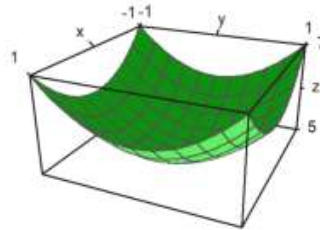


- Perintah "plot3d("g(x,y)",-1,2,0,2*pi)" adalah perintah untuk menggambar grafik fungsi tiga dimensi "g(x,y)" pada rentang x dari -1 hingga 2 dan rentang y dari 0 hingga 2pi.

2. Fungsi Kuadrat Dua Variabel

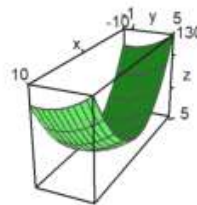
$$g(x, y) = x^2 + y^2 + 5$$

```
>function g(x,y)&= x^2+y^2+5;  
>plot3d("g(x,y)":
```



- Fungsi $g(x,y)$ adalah fungsi matematika yang mengambil dua variabel, x dan y , dan menghasilkan sebuah nilai berdasarkan rumus x^2+y^2+5
- Perintah "plot3d" digunakan untuk menghasilkan grafik tiga dimensi dari fungsi tersebut.

```
>plot3d("g(x,y)",-10,10,-1,5):
```

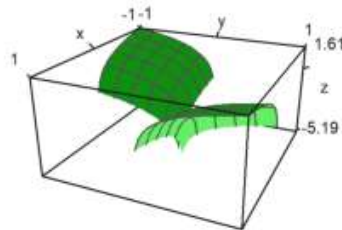


- Perintah "plot3d("g(x,y)",-10,10,-1,5)" adalah perintah untuk menggambar grafik fungsi tiga dimensi g(x,y) pada rentang x dari -10 hingga 10 dan rentang y dari -1 hingga 5

3. Fungsi Logaritma Dua Variabel

$$g(x, y) = \log(xy^5)$$

```
>function g(x,y)= log(x*y*5);  
>plot3d("g(x,y)");
```



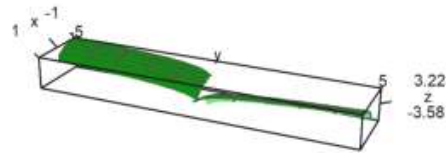
- Fungsi $g(x,y)$ adalah fungsi matematika yang mengambil dua variabel, x dan y , dan menghasilkan sebuah nilai berdasarkan rumus logaritma x dikalikan y dikalikan 5
- Perintah "plot3d" digunakan untuk menghasilkan grafik tiga dimensi dari fungsi tersebut.

```
>plot3d("g(x,y)",-1,1,-5,5):
```



- Perintah "plot3d("g(x,y)",-1,1,-5,5)" adalah perintah untuk menggambar grafik fungsi tiga dimensi $g(x,y)$ pada rentang x dari -1 hingga 1 dan rentang y dari -5 hingga 5

```
>plot3d("g(x,y)",-1,1,-5,5,zoom=4.5):
```

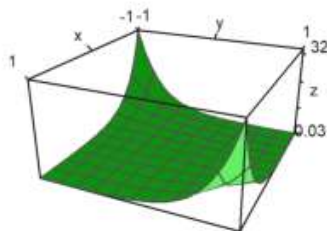


- plot3d: perintah untuk membuat grafik 3D.
 - "g(x,y)": fungsi matematika yang akan digunakan untuk membuat grafik.
 - -1,1: rentang nilai variabel x yang akan digunakan dalam grafik.
 - -5,5: rentang nilai variabel y yang akan digunakan dalam grafik.
 - zoom=4.5: perintah untuk memperbesar tampilan grafik dengan faktor 4.5.
-

4. Fungsi Eksponen Dua Variabel

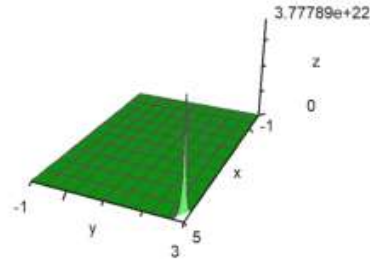
$$g(x, y) = 2^{xy^5}$$

```
>function g(x,y)= 2^(x*y*5);  
>plot3d("g(x,y)");
```



- Fungsi $g(x,y)$ adalah fungsi matematika yang mengambil dua variabel, x dan y , dan menghasilkan sebuah nilai berdasarkan rumus $2^{(xy^5)}$
- Perintah "plot3d" digunakan untuk menghasilkan grafik tiga dimensi dari fungsi tersebut.

```
>plot3d("g(x,y)",-1,5,-1,3,frame=3,zoom=3):
```

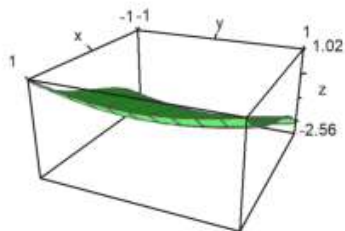


- Perintah `plot3d("g(x,y)",-1,5,-1,3,frame=3,zoom=3)` adalah perintah untuk membuat plot tiga dimensi dari fungsi 'g(x,y)' dengan batas 'x' dari '-1' hingga '5' dan batas 'y' dari '-1' hingga '3'.
 - `plot3d`: perintah untuk membuat plot tiga dimensi.
 - `"g(x,y)"`: fungsi yang akan diplot.
 - `(-1,5)`: batas 'x' dari '-1' hingga '5'.
 - `(-1,3)`: batas 'y' dari '-1' hingga '3'.
 - `frame=3`: menampilkan frame nomor 3.
 - `zoom=3`: memperbesar tampilan plot sebanyak 3 kali.
-

5. Fungsi Trigonometri Dua Variabel

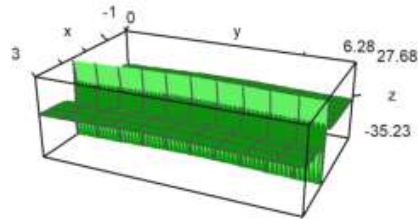
$$g(x, y) = \tan(x) - \cot(y)$$

```
>function g(x,y)&= tan(x)-cos(y);  
>plot3d("g(x,y)");
```



- Fungsi $g(x,y)$ adalah fungsi matematika yang mengambil dua variabel, x dan y , dan menghasilkan sebuah nilai berdasarkan rumus $\tan(x)-\cos(y)$
- Perintah "plot3d" digunakan untuk menghasilkan grafik tiga dimensi dari fungsi tersebut.

```
>plot3d("g(x,y)",-1,3,0,2*pi,frame=1,zoom=3.5):
```

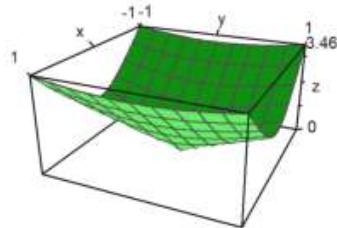


- Perintah `plot3d("g(x,y)",-1,3,0,2*pi,frame=5,zoom=3)` adalah perintah untuk membuat plot tiga dimensi dari fungsi 'g(x,y)' dengan batas 'x' dari '-1' hingga '3' dan batas 'y' dari '0' hingga '2pi'.
 - `plot3d`: perintah untuk membuat plot tiga dimensi.
 - `"g(x,y)"`: fungsi yang akan diplot.
 - `(-1,3)`: batas 'x' dari '-1' hingga '3'.
 - `(0,2pi)`: batas 'y' dari '0' hingga '2pi'.
 - `frame=1`: menampilkan frame nomor 1.
 - `zoom=3.5`: memperbesar tampilan plot sebanyak 3.5 kali.
-

6. Fungsi Akar Kuadrat

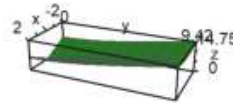
$$P(x, y) = \sqrt{10x^2 + 2y^2}$$

```
>function P(x,y) &= sqrt(10*x^2+2*y^2);  
>plot3d("P(x,y)");
```



- Fungsi $P(x,y)$ adalah fungsi matematika yang mengambil dua variabel, x dan y , dan menghasilkan sebuah nilai berdasarkan rumus akar kuadrat dari $10x^2 + 2y^2$
- Perintah "plot3d" digunakan untuk menghasilkan grafik tiga dimensi dari fungsi tersebut.

```
>plot3d("P(x,y)",-2,2,0,3*pi,frame=5,zoom=2,scale=1):
```

- $P(x,y)$: Merupakan fungsi yang akan digambarkan dalam grafik tiga dimensi.
- $(-2,2)$: Merupakan rentang nilai dari sumbu x yang akan digunakan dalam grafik.
- $(0,3\pi)$: Merupakan rentang nilai dari sumbu y yang akan digunakan dalam grafik. Nilai π dikalikan dengan 3 agar rentang nilai y mencakup tiga putaran lingkaran penuh.
- $\text{frame}=5$: Menentukan nomor bingkai (frame) yang akan digunakan dalam animasi grafik.
- $\text{zoom}=2$: Menentukan faktor pembesaran grafik. Dengan memperbesar tampilan, kita dapat melihat detail yang lebih kecil pada plot.
- $\text{scale}=1$: Menentukan skala grafik. Dengan mengatur skala, kita dapat mengubah jarak antara titik-titik pada sumbu tersebut.

Menggambar Data x, y, z pada ruang Tiga

Dimensi (3D)

Definisi

Menggambar data pada ruang tiga dimensi (3D) adalah proses

visualisasi data yang mengubah informasi dalam tiga dimensi, yaitu panjang, lebar, dan tinggi, menjadi representasi visual yang dapat dipahami dan dianalisis.

Tujuan:

Tujuan dari menggambar data 3D adalah untuk membantu pemahaman dan

interpretasi data yang lebih baik, terutama ketika data tersebut memiliki komponen yang tidak dapat direpresentasikan dengan baik dalam dua dimensi.

Sama seperti plot2d, plot3d menerima data. Untuk objek 3D, Anda perlu menyediakan matriks nilai x-, y- dan z, atau tiga fungsi atau ekspresi $f_x(x,y)$, $f_y(x,y)$, $f_z(x,y)$.

$$\gamma(t, s) = (x(t, s), y(t, s), z(t, s))$$

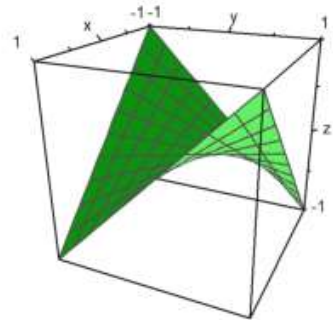
Karena x,y,z adalah matriks, kita asumsikan bahwa (t,s) melalui sebuah kotak persegi. Hasilnya, Anda dapat memplot gambar persegi panjang di ruang angkasa.

Kita dapat menggunakan bahasa matriks Euler untuk menghasilkan koordinat secara efektif.

Dalam contoh berikut, kami menggunakan vektor nilai t dan vektor kolom nilai s untuk membuat parameter permukaan bola. Dalam gambar kita dapat menandai daerah, dalam kasus kita daerah kutub.

Contoh 1

```
>t=-1:0.1:1; s=(-1:0.1:1)'; plot3d(t,s,t*s,grid=10):
```



Baris pertama kode "t=-1:0.1:1" membuat vektor baris t yang berisi nilai dari -1 hingga 1 dengan interval 0.1. Baris kedua "s=(-1:0.1:1)'" membuat vektor kolom s yang berisi nilai dari -1 hingga 1 dengan interval 0.1. Operator transpose ' digunakan untuk mengubah vektor baris t menjadi vektor kolom.

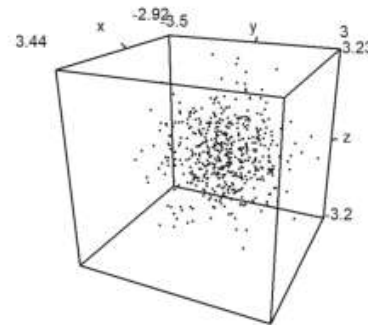
Baris ketiga "plot3d(t,s,ts,grid=10)" membuat plot tiga dimensi dari fungsi $f(x,y) = xy$ pada domain $[-1,1] \times [-1,1]$. Plot dibuat menggunakan fungsi plot3d, yang mengambil tiga argumen: koordinat x, y, dan z dari titik-titik yang akan diplot. Dalam hal ini, koordinat x diberikan oleh vektor t, koordinat y diberikan oleh vektor s, dan koordinat z diberikan oleh hasil perkalian t dan s, yaitu ts. Parameter grid diatur menjadi 10, yang menunjukkan jumlah garis grid yang akan ditampilkan pada plot.

