

# Penggunaan Software EMT untuk Plot 3D

---

## Sub-topik Bahasan dan Pembagian Tugas:

---

1. Menggambar Grafik Fungsi Dua Variabel dalam Bentuk Ekspresi Langsung (Nasywa)
  2. Menggambar Grafik Fungsi Dua Variabel yang Rumusnya Disimpan dalam Variabel Ekspresi (Fivi)
  3. Menggambar Grafik Fungsi Dua Variabel yang Fungsinya Didefinisikan sebagai Fungsi Numerik (Wildan)
  4. Menggambar Grafik Fungsi Dua Variabel yang Fungsinya Didefinisikan sebagai Fungsi Simbolik (Ferdian)
  5. Menggambar Data  $x$ ,  $y$ ,  $z$  pada ruang Tiga Dimensi (3D) (Saveria)
  6. Membuat Gambar Grafik Tiga Dimensi (3D) yang Bersifat Interaktif dan animasi grafik 3D (Ega)
  7. Menggambar Fungsi Parametrik Tiga Dimensi (3D) (Arini)
  8. Menggambar Fungsi Implisit Tiga Dimensi (3D) (Artha)
  9. Menggambar Titik pada ruang Tiga Dimensi (3D) (Saveria)
  10. Mengatur tampilan, warna dan sudut pandang gambar permukaan Tiga Dimensi (3D) (Artha)
  11. Menampilkan kontur dan bidang kontur permukaan Tiga Dimensi (3D) (Nasywa)
  12. Menggambar Grafik Tiga Dimensi alam modus anaglif (Ega)
  13. Menggambar Diagram Batang Tiga Dimensi (Arini)
  14. Menggambar Permukaan Benda Putar (Ferdian)
  15. Menggambar Grafik 3D dengan Povray di EMT (Fivi dan Wildan)
-

## **1. Menggambar Grafik Fungsi Dua Variabel dalam Bentuk Ekspresi**

---

Langsung

Secara umum Euler menggunakan proyeksi sentral. Defaultnya adalah dari kuadran x-y positif menuju titik asal  $x=y=z=0$ , tetapi sudut= $0^\circ$  dilihat dari arah sumbu y. Dimana sudut pandang dan ketinggiannya dapat diubah.

Euler bisa merencanakan

- permukaan dengan bayangan dan level atau rentang level,
- point cloud,
- kurva parametrik,
- permukaan implisit.

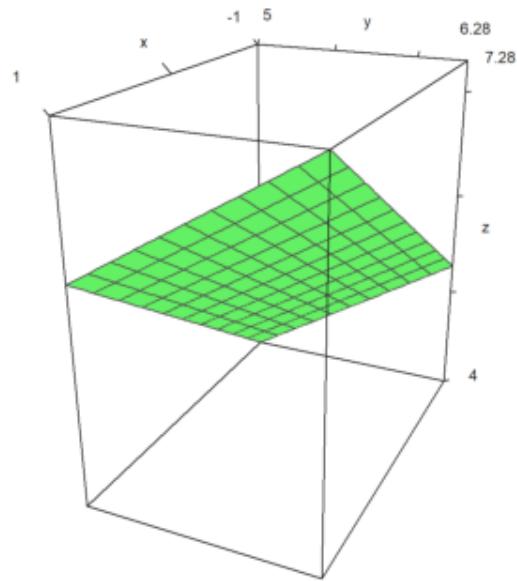
Pada ekspresi langsung ini fungsi atau ekspresi matematis dapat digunakan satu kali, jadi jika kita ingin menggunakan lagi pada baris perintah selanjutnya kita harus menuliskannya kembali secara lengkap.

Perintah yang digunakan pada plot3d adalah plot3d. Cara termudah adalah dengan memplot ekspresi dalam x dan y. Parameter r menetapkan rentang plot disekitar (0,0).