Contoh 2

Tentu saja, titik cloud juga dimungkinkan. Untuk memplot data titik dalam ruang, kita membutuhkan tiga vektor untuk koordinat titik-titik tersebut.

Gayanya sama seperti di plot2d dengan `points=true`;

```
>n=500;...  
>plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style="."):
```



Kode `n=500; plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style='.')` digunakan untuk membuat plot tiga dimensi dari tiga vektor normal yang dihasilkan secara acak dengan menggunakan fungsi `normal()` pada Euler Math Toolbox (EMT). Parameter `n=500` menunjukkan bahwa setiap vektor normal memiliki 500 elemen. Parameter `points=true` digunakan untuk menampilkan titik-titik pada plot, sedangkan parameter `style='.'` digunakan untuk mengatur gaya titik pada plot menjadi titik bulat.

Contoh 3

Dengan lebih banyak usaha, kami dapat menghasilkan banyak permukaan.

Dalam contoh berikut, kita membuat tampilan bayangan dari bola yang

terdistorsi. Koordinat biasa untuk bola adalah

$$\gamma(t, s) = (\cos(t) \cos(s), \sin(t) \sin(s), \cos(s))$$

dengan

$$0 \leq t \leq 2\pi, \quad -\frac{\pi}{2} \leq s \leq \frac{\pi}{2}.$$

Kami mendistorsi ini dengan sebuah faktor

$$d(t, s) = \frac{\cos(4t) + \cos(8s)}{4}$$

```
>t=linspace(0,2pi,320); s=linspace(-pi/2,pi/2,160)';...  
>d=1+0.2*(cos(4*t)+cos(8*s));...  
>plot3d(cos(t)*cos(s)*d,sin(t)*cos(s)*d,sin(s)*d,hue=1,...  
>light=[1,0,1],frame=0,zoom=5):
```



Kode ini terdiri dari beberapa baris. Baris pertama `"t=linspace(0,2pi,320)"` membuat vektor `t` yang berisi 320 nilai yang sama terdistribusi secara merata antara 0 dan 2π . Baris kedua `"s=linspace(-pi/2,pi/2,160)"` membuat vektor `s` yang berisi 160 nilai yang sama terdistribusi secara merata antara $-\pi/2$ dan $\pi/2$. Operator transpose `'` digunakan untuk mengubah vektor baris `s` menjadi vektor kolom.

Baris ketiga `"d=1+0.2*(cos(4t)+cos(8s))"` membuat vektor `d` yang berisi nilai dari $1 + 0.2 * (\cos(4t) + \cos(8s))$. Baris keempat `"plot3d(cos(t)*cos(s)*d,sin(t)*cos(s)*d,sin(s)*d,hue=1,light=[1,0,1],frame=0,zoom=5)"` membuat plot tiga dimensi dari fungsi $f(x,y) = 2x^2 + y^3$. Plot dibuat menggunakan fungsi `plot3d`, yang

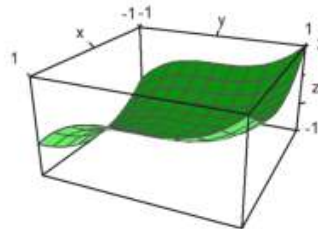
mengambil empat argumen: koordinat x, y, dan z dari titik-titik yang akan diplot, serta beberapa parameter lainnya. Dalam hal ini, koordinat x diberikan oleh ekspresi $\cos(t) \cdot \cos(s) \cdot d$, koordinat y diberikan oleh ekspresi $\sin(t) \cdot \cos(s) \cdot d$, dan koordinat z diberikan oleh ekspresi $\sin(s) \cdot d$. Parameter "hue=1" digunakan untuk mengatur warna pada plot berdasarkan nilai fungsinya. Parameter "light=[1,0,1]" digunakan untuk mengatur pencahayaan pada plot. Parameter "frame=0" digunakan untuk menghilangkan frame pada plot. Parameter "zoom=5" digunakan untuk mengatur level zoom pada plot. **Grafik Tiga**

Dimensi yang bersifat Interaktif dan animasi grafik 3D

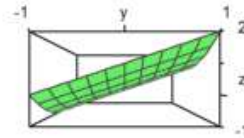
Membuat gambar grafik tiga dimensi (3D) yang bersifat interaktif dan animasi grafik 3D adalah proses menciptakan visualisasi tiga dimensi yang memungkinkan pengguna berinteraksi dengan objek-objek 3D. Interaktivitas dalam gambar 3D memungkinkan pengguna untuk melakukan tindakan seperti mengubah sudut pandang, memindahkan objek, atau berinteraksi dengan elemen-elemen dalam adegan 3D. Animasi grafik 3D dapat mencakup pergerakan, tetapi juga dapat berarti perubahan dalam tampilan atau atribut objek tanpa pergerakan fisik yang mencolok.

CONTOH GAMBAR

```
>function testplot () := plot3d("x^2+y^3"); ...  
>rotate("testplot"); testplot():
```



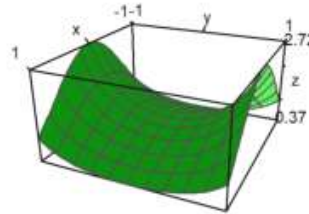

```
>function testplot () := plot3d("x^2+y",distance=3,zoom=1,angle=pi/2,height=0); ...  
>rotate("testplot"); testplot():
```



Hilangkan command angle untuk bisa merotasikan grafik,dan height = 0 untuk membuat posisi sejajar dengan mata jadi tidak mempengaruhi pergerakan hanya berbeda sudut pandang saja

```
>plot3d("exp(-x^2+y^2)",>user, ...  
> title="Turn with the vector keys (press return to finish)":
```

Turn with the vector keys (press return to finish)

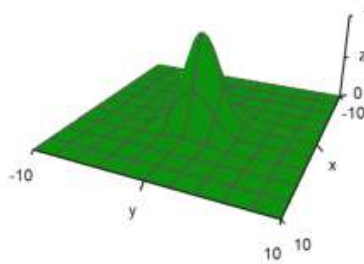


```
>plot3d("exp(x^2+y^2)",>user, ...  
>title="Coba gerakan")
```

Interaksi pengguna dimungkinkan dengan parameter. Pengguna dapat menekan tombol berikut.

1. kiri, kanan, atas, bawah: memutar sudut pandang
2. +,-: memperbesar atau memperkecil
3. a: menghasilkan anaglyph (lihat di bawah)
4. l: beralih memutar sumber cahaya (lihat di bawah)
5. spasi: disetel ulang ke default
6. enter: akhiri interaksi

```
>plot3d("exp(-(x^2+y^2)/5)",r=10,n=80,fscale=4,scale=1.2,frame=3,>user):
```



Parameter "r=10" menunjukkan jari-jari bola yang digunakan untuk membuat plot tiga dimensi. Dalam hal ini, jari-jari bola yang digunakan adalah 10.

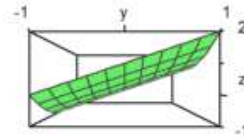
Parameter "n=80" menunjukkan jumlah titik yang digunakan untuk membuat plot. Semakin besar nilai n, semakin banyak titik yang digunakan untuk membuat plot, sehingga plot akan menjadi lebih halus dan akurat.

Parameter "fscale=4" menunjukkan faktor skala pada sumbu z. Dalam hal ini, faktor skala pada sumbu z adalah 4.

Parameter "scale=1.2" menunjukkan faktor skala pada plot. Semakin besar nilai scale, semakin besar ukuran plot yang dihasilkan.

Parameter "frame=3" menunjukkan jenis frame yang digunakan pada plot. Dalam hal ini, jenis frame yang digunakan adalah frame kotak dengan sumbu x, y, dan z yang ditampilkan.

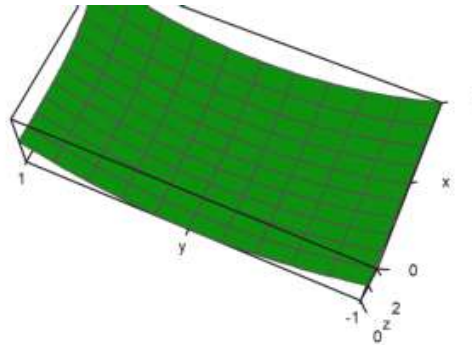
```
>plot3d("x^2+y",distance=3,zoom=1,angle=pi/2,height=0):
```



Tampilan dapat diubah dengan berbagai cara.

- distance: jarak pandang ke plot.
- zoom: nilai zoom.
- angle: sudut terhadap sumbu y negatif dalam radian.
- height: ketinggian tampilan dalam radian.

```
>plot3d("x^4+y^2",a=0,b=1,c=-1,d=1, angle=-20, height=20, ...  
> center=[0.4,0,0], zoom=5):
```



Plot selalu terlihat berada di tengah kubus plot. Anda dapat memindahkan bagian tengah dengan parameter center.

Parameter center digunakan untuk memindahkan pusat plot ke lokasi tertentu dalam ruang. Dalam hal ini, pusat plot diatur ke titik (0.4, 0, 0) dalam ruang tiga dimensi. Parameter center berguna ketika kita ingin mengubah sudut pandang plot atau ketika kita ingin menyelaraskan plot dengan objek lain dalam scene. Dengan menentukan pusat plot, kita dapat mengontrol posisi kamera dan arah tampilan plot.

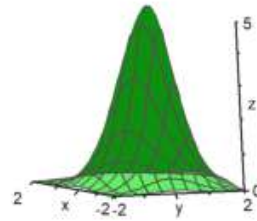
Ada beberapa parameter untuk menskalakan fungsi atau mengubah tampilan grafik.

fscale: menskalakan ke nilai fungsi (defaultnya adalah <fscale>).

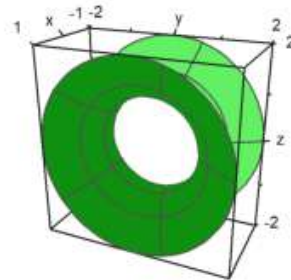
scale: angka atau vektor 1x2 untuk diskalakan ke arah x dan y.

frame: jenis bingkai (default 1).

```
>function testplot () := plot3d("5*exp(-x^2-y^2)",r=2,<fscale>,<scale>,distance=13,height=50, ...
>center=[0,0,-2],frame=3); ...
>rotate("testplot"); testplot():
```



```
>plot3d("x^2+1",a=-1,b=1,rotate=true,grid=5):
```



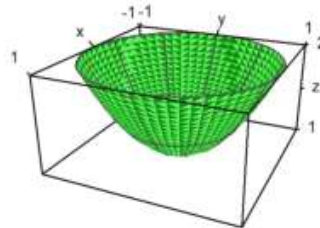
Penjelasan:

Secara umum, parameter "a" dan "b" digunakan untuk menentukan rentang nilai variabel independen dalam suatu fungsi. Dalam kasus ini, "a=-1" dan "b=1" menunjukkan bahwa fungsi tersebut akan diplot pada interval [-1, 1]. Parameter "rotate=true" menunjukkan bahwa grafik akan diputar untuk memberikan tampilan bentuk tiga dimensi yang lebih baik. Parameter "grid=5" menunjukkan bahwa grid dengan jarak 5 unit akan ditampilkan pada grafik.

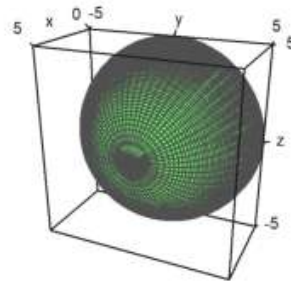
Parameter memutar memutar fungsi dalam x di sekitar sumbu x.

- rotate=1: Menggunakan sumbu x
- rotate=2: Menggunakan sumbu z

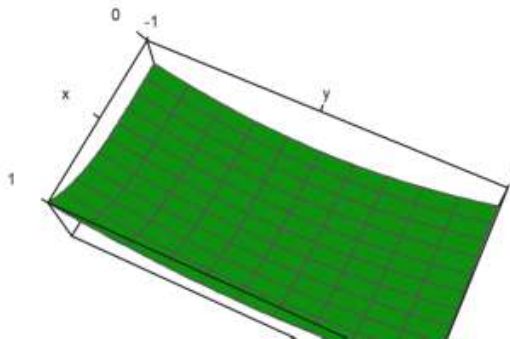
```
>plot3d("x^2+1",a=-1,b=1,rotate=2,grid=5):
```



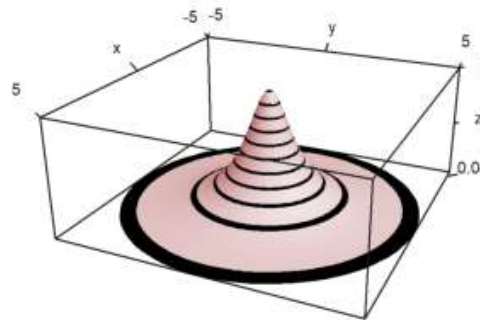
```
>function testplot () := plot3d("sqrt(25-x^2)",a=0,b=5,rotate=1); ...  
>rotate("testplot"); testplot():
```



```
>function testplot () := plot3d("x^4+y^2",a=0,b=1,c=-1,d=1,height=20, ...  
>center=[0.4,0,0],zoom=5); ...  
>rotate("testplot"); testplot():
```




```
>function testplot () := plot3d("1/(x^2+y^2+1)",r=5,>polar, ...  
>fscale=2,>hue,n=100,zoom=4,>contour,color=red); ...  
>rotate("testplot"); testplot():
```



Parameter "r=5" menunjukkan jari-jari bola yang digunakan untuk membuat plot tiga dimensi. Dalam hal ini, jari-jari bola yang digunakan adalah 5.