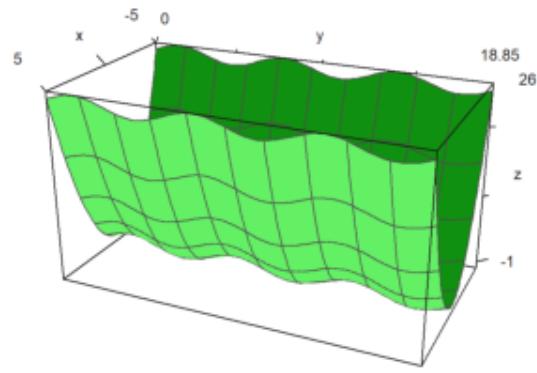
Contoh Grafik Fungsi Dua Variabel

---

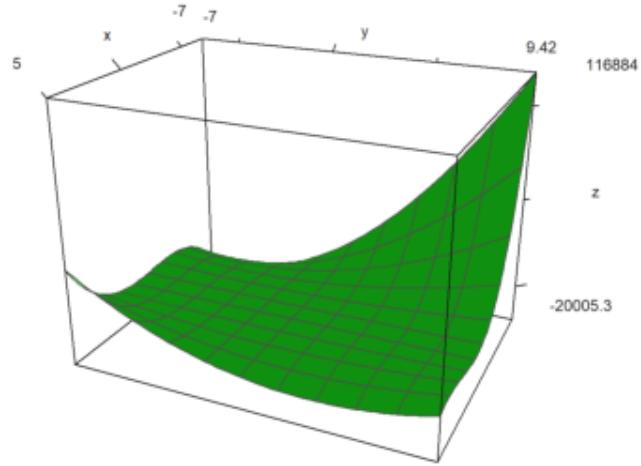
```
>plot3d("(x+y)",-1,1,5,2*pi):
```



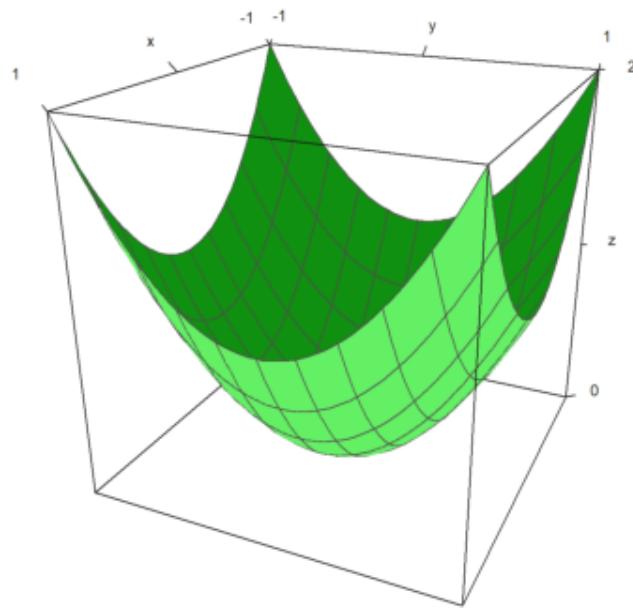
```
>aspect(1); plot3d("x^2+sin(y)",-5,5,0,6*pi):
```



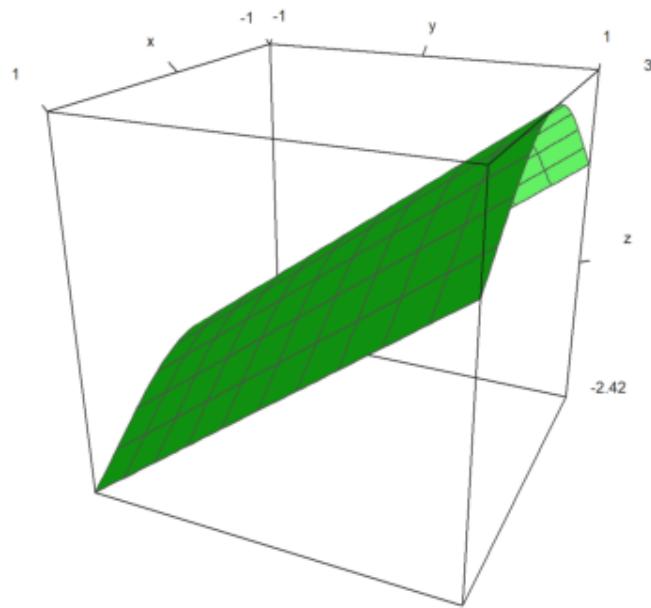
```
>plot3d("(12*x^2*y^2-20*x^3*y)", -7,5, -7,3*pi):
```