Parameter ">polar" menunjukkan bahwa plot yang dibuat adalah plot polar tiga dimensi. Plot polar adalah plot yang dibuat dengan menggunakan koordinat polar, yaitu koordinat yang terdiri dari jarak dan sudut.

Parameter "fscale=2" menunjukkan faktor skala pada sumbu z. Dalam hal ini, faktor skala pada sumbu z adalah 2.

Parameter ">hue" menunjukkan bahwa warna pada plot akan diatur berdasarkan nilai fungsinya. Semakin tinggi nilai fungsinya, semakin terang warnanya.

Parameter "n=100" menunjukkan jumlah titik yang digunakan untuk membuat plot. Semakin besar nilai n, semakin banyak titik yang digunakan untuk membuat plot, sehingga plot akan menjadi lebih halus dan akurat.

Parameter "zoom=4" menunjukkan level zoom pada plot.

Parameter ">contour" menunjukkan bahwa garis kontur akan ditampilkan pada plot.

Parameter "color=blue" menunjukkan warna garis kontur pada plot. Dalam hal ini, warna yang digunakan adalah biru.

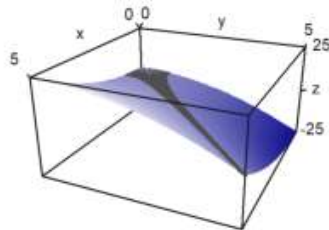
Untuk plotnya, Euler menambahkan garis grid. Sebaliknya dimungkinkan untuk menggunakan garis level dan satu warna atau warna spektral. Euler dapat menggambar ketinggian fungsi pada sebuah plot dengan bayangan. Di semua plot 3D, Euler dapat menghasilkan anaglyph merah/cyan.

-hue: Mengaktifkan bayangan cahaya, bukan kabel.

-contour: Membuat plot garis kontur otomatis pada plot.

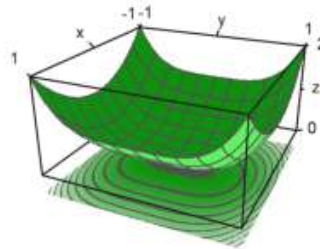
-level=... (atau level): Vektor nilai garis kontur.

```
>function testplot () := plot3d("x^2-y^2",0,5,0,5,level=-1:0.1:1,color=blue); ...  
>rotate("testplot"); testplot():
```



Parameter "level=-1:0.1:1" menunjukkan rentang nilai fungsinya yang akan ditampilkan pada plot. Dalam hal ini, rentang nilai fungsinya adalah dari -1 hingga 1 dengan interval 0.1.

```
>function testplot () := plot3d("x^2+y^4",>cp,cpcolor=green,cpdelta=0.2); ...  
>rotate("testplot"); testplot():
```



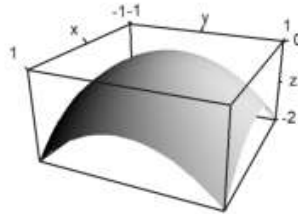
Parameter ">cp" menunjukkan bahwa titik kontrol akan ditambahkan pada plot. Titik kontrol digunakan untuk menentukan bentuk dan posisi plot tiga dimensi.

Parameter "cpcolor=green" menunjukkan warna titik kontrol yang akan digunakan. Dalam hal ini, warna yang digunakan adalah hijau.

Parameter "cpdelta=0.2" menunjukkan jarak antara titik kontrol. Semakin kecil nilai cpdelta, semakin banyak titik kontrol yang akan ditambahkan pada plot.

```
>plot3d("-x^2-y^2", ...  
>hue=true,light=[0,1,1],amb=0,user=true, ...  
> title="Press 1 and cursor keys (return to exit)":
```

Press I and cursor keys (return to exit)



parameter "hue=true" menunjukkan bahwa warna pada plot akan diatur berdasarkan nilai fungsinya. Semakin tinggi nilai fungsinya, semakin terang warnanya.

Parameter "light=light=[0,1,1]" menunjukkan intensitas cahaya pada plot. Nilai light=[0,1,1] menunjukkan bahwa cahaya datang dari arah positif y dan z.

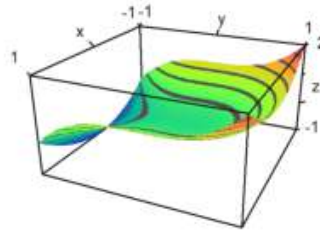
Parameter "amb=0" menunjukkan intensitas cahaya ambient pada plot. Nilai 0 menunjukkan bahwa tidak ada cahaya ambient yang digunakan.

```
>function testplot () := plot3d("-x^2-y^2",color=rgb(0.2,0.2,0),hue=true,frame=false, ...  
> zoom=3,contourcolor=red,level=-2:0.1:1,d1=0.01); ...  
>rotate("testplot"); testplot():
```

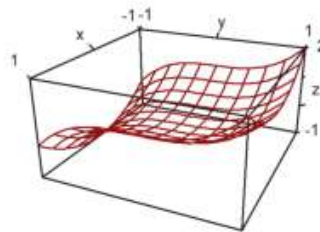


Parameter "frame=false" digunakan untuk menghilangkan frame pada plot tiga dimensi. Parameter "color=rgb(0.2,0.2,0)" menunjukkan warna dasar plot. Dalam hal ini, warna yang digunakan adalah hitam dengan nilai RGB (0.2, 0.2, 0). Parameter "dl=0.01" menunjukkan jarak antara titik-titik pada plot. Semakin kecil nilai dl, semakin banyak titik yang digunakan untuk membuat plot, sehingga plot akan menjadi lebih halus dan akurat. Namun, semakin kecil nilai dl, semakin lama waktu yang dibutuhkan untuk membuat plot.

```
>function testplot () := plot3d("x^2+y^3",>contour,>spectral); ...  
>rotate("testplot"); testplot():
```



```
>function testplot () := plot3d("x^2+y^3", >transparent, grid=10, wirecolor=red); ...  
>rotate("testplot"); testplot():
```

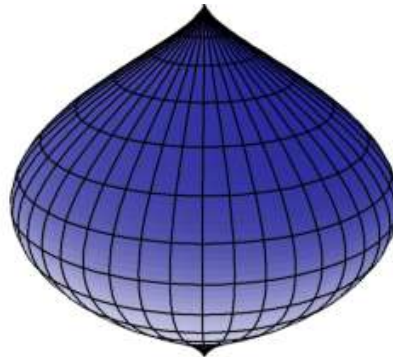


Fungsi Parametrik 3D

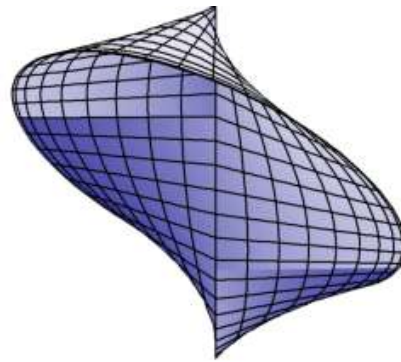
Fungsi parametrik merupakan jenis fungsi matematika yang menggambarkan hubungan antara dua atau lebih variabel, dimana masing-masing koordinat (x, y, z, \dots) dinyatakan sebagai fungsi lain dari beberapa parameter. Fungsi parametrik dapat digunakan untuk menggambar kurva, lintasan, atau hubungan antara berbagai variabel yang bergantung pada parameter-parameter tertentu.

Sebagai contoh :

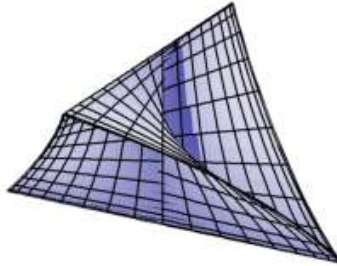
```
>plot3d("cos(x)*cos(y)^3","sin(x)*cos(y)^3","sin(y)", a=0,b=2*pi,c=pi/2,d=-pi/2,...  
>>hue,color=blue,light=[0,1,3],<frame,...  
>n=90,grid=[20,50],wirecolor=black,zoom=5):
```



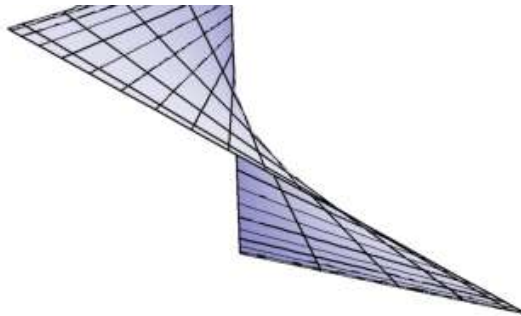
```
>plot3d("cos(x)*cos(y)", "sin(x)*cos(y)", "cos(x)", a=0, b=2*pi, c=pi/2, d=-pi/2, ...  
>>hue, color=blue, light=[0,1,3], <frame, ...  
>n=90, grid=[20,50], wirecolor=black, zoom=5):
```



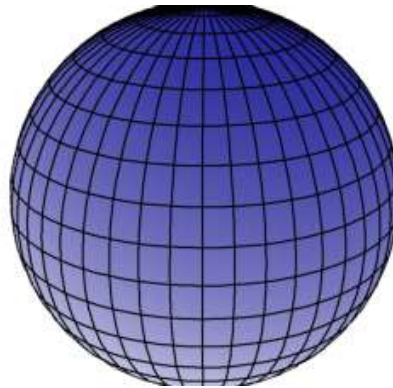
```
>plot3d("cos(x)^3*sin(y)", "sin(x)^2*sin(y)", "cos(x)^2", a=0, b=2*pi, c=pi/2, d=-pi/2, ...  
>>hue, color=blue, light=[0,1,5], <frame, ...  
>n=90, grid=[20,50], wirecolor=black, zoom=5):
```

```
>plot3d("cos(x)^2*cos(y)","sin(x)^2*cos(y)","cos(x)^2", a=0,b=2*pi,c=pi/2,d=-pi/2,...  
>>hue,color=blue,light=[0,1,5],<frame,...  
>n=90,grid=[10,50],wirecolor=black,zoom=5):
```



```
>plot3d("cos(x)*cos(y)", "sin(x)*cos(y)", "sin(y)", a=0, b=2*pi, c=pi/2, d=-pi/2, ...  
>>hue, color=blue, light=[0,1,3], <frame, ...  
>n=90, grid=[20,50], wirecolor=black, zoom=5):
```



Menggambar Fungsi Implisit Implisit

Fungsi implisit (implicit function) adalah fungsi yang memuat lebih dari satu variabel, berjenis variabel bebas dan variabel terikat yang berada dalam satu ruas sehingga tidak bisa dipisahkan pada ruas yang berbeda.

$$F(x, y, z) = 0$$

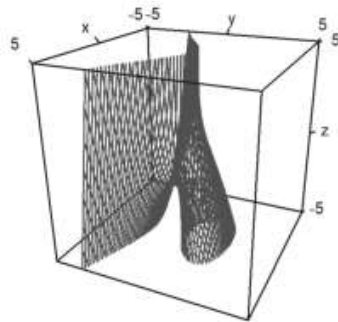
(1 persamaan dan 3 variabel), terdiri dari 2 variabel bebas dan 1 terikat

Misalnya,

$$F(x, y, z) = x^2 + y^2 + z^2 = 1$$

adalah persamaan implisit yang menggambarkan bola dengan jari-jari 1 dan pusat di (0,0,0).

```
>plot3d("x^2+y^3+z*y-1", r=5, implicit=3):
```

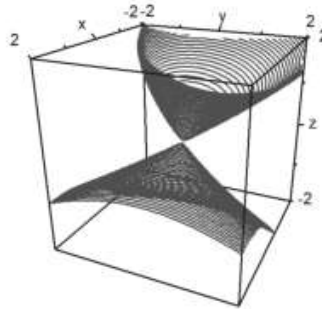


```
>c=1; d=1;
```

```
>plot3d("((x^2+y^2-c^2)^2+(z^2-1)^2)*((y^2+z^2-c^2)^2+(x^2-1)^2)*((z^2+x^2-c^2)^2+(y^2-1)^2)-d", r=2
```



```
>plot3d("x^2+y^2+4*x*z+z^3",>implicit, r=2, zoom=2.5):
```



Selain plot kontur yang sudah di jelaskan sebelumnya, pada EMT juga ada plot umplisit dalam tiga dimensi. Euler menghasilkan potongan melalui objek. Fitur plot3d termasuk plot implisit. Plot-plot ini menunjukkan himpunan nol dari sebuah fungsi dalam tiga variabel.

Solusi dari

$$f(x, y, z) = 0$$

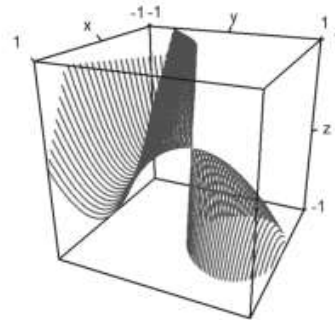
dapat divisualisasikan dalam potongan yang sejajar dengan bidang x-y, bidang x-z, dan bidang y-z.

- implicit = 1: potong sejajar dengan bidang-y-z
- implicit = 2: memotong sejajar dengan bidang x-z
- implicit=4: memotong sejajar dengan bidang x-y

Ambil contoh dari persamaan latex pada fungsi implisit tadi dan tambahkan nilai-nilai ini, sehingga kita dapat memplot persamaan ini

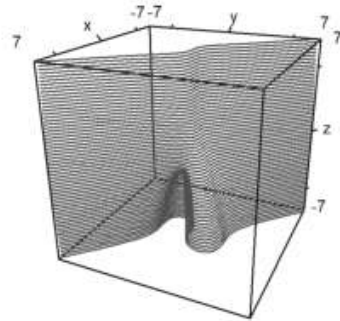
$$M = (x, y, z) : x^2 + y^3 + zy = 1$$

```
>plot3d("x^2+y^3+z*y", r=1, implicit=2):
```

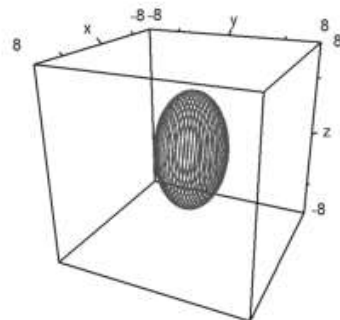


Contoh fungsi implisit yang lainnya

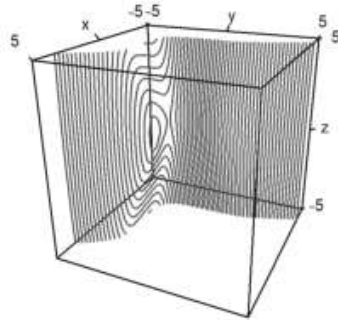
```
>plot3d("x^3+y^3+z*y-1",r=7,implicit=4):
```



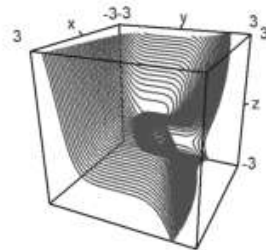
```
>plot3d("2*x^2 + 3*y^2 + z^2 - 25",r=8,implicit=2):
```



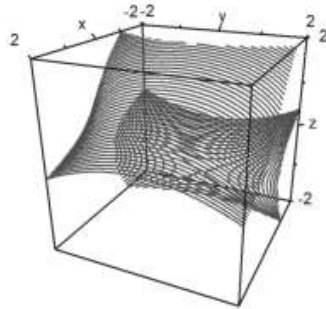
```
>plot3d("x^5 + 5*y^3 + 3*z^2 - 5*x - 7*y - 5*z + 10",r=5,implicit=2):
```



```
>plot3d("x^3+y^5+5*x*z+z^3",>implicit,r=3,zoom=2):
```



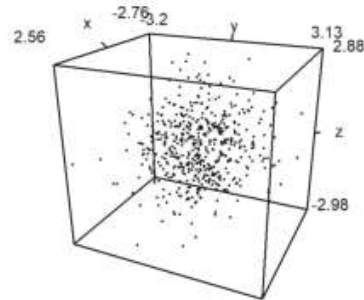

```
>plot3d("x^2+y^2+4*x*z+z^3-5",>implicit,r=2,zoom=2.5):
```



Menggambar Titik pada Ruang Tiga Dimensi

Untuk menggambar titik pada ruang tiga dimensi kita memerlukan tiga vektor untuk koordinat titik serta menambahkan parameter `points=true`.

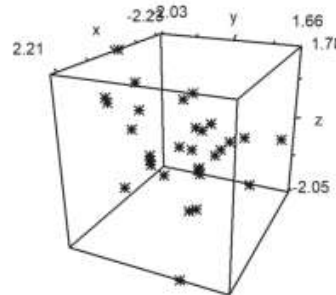
```
>n=510; ...  
>plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style="."):
```



```
>n=510; ...  
>plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style="."):
```

- plot3d() untuk menjalankan perintah membuat plot 3D.
- normal(1,n) sebagai titik koordinat yang akan diplot pada sumbu x,y,z dengan nilai angka distribusi normal yang dicetak secara random sebanyak n sehingga membentuk matriks 1xn atau 1x500.
- points=true sebagai parameter yang memerintahkan plot3d akan menampilkan points(titik-titik).

```
>n=30; ...  
>plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style="*"):
```



1. $n = 30$; Ini adalah pernyataan untuk menginisialisasi variabel n dengan nilai 30. Variabel ini kemungkinan akan digunakan untuk menentukan jumlah titik yang akan digunakan dalam plot.
2. $\text{normal}(1, n)$: Fungsi normal digunakan untuk menghasilkan nilai-nilai acak yang terdistribusi secara normal (disebut juga Gaussian) dengan rata-rata 1 dan deviasi standar 1. Ini berarti Anda akan mendapatkan n nilai acak yang terdistribusi secara normal dengan rata-rata 1 dan deviasi standar 1. Anda melakukan ini untuk mendapatkan koordinat x , y , dan z untuk plot 3D.

3. `plot3d(...)`: Ini adalah fungsi yang digunakan untuk membuat plot 3D dengan parameter-parameter berikut:

`-normal(1, n)`: Ini adalah data koordinat x , y , dan z yang telah

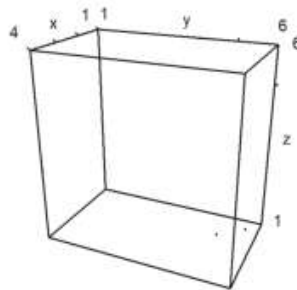
dihasilkan sebelumnya.

`-points = true`: Ini mengatur agar titik-titik data ditampilkan dalam

plot.

`-style = "*"`: Ini mengatur gaya plot menjadi tanda bintang (*).

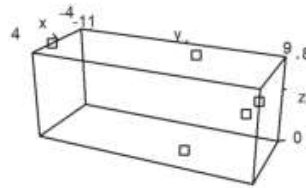
```
>x=[1,2,3,4]; y=[4,5,6,1]; z=[6,1,2,3];  
>plot3d(x,y,z,points=true,style="."):
```



```
>x=[1,2,3,4]; y=[4,5,6,1]; z=[6,1,2,3];  
>plot3d(x,y,z,points=true,style="."):
```

- plot3d() untuk menjalankan perintah membuat plot 3D.
- Titik koordinat yang akan diplot pada sumbu x,y,z telah didefinisikan oleh vektor baris x,y,z sebelumnya.
- points=true sebagai parameter yang memerintahkan plot3d akan menampilkan points(titik-titik).

```
>x=[1,1,0,-4,4]; y=[2,-11,7,1,9]; z=[0,8,4,7,7];  
>plot3d(x,y,z,points=true,zoom=3,style="/"):
```



1.x, y, dan z: Ini adalah tiga vektor yang berisi koordinat titik-titik dalam tiga dimensi. Dalam contoh ini, x berisi [1, 1, 0, -4, 4], y berisi [2, -11, 7, 1, 9], dan z berisi [0, 8, 4, 7, 7]. Setiap elemen dalam vektor-vektor ini mewakili koordinat satu titik dalam ruang 3D.

2.plot3d(...): Ini adalah fungsi yang digunakan untuk membuat plot 3D dengan parameter-parameter berikut:

-x, y, dan z: Ini adalah data koordinat yang akan digunakan untuk membuat plot.

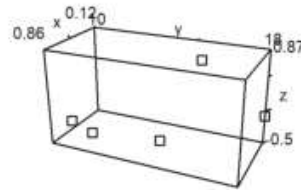
-points = true: Ini mengatur agar titik-titik data ditampilkan dalam plot. Dengan points = true, Anda

akan melihat titik-titik yang mewakili koordinat data.

-zoom = 3: Ini mengatur tingkat zoom plot. Dengan zoom = 3, plot akan diperbesar sebanyak tiga kali dari ukuran defaultnya.

-style = "/" : Ini mengatur gaya plot menjadi tanda garis miring ("/").

```
>a=random(1,5); b=linspace(10,18,4); c=normal(1,5); ...  
>plot3d(a,b,c,scale=[5,1,3],points=true,style="''):
```



```
>a=random(1,5); b=linspace(10,18,4); c=normal(1,5); ...
```

```
plot3d(a,b,c,scale=[5,1,3],points=true,style="''):
```

- `plot3d()` untuk menjalankan perintah membuat plot 3D.
- Titik koordinat yang akan diplot pada sumbu x,y,z didefinisikan oleh `x=a=random(1,5)` yaitu bilangan acak dari 0-1 sebanyak 5 bilangan, `y=b=linspace(10,18,4)` yaitu bilangan dari 10 hingga 18 dengan selisih yang sama sebanyak 5 bilangan, `z=c=normal(1,5)` yaitu bilangan acak distribusi normal sebanyak 5 bilangan.
- `points=true` sebagai parameter yang memerintahkan `plot3d` akan menampilkan `points`(titik-titik).

Mengatur Tampilan, Warna dan Angle Gambar Permukaan 3D

Dalam plot3d terdapat banyak function terkait tampilan gambar 3D, di antaranya:
sliced:

Memplot versi irisan (0=tidak, 1=arah-x, 2=arah-y).

hue :

Menghitung bayangan menggunakan sumber cahaya.

light, amb, max :

Mengontrol pengaturan bayangan titik cahaya, ambient dan maksimum.

contour :

Menampilkan garis level tebal (dengan level otomatis).

spectral:

Gunakan warna spektral alih-alih rona monokrom. Terdapat skema spektral dari spektral = 1 (> spektral) hingga spektral = 9. >Default >spektral adalah rona warna dan ini setara dengan color=-2 hingga color=-10.

xhue, yhue, zhue:

Gunakan koordinat ini sebagai pengganti sumber cahaya.

hues :

Matriks nilai rona dari 0 sampai 1 untuk bayangan untuk plot x-y-z. Matriks harus memiliki ukuran yang kompatibel dengan x, y, z.

contourcolor :

Warna garis kontur.

contourwidth :

Lebar garis kontur.

fillcolor :

Warna isian untuk permukaan 3d tanpa rona.

user :

Pengguna dapat memutar plot dengan keyboard kiri, kanan, atas, bawah. +,- memperbesar plot. Spasi mengatur ulang plot. Return mengakhiri interaksi pengguna. Tombol a menghasilkan plot anaglyph. Tombol l mengalihkan pergerakan sumber cahaya untuk plot rona. Tombol c menggerakkan plot ke atas, bawah, kiri, atau kanan.

rotate :

Memutar plot sebuah fungsi dalam satu ekspresi dalam x.

anaglyph :

Menghasilkan plot 3d anaglyph (>anaglyph). Plot ini membutuhkan kacamata merah untuk dapat dilihat dengan baik.

viewangle :

Sudut pandang default, diputar di sekitar z-

zoom :

Pembesaran tampilan. Standarnya adalah sekitar 2,6.

view :

Tampilan lengkap, vektor 1x4 yang berisi jarak, zoom, sudut pandang, tinggi pandang.

center :

Vektor ini memindahkan pusat plot. Hal ini diperlukan jika plot tidak dipusatkan di (0,0,0) secara otomatis.

style :

Gaya plot.

color :

Warna untuk objek dan permukaan yang diarsir

wirecolor :

Warna untuk plot kawat

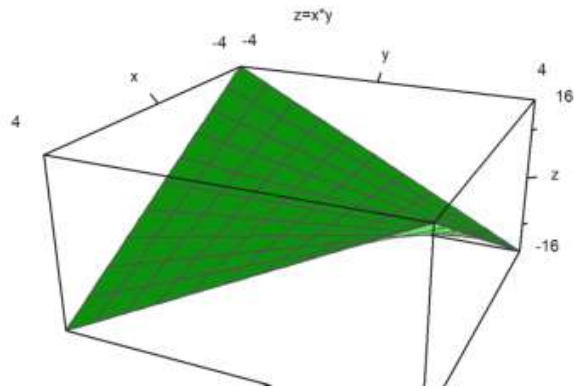
cp :

Menggambar bidang kontur di bawah plot (>cp).

cpcolor :

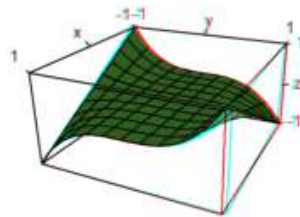
Warna untuk bidang kontur.

```
>plot3d("x*y",r=4,title="z=x*y",zoom=5):
```

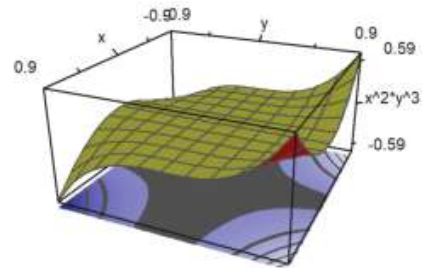


```
>plot3d("x*y^3",>user,r=1,>anaglyph,title="Press cursor keys or return!"):
```

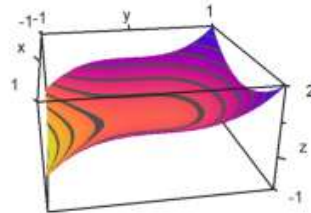
Press cursor keys or return!



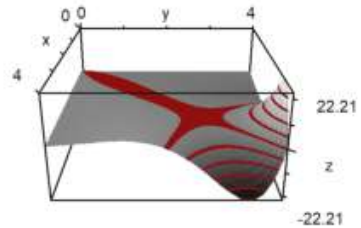
```
>plot3d("x^2*y^3",r=0.9,zlabel="x^2*y^3",>user,zoom=3, ...  
>fillcolor=[2,6],>cp,cpcolor=blue):
```



```
>plot3d("x^2+y^3",angle=80,>contour,spectral=2):
```

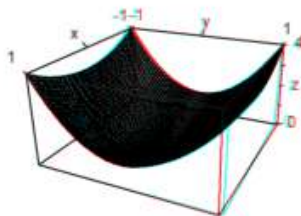


```
>plot3d("x^y-y^x",a=0,b=4,c=0,d=4,angle=90°,>contour, ...  
> contourwidth=4,contourcolor=red):
```

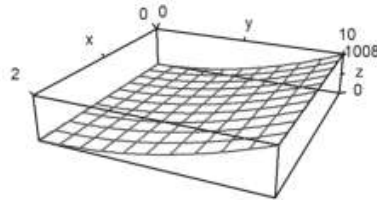


```
>plot3d("x^2+3y^2",>wire,>anaglyph,title="Use Red/Cyan Glasses!"):
```

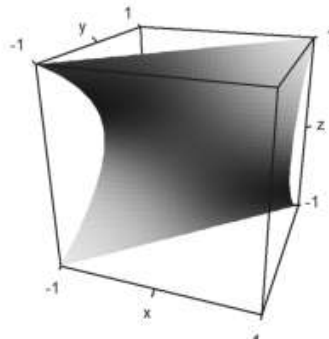
Use Red/Cyan Glasses!



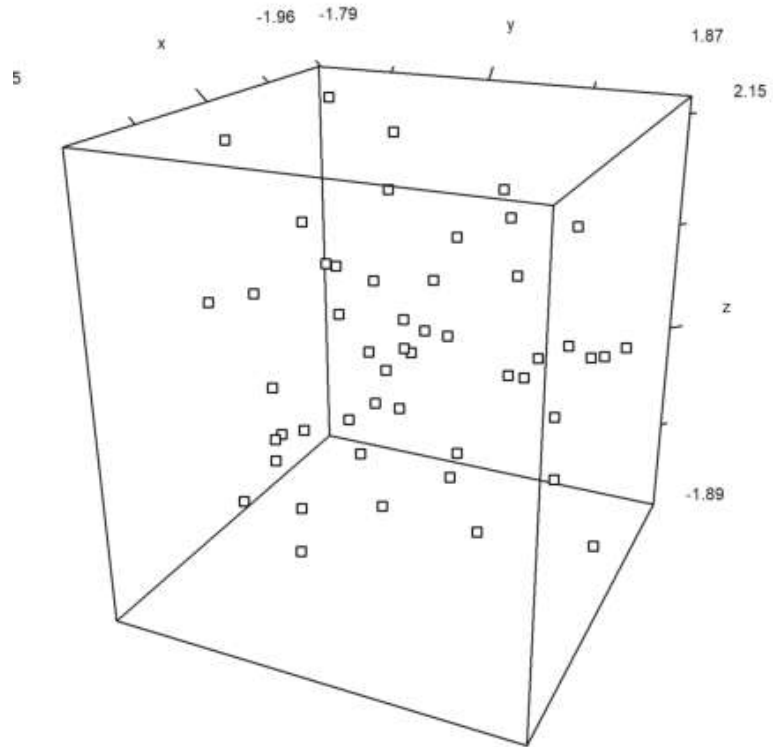
```
>plot3d("x^3+10y^2",0,2,0,10,scale=[5,1,2],zoom=3,grid=10,>transparent):
```



```
>x=-1:0.05:1; y=x'; plot3d(x,x*y^2,y,>user,>hue,angle=20°):
```




```
>X=normal(3,50); plot3d(X[1],X[2],X[3],>points,style="/",zoom=3,>user):
```



Plot Kontur

Untuk plot, Euler menambahkan garis grid. Sebagai gantinya dimungkinkan untuk menggunakan garis level dan rona satu warna atau rona berwarna spektral. Euler dapat menggambar tinggi fungsi pada plot dengan bayangan. Di semua plot 3D, Euler dapat menghasilkan anaglyph merah/sian.

- >hue: Menyalakan bayangan cahaya alih-alih kabel.
- >contour: Memplot garis kontur otomatis pada plot.
- level=... (atau levels): Sebuah vektor nilai untuk garis kontur.

Standarnya adalah level="auto", yang menghitung beberapa garis level secara otomatis. Seperti yang Anda lihat di plot, level sebenarnya adalah rentang level.

Gaya default dapat diubah. Untuk plot kontur berikut, kami menggunakan grid yang lebih halus untuk 100x100 poin, skala fungsi dan plot, dan menggunakan sudut pandang yang berbeda.

```
>plot3d("exp(-x^2-y^3)",r=2,n=100,level="thin", ...
```

1."exp(-x²-y³)": Ini adalah fungsi matematika yang akan digunakan untuk membuat plot 3D. Fungsi ini adalah fungsi Gaussian dua dimensi yang bergantung pada variabel x dan y.

2.r = 2: Ini adalah parameter r yang mengatur jangkauan atau rentang dari plot 3D. Dalam hal ini, r diatur menjadi 2, yang mungkin mengacu pada jangkauan plot dalam koordinat x dan y.

3.n = 100: Ini adalah parameter n yang mengatur jumlah titik sampel dalam plot. Dalam hal ini, ada 100 titik sampel yang akan digunakan untuk menggambarkan plot.

4.level = "thin": Ini adalah parameter level yang mengatur ketebalan atau tipe garis yang digunakan dalam plot. "Thin" mungkin mengacu pada penggunaan garis tipis dalam plot.

5.contour: Ini adalah parameter yang menambahkan garis kontur ke plot. Ini memungkinkan Anda melihat kontur atau garis isovalue dalam plot yang menggambarkan tingkat nilai fungsi.

6.spectral: Ini adalah parameter yang mengatur skema warna plot menjadi skema warna spektral.

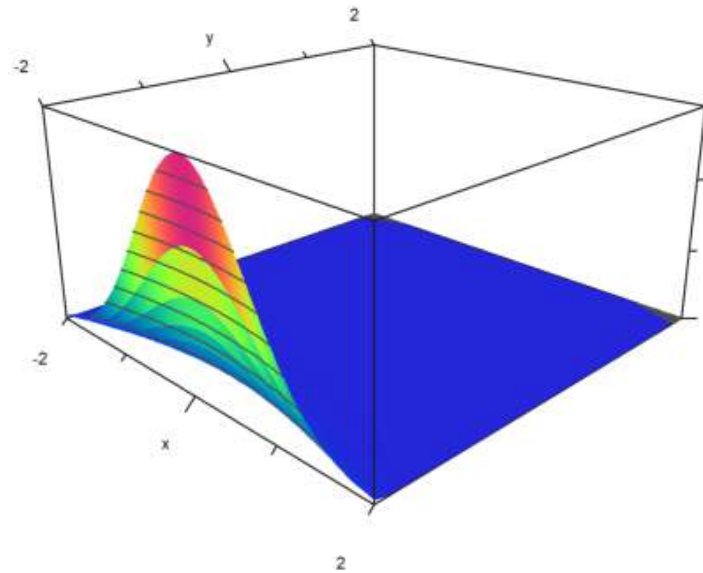
7.fscale = 1: Ini adalah parameter fscale yang mengatur skala faktor untuk plot.

8.scale = 1.1: Ini adalah parameter scale yang mengatur faktor skala untuk plot.

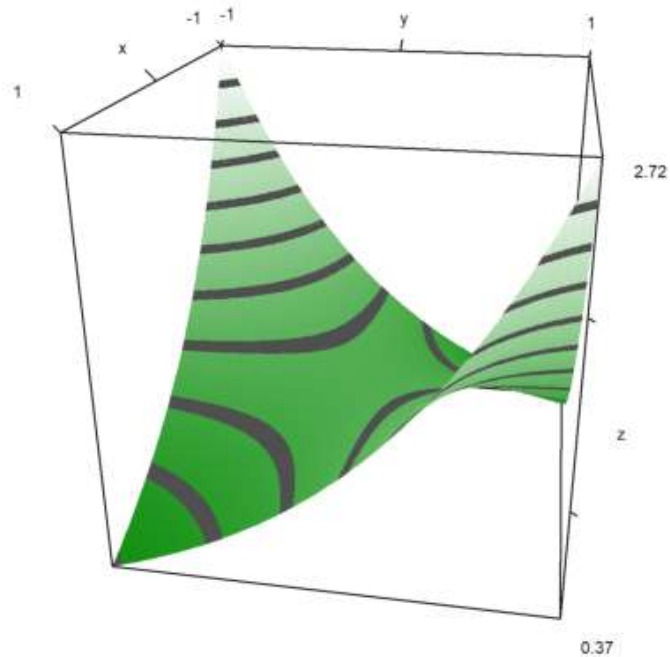
9.angle = 45°: Ini adalah parameter angle yang mengatur sudut tampilan plot. Dalam hal ini, plot akan dilihat dari sudut 45 derajat.

10.height = 20°: Ini adalah parameter height yang mengatur tinggi plot dalam derajat.

```
>>contour,>spectral,fscale=1,scale=1.1,angle=45°,height=20°):
```



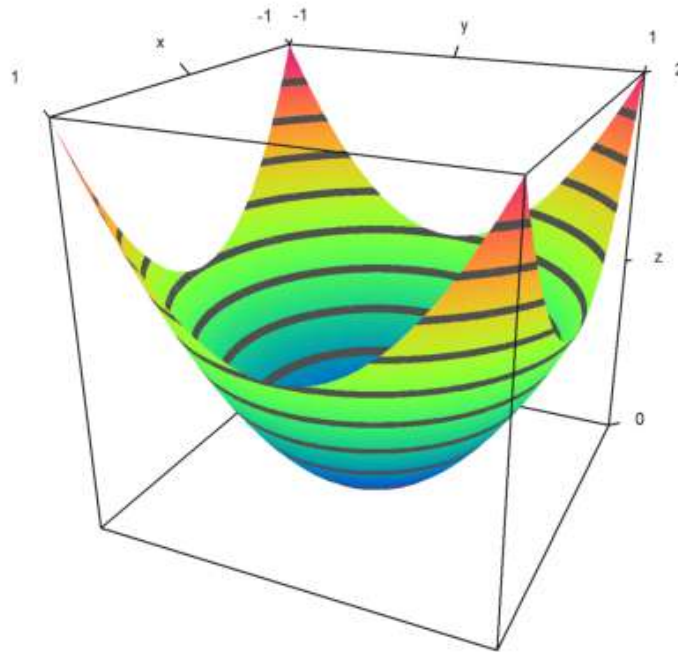
```
>plot3d("exp(x*y)",angle=100°,>contour,color=green):
```



1. "exp(x*y)": Ini adalah fungsi matematika yang akan digunakan untuk membuat plot 3D. Fungsi ini adalah eksponensial dari hasil perkalian antara x dan y. Dalam konteks ini, x dan y adalah variabel-variabel dalam plot.
2. angle = 100°: Ini adalah parameter angle yang mengatur sudut tampilan plot. Dalam hal ini, plot akan dilihat dari sudut 100 derajat.
3. contour: Ini adalah parameter yang menambahkan garis kontur ke plot. Ini memungkinkan Anda melihat kontur atau garis isovalue dalam plot yang menggambarkan tingkat nilai fungsi.

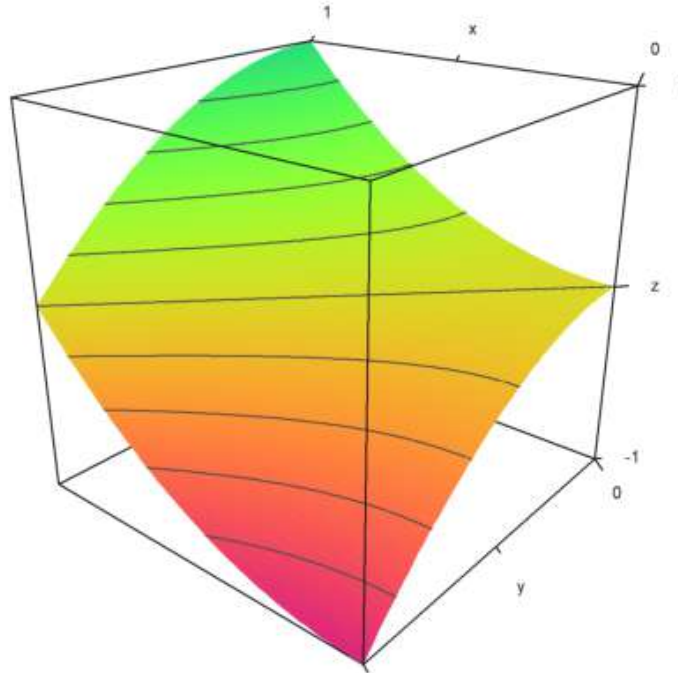
4. color = green: Ini adalah parameter color yang mengatur warna plot. Dalam hal ini, plot akan menggunakan warna hijau.

```
>plot3d("x^2+y^2",>spectral,>contour,n=100):
```



1. $x^2 + y^2$: Ini adalah fungsi matematika yang akan digunakan untuk membuat plot 3D. Fungsi ini adalah fungsi kuadrat dari variabel x dan y . Dalam konteks ini, x dan y adalah variabel-variabel dalam plot.
2. spectral: Ini adalah parameter yang mengatur skema warna plot menjadi skema warna spektral. Dengan menggunakan skema warna spektral, berbagai nilai dalam plot akan diberikan warna yang berbeda, yang memudahkan untuk memahami perubahan nilai dalam fungsi.
3. contour: Ini adalah parameter yang menambahkan garis kontur ke plot. Ini memungkinkan Anda melihat kontur atau garis isovalue dalam plot yang menggambarkan tingkat nilai fungsi.
4. $n = 100$: Ini adalah parameter n yang mengatur jumlah titik sampel dalam plot. Dalam hal ini, ada 100 titik sampel yang akan digunakan untuk menggambarkan plot. Semakin banyak titik sampel, semakin halus plotnya.

```
>plot3d("x^2-y^2",0,1,0,1,angle=220°,level=-1:0.2:1,color=redgreen):
```

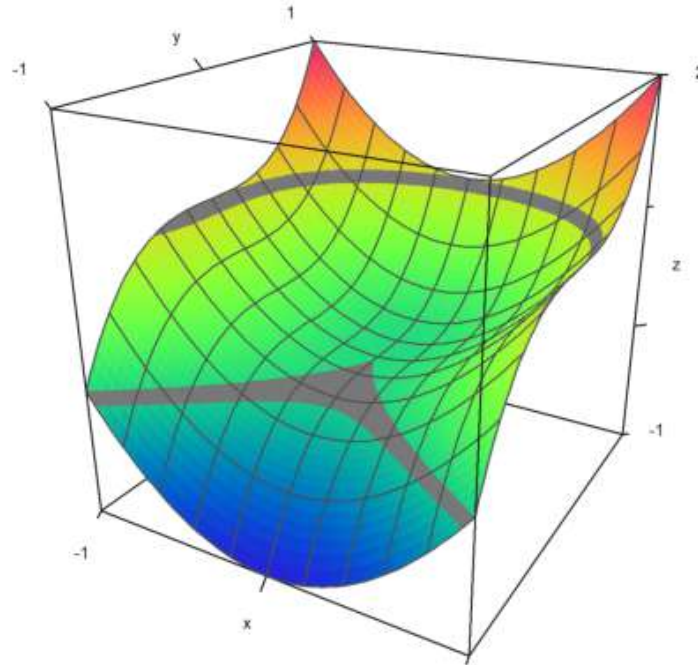


1. " $x^2 - y^2$ ": Ini adalah fungsi matematika yang akan digunakan untuk membuat plot 3D. Fungsi ini adalah perbedaan antara kuadrat variabel x dan kuadrat variabel y . Dalam konteks ini, x dan y adalah variabel-variabel dalam plot.
2. 0, 1, 0, 1: Ini adalah parameter yang mengatur batasan tampilan plot. Angka-angka ini mewakili batas minimum dan maksimum untuk x dan y . Dalam hal ini, plot akan dibatasi pada rentang x dan y antara 0 dan 1.
3. $\text{angle} = 220^\circ$: Ini adalah parameter angle yang mengatur sudut tampilan plot. Dalam hal ini, plot akan dilihat dari sudut 220 derajat.

4. `level = -1:0.2:1`: Ini adalah parameter `level` yang mengatur tingkat nilai fungsi yang akan ditampilkan dalam plot. Rentang ini (-1 hingga 1) akan dibagi menjadi beberapa tingkat, dengan selang 0.2 antara masing-masing tingkat. Ini akan menghasilkan garis kontur pada tingkat nilai fungsi tertentu.

5. `color = redgreen`: Ini adalah parameter `color` yang mengatur skema warna plot. Warna yang digunakan adalah kombinasi warna merah dan hijau.

```
>plot3d("x^2+y^3",level=[-0.1,0.9;0,1], ...  
> >spectral,angle=30°,grid=10,contourcolor=gray):
```



1. " $x^2 + y^3$ ": Ini adalah fungsi matematika yang akan digunakan untuk membuat plot 3D. Fungsi ini adalah hasil penjumlahan dari kuadrat variabel x dan kubik variabel y . Dalam konteks ini, x dan y adalah variabel-variabel dalam plot.

2. `level = [-0.1, 0.9; 0, 1]`: Ini adalah parameter level yang mengatur tingkat nilai fungsi yang akan ditampilkan dalam plot. Parameter ini didefinisikan sebagai matriks dua baris dengan dua kolom. Setiap baris berisi batasan tingkat nilai fungsi yang akan ditampilkan dalam plot. Misalnya, `[-0.1, 0.9]` menunjukkan bahwa tingkat nilai akan ditampilkan dari -0.1 hingga 0.9, dan `[0, 1]` menunjukkan bahwa tingkat nilai kedua akan ditampilkan dari 0 hingga 1.

3.spectral: Ini adalah parameter yang mengatur skema warna plot menjadi skema warna spektral.

4.angle = 30°: Ini adalah parameter angle yang mengatur sudut tampilan plot. Dalam hal ini, plot akan dilihat dari sudut 30 derajat.

5.grid = 10: Ini adalah parameter grid yang mengatur jumlah garis kisi dalam plot. Dalam hal ini, akan ada 10 garis kisi dalam plot.

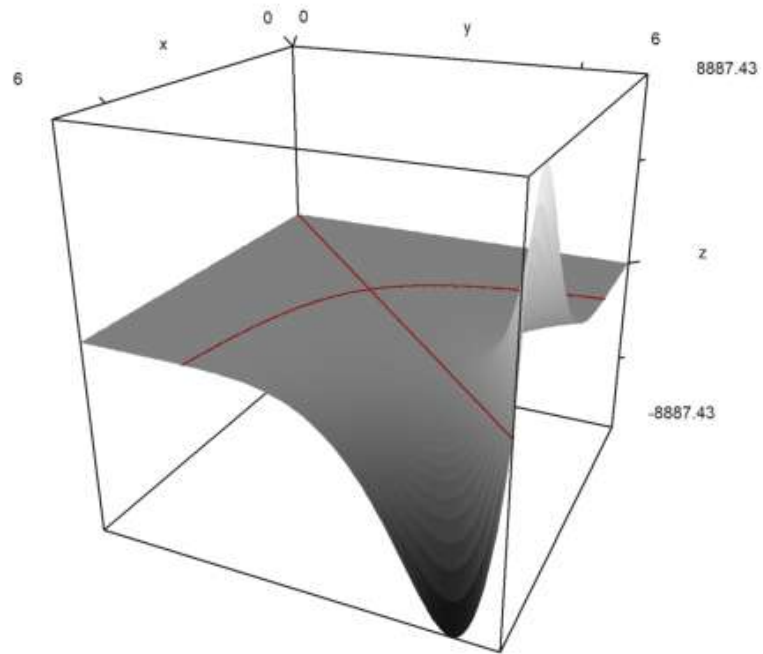
6.contourcolor = gray: Ini adalah parameter contourcolor yang mengatur warna garis kontur dalam plot menjadi abu-abu (gray).

Dalam contoh berikut, kami memplot himpunan, di mana

$$f(x, y) = x^y - y^x = 0$$

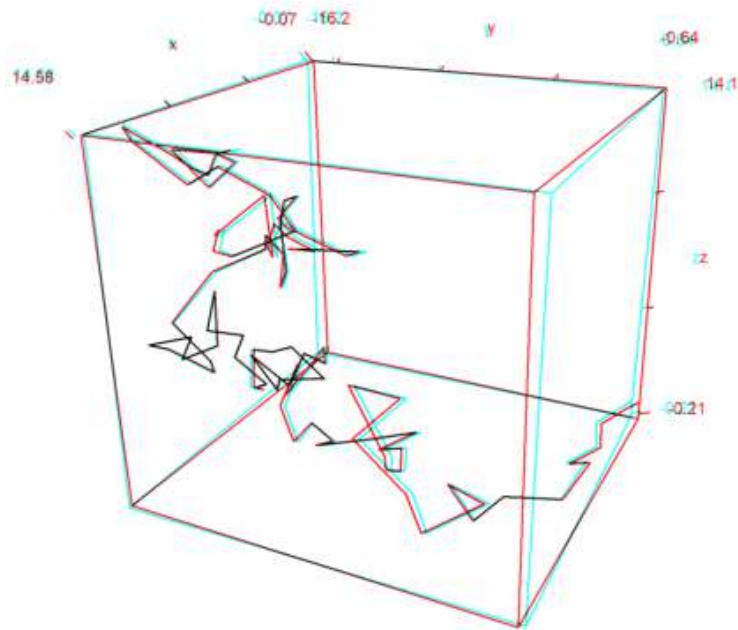
Kami menggunakan satu garis tipis untuk garis level.

```
>plot3d("x^y-y^x",level=0,a=0,b=6,c=0,d=6,contourcolor=red,n=100):
```



Menggambar Grafik Tiga Dimensi Alam Modus Anaglif

```
>X=cumsum(normal(3,100)); ...  
> plot3d(X[1],X[2],X[3],>anaglyph,>wire):
```



1.X = cumsum(normal(3, 100));: Ini adalah urutan perintah yang melakukan beberapa operasi berurutan.

a)normal(3, 100): Ini adalah panggilan fungsi normal yang digunakan untuk menghasilkan 100 bilangan acak dengan distribusi normal (Gaussian) dengan rata-rata 3 dan deviasi standar 1. Hasilnya adalah vektor tiga dimensi yang berisi 100 titik acak dalam ruang tiga dimensi.

b)cumsum(...): Ini adalah panggilan fungsi cumsum yang digunakan untuk menghitung kumulatif dari vektor 3D yang dihasilkan sebelumnya. Dengan kata lain, ini akan menghasilkan vektor yang merupakan akumulasi (penjumlahan berulang) dari vektor 3D tersebut. Hasilnya adalah vektor tiga dimensi yang menggambarkan perjalanan dalam ruang 3D berdasarkan perubahan titik acak.

2.plot3d(X[1], X[2], X[3], anaglyph, wire);: Ini adalah perintah untuk membuat plot 3D dari data yang telah dihasilkan sebelumnya.

a).X[1], X[2], dan X[3] adalah komponen vektor tiga dimensi X yang akan digunakan sebagai koordinat dalam plot 3D. X[1] digunakan sebagai koordinat sumbu x, X[2] digunakan sebagai koordinat sumbu y, dan X[3] digunakan sebagai koordinat sumbu z.

b)anaglyph: Ini adalah parameter yang mengatur plot menggunakan efek anaglif. Anaglif adalah teknik yang digunakan untuk menghasilkan efek tiga dimensi (3D) dengan menggunakan dua gambar yang sedikit berbeda untuk mata kiri dan kanan, dan penonton memerlukan kacamata anaglif khusus untuk melihat efek 3D.

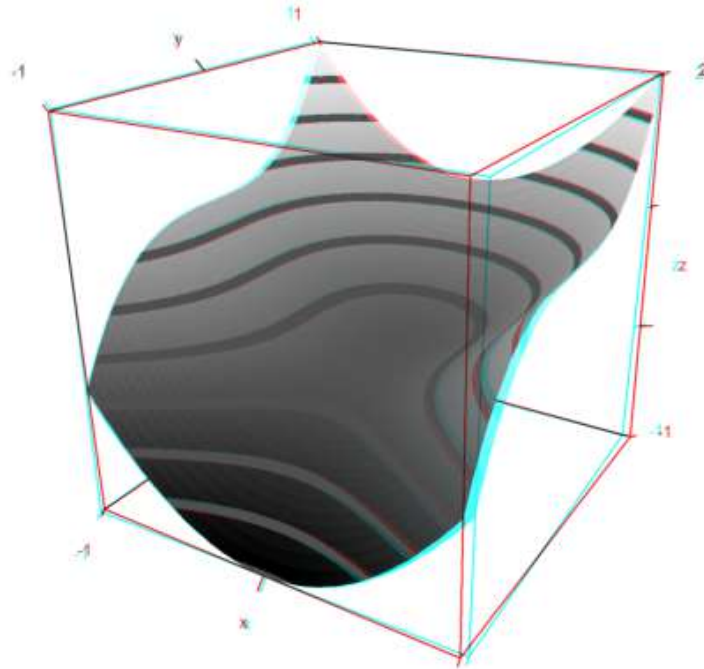
c)wire: Ini adalah parameter yang mengatur plot sebagai plot tali (wireframe), yang berarti hanya garis-garis yang menghubungkan titik-titik yang akan ditampilkan dalam plot.

4.X[1], X[2], dan X[3]: Ini adalah komponen dari vektor X. X[1] digunakan sebagai koordinat sumbu x, X[2] digunakan sebagai koordinat sumbu y, dan X[3] digunakan sebagai koordinat sumbu z. Dengan menggunakan komponen vektor ini sebagai koordinat, Anda membuat plot 3D yang merepresentasikan perubahan dalam tiga dimensi berdasarkan data dalam vektor X.

5.anaglyph: Ini adalah parameter yang mengatur plot menggunakan efek anaglif. Efek anaglif adalah teknik yang digunakan untuk menghasilkan efek tiga dimensi (3D) dengan menggunakan dua gambar yang sedikit berbeda untuk mata kiri dan kanan. Penonton memerlukan kacamata anaglif khusus dengan lensa berwarna berbeda untuk mata kiri dan kanan untuk melihat efek 3D ini. Penggunaan anaglyph dalam kode ini menunjukkan bahwa plot akan dibuat dengan efek anaglif.

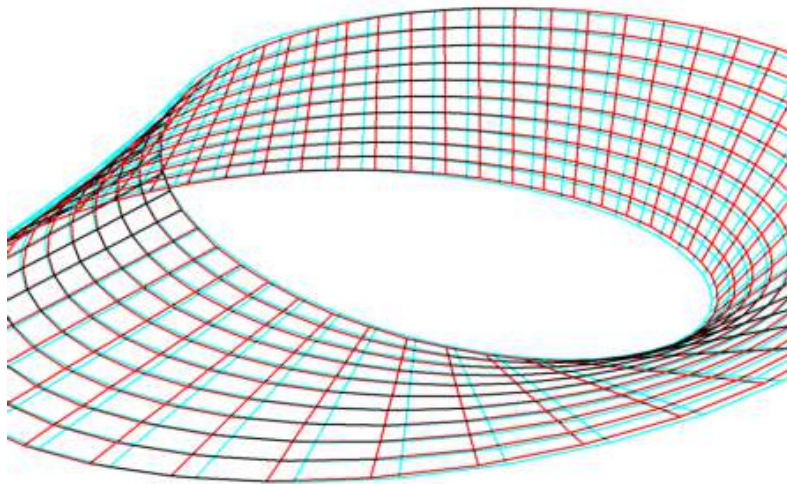
6.wire: Ini adalah parameter yang mengatur plot sebagai plot tali (wireframe). Wireframe adalah gaya plot di mana hanya garis-garis yang menghubungkan titik-titik yang akan ditampilkan dalam plot. Dengan pengaturan wire, Anda akan melihat plot dalam bentuk rangkaian garis-garis yang menggambarkan bentuk objek dalam tampilan 3D.

```
>plot3d("x^2+y^3",>anaglyph,>contour,angle=30°):
```



1. $x^2 + y^3$: Ini adalah fungsi matematika yang akan digunakan untuk membuat plot 3D. Fungsi ini adalah hasil penjumlahan dari kuadrat variabel x dan kubik variabel y . Dalam konteks ini, x dan y adalah variabel-variabel dalam plot.
2. `anaglyph`: Ini adalah parameter yang mengatur plot menggunakan efek anaglif. Anaglif adalah teknik yang digunakan untuk menghasilkan efek tiga dimensi (3D) dengan menggunakan dua gambar yang sedikit berbeda untuk mata kiri dan kanan. Penonton memerlukan kacamata anaglif khusus dengan lensa berwarna berbeda untuk mata kiri dan kanan untuk melihat efek 3D ini. Dengan pengaturan ini, plot akan dibuat dengan efek 3D anaglif.
3. `contour`: Ini adalah parameter yang menambahkan garis kontur ke plot. Ini memungkinkan Anda melihat kontur atau garis isovalue dalam plot yang menggambarkan tingkat nilai fungsi.
4. `angle = 30`: Ini adalah parameter `angle` yang mengatur sudut tampilan plot. Dalam hal ini, plot akan dilihat dari sudut 30 derajat.

```
>u=linspace(-1,1,10); v=linspace(0,2*pi,50)'; ...  
>X=(3+u*cos(v/2))*cos(v); Y=(3+u*cos(v/2))*sin(v); Z=u*sin(v/2); ...  
>plot3d(X,Y,Z,>anaglyph,<frame,>wire,scale=2.3):
```

1. `u = linspace(-1, 1, 10);` Ini adalah perintah untuk membuat vektor `u` yang berisi 10 nilai yang dihasilkan secara merata dalam rentang -1 hingga 1. Vektor ini akan digunakan dalam perhitungan selanjutnya.

2. `v = linspace(0, 2 * pi, 50);` Ini adalah perintah untuk membuat vektor `v` yang berisi 50 nilai yang dihasilkan secara merata dalam rentang 0 hingga 2π (dua kali nilai π). Vektor ini juga akan digunakan dalam perhitungan selanjutnya.

3.X, Y, dan Z: Ini adalah perintah-perintah yang digunakan untuk menghasilkan data koordinat dalam tiga dimensi. Data ini dihasilkan berdasarkan persamaan yang menggunakan nilai u dan v. Data ini akan digunakan untuk membuat plot 3D.

4.plot3d(X, Y, Z, anaglyph, <frame, wire, scale = 2.3); Ini adalah perintah untuk membuat plot 3D berdasarkan data X, Y, dan Z yang telah dihasilkan sebelumnya. Parameter-parameter yang digunakan dalam perintah ini adalah:

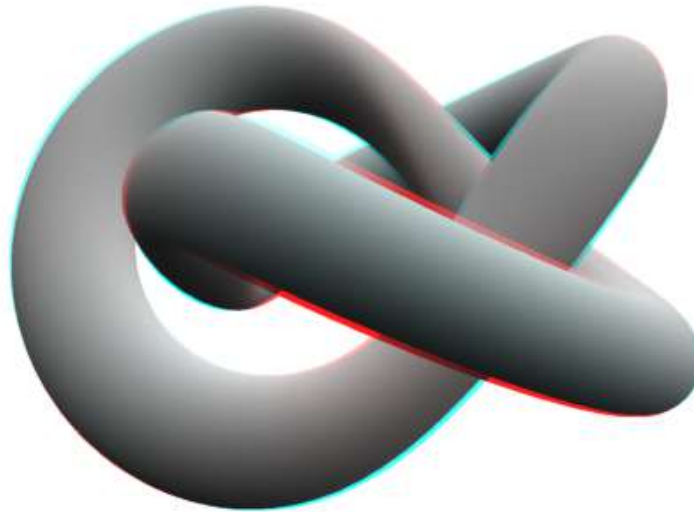
a)anaglyph: Ini adalah parameter yang mengatur plot menggunakan efek anaglif, yang memberikan efek tiga dimensi (3D) saat melihat plot dengan kacamata anaglif.

b)<frame: Ini adalah parameter yang mengatur agar frame (kerangka) plot ditampilkan. Ini adalah bingkai atau batasan dari plot.

c)wire: Ini adalah parameter yang mengatur plot dalam bentuk rangkaian garis-garis (wireframe).

d)scale = 2.3: Ini adalah parameter yang mengatur faktor skala plot sebesar 2.3. Ini akan memperbesar plot.

```
>u=linspace(-pi,pi,160); v=linspace(-pi,pi,400)'; ...
>x=(4*(1+.25*sin(3*v))+cos(u))*cos(2*v); ...
>y=(4*(1+.25*sin(3*v))+cos(u))*sin(2*v); ...
> z=sin(u)+2*cos(3*v); ...
>plot3d(x,y,z,frame=0,scale=1.5,hue=1,light=[1,0,-1],zoom=2.8,>anaglyph):
```



1. $u := \text{linspace}(-\pi, \pi, 160);$: Ini adalah perintah untuk membuat vektor u yang berisi 160 nilai yang dihasilkan secara merata dalam rentang dari $-p$ hingga p .

2. $v := \text{linspace}(-\pi, \pi, 400)';$: Ini adalah perintah untuk membuat vektor v yang berisi 400 nilai yang dihasilkan secara merata dalam rentang dari $-p$ hingga p . Vektor ini diubah menjadi matriks kolom dengan penambahan tanda apostrof ($'$) di belakangnya.

3. $x, y,$ dan z : Ini adalah perintah-perintah yang digunakan untuk menghasilkan data koordinat dalam tiga dimensi. Data ini dihasilkan berdasarkan persamaan yang menggunakan nilai u dan v . Data ini akan digunakan untuk membuat plot 3D.

4.plot3d(x, y, z, frame = 0, scale = 1.5, hue = 1, light = [1, 0, -1], zoom = 2.8, anaglyph);: Ini adalah perintah untuk membuat plot 3D berdasarkan data x, y, dan z yang telah dihasilkan sebelumnya. Parameter-parameter yang digunakan dalam perintah ini adalah:

a)frame = 0: Ini adalah parameter yang mengatur agar frame (kerangka) plot tidak ditampilkan.

b)scale = 1.5: Ini adalah parameter yang mengatur faktor skala plot sebesar 1.5. Ini akan memperbesar plot.

c)hue = 1: Ini adalah parameter yang mengatur warna plot dengan skala warna tunggal.

d)light = [1, 0, -1]: Ini adalah parameter yang mengatur pencahayaan plot dengan arah cahaya yang ditentukan oleh vektor [1, 0, -1].

e)zoom = 2.8: Ini adalah parameter yang mengatur tingkat zoom plot sebesar 2.8. Ini akan memperbesar plot.

f)anaglyph: Ini adalah parameter yang mengatur plot menggunakan efek anaglif, yang memberikan efek tiga dimensi (3D) saat dilihat dengan kacamata anaglif.

13. Plot Statistik batang 3d

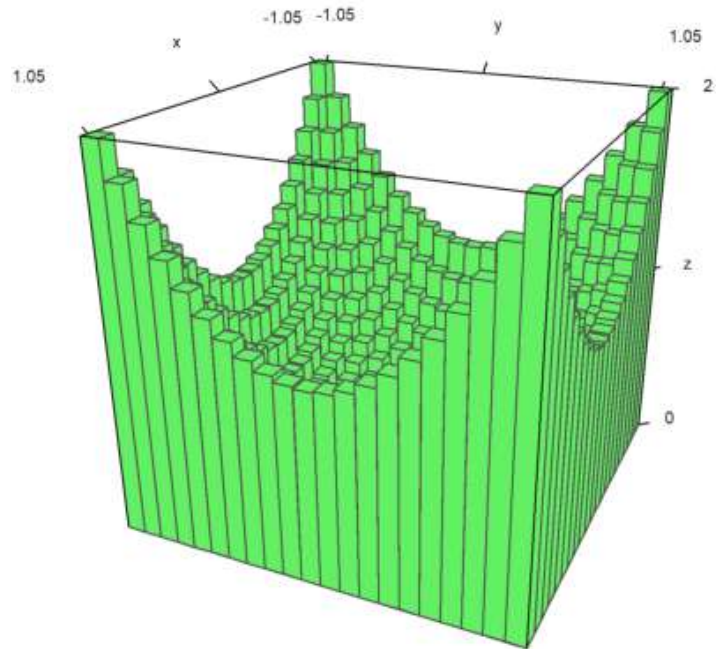
Plot bar juga dimungkinkan. Untuk ini, kita harus menyediakan

- x: vektor baris dengan n+1 elemen
- y: vektor kolom dengan n+1 elemen
- z: matriks nilai nxn.

z bisa lebih besar, tetapi hanya nilai nxn yang akan digunakan.

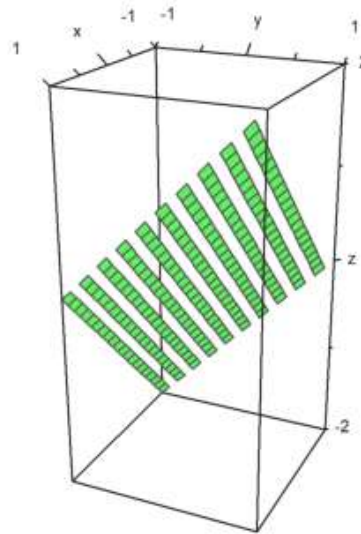
Dalam contoh, pertama-tama kita menghitung nilainya. Kemudian kita sesuaikan x dan y, sehingga vektor berpusat pada nilai yang digunakan.

```
>x=-1:0.1:1; y=x'; z=x^2+y^2; ...  
>xa=(x|1.1)-0.05; ya=(y_1.1)-0.05; ...  
>plot3d(xa,ya,z,bar=true):
```



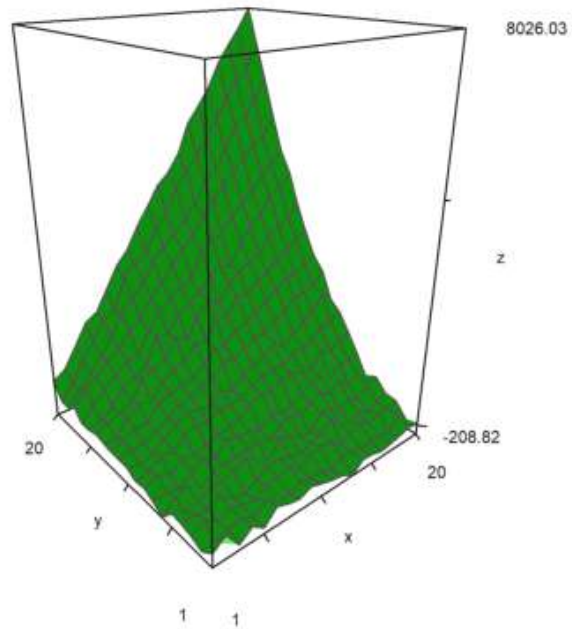
Dimungkinkan untuk membagi plot permukaan menjadi dua atau lebih bagian.

```
>x=-1:0.1:1; y=x'; z=x+y; d=zeros(size(x)); ...  
>plot3d(x,y,z,disconnect=2:2:20):
```

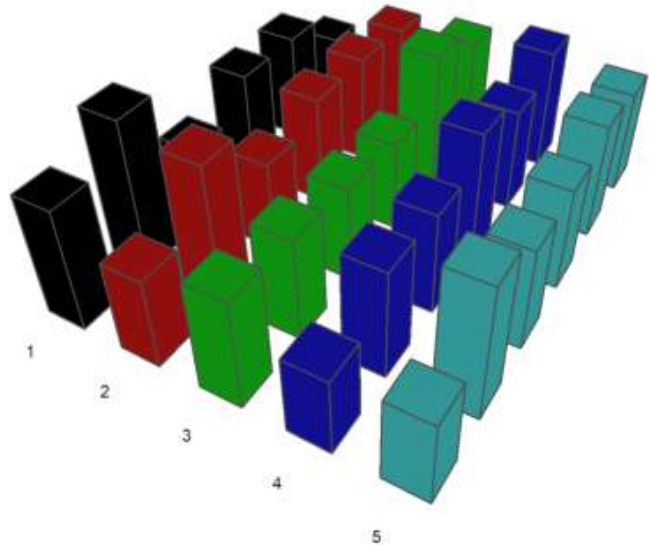


Jika memuat atau menghasilkan matriks data M dari file dan perlu memplotnya dalam 3D, Anda dapat menskalakan matriks ke $[-1,1]$ dengan `scale(M)`, atau menskalakan matriks dengan `>zscale`. Ini dapat dikombinasikan dengan faktor penskalaan individu yang diterapkan sebagai tambahan.

```
>i=1:20; j=i'; ...  
>plot3d(i*j^2+100*normal(20,20),>zscale,scale=[1,1,1.5],angle=-40°,zoom=1.8):
```

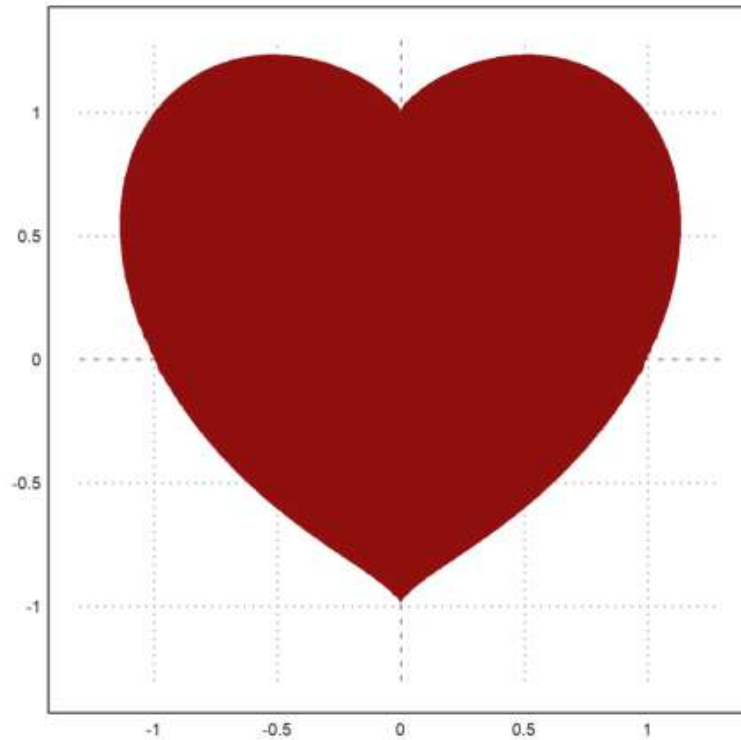


```
>Z=intrandom(5,100,6); v=zeros(5,6); ...  
>loop 1 to 5; v[#]=getmultiplicities(1:6,Z[#]); end; ...  
>columnplot3d(v',scols=1:5,ccols=[1:5]):
```



Permukaan Benda Putar

```
>plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1.3, ...  
>style="#",color=red,<outline, ...  
>level=[-2;0],n=100):
```



```
>ekspresi &= (x^2+y^2-1)^3-x^2*y^3; $ekspresi
```

$$(y^2 + x^2 - 1)^3 - x^2 y^3$$

Kami ingin memutar kurva jantung di sekitar sumbu y. Berikut adalah ungkapan, yang mendefinisikan hati:

$$f(x, y) = (x^2 + y^2 - 1)^3 - x^2 \cdot y^3.$$

Selanjutnya kita atur

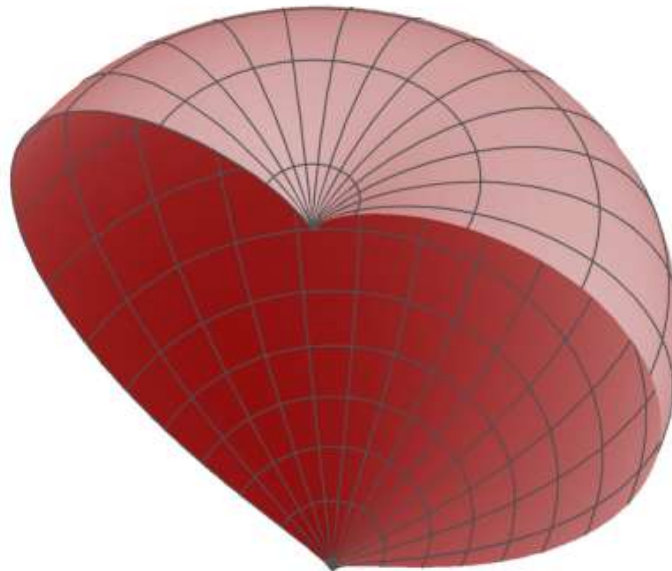
$$x = r \cdot \cos(a), \quad y = r \cdot \sin(a).$$

```
>function fr(r,a) &= ekspresi with [x=r*cos(a),y=r*sin(a)] | trigreduce; $fr(r,a)
```

$$(r^2 - 1)^3 + \frac{(\sin(5a) - \sin(3a) - 2 \sin a) r^5}{16}$$

Hal ini memungkinkan untuk mendefinisikan fungsi numerik, yang memecahkan r, jika a diberikan. Dengan fungsi itu kita dapat memplot jantung yang diputar sebagai permukaan parametrik.

```
>function map f(a) := bisect("fr",0,2;a); ...  
>t=linspace(-pi/2,pi/2,100); r=f(t); ...  
>s=linspace(pi,2pi,100)'; ...  
>plot3d(r*cos(t)*sin(s),r*cos(t)*cos(s),r*sin(t), ...  
>>hue,<frame,color=red,zoom=4,amb=0,max=0.7,grid=12,height=50°):
```

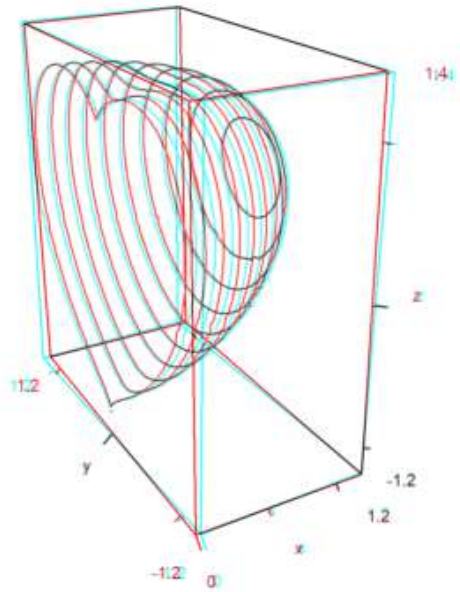


Berikut ini adalah plot 3D dari gambar di atas yang diputar di sekitar sumbu z. Kami mendefinisikan fungsi, yang menggambarkan objek.

```
>function f(x,y,z) ...
```

```
    r=x^2+y^2; ...  
    return (r+z^2-1)^3-r*z^3;  
endfunction
```

```
>plot3d("f(x,y,z)", ...  
>xmin=0,xmax=1.2,ymin=-1.2,ymax=1.2,zmin=-1.2,zmax=1.4, ...  
>implicit=1,angle=-30°,zoom=2.5,n=[10,60,60],>anaglyph):
```



Lebih Banyak Contoh

Anda dapat menemukan beberapa contoh lagi untuk Povray di Euler di file berikut.

See: [Examples/Dandelin Spheres](#)

See: [Examples/Donat Math](#)

See: [Examples/Trefoil Knot](#)

See: [Examples/Optimization by Affine Scaling](#)