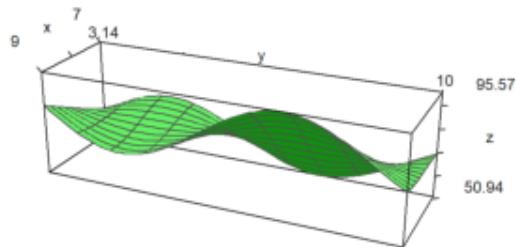
```
>aspect(1); plot3d("(6*x+y)^2", -3,5,0,5*pi)
>plot3d("x^2+y^2"):
```



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



```
>plot3d("9*x-16*cos(x)*sin(y)",7,9,10,pi):
```



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

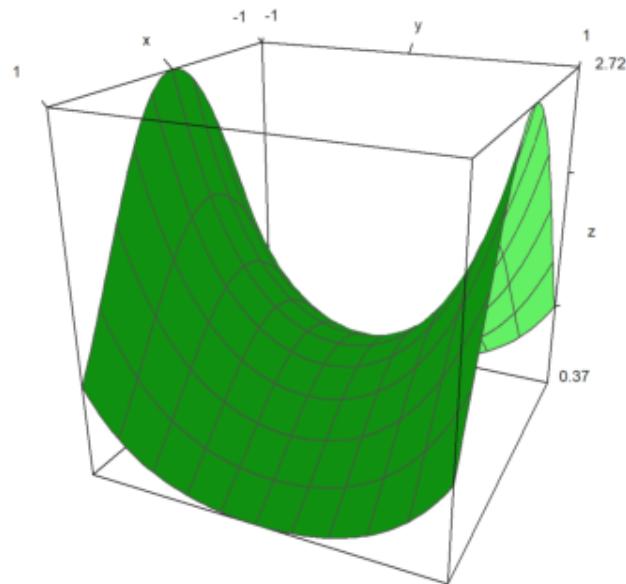
- left, right, up, down: memutar sudut pandang
- +,-: memperbesar atau memperkecil
- a: menghasilkan anaglyph (lihat di bawah)
- l : tombol nyalakan sumber cahaya (lihat dibawah)
- space: reset ke default
- return: mengakhiri interaksi

Contoh

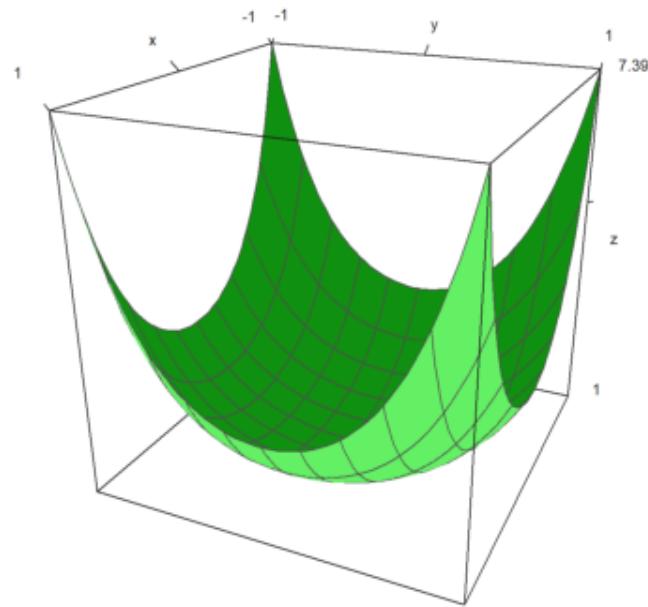
---

```
>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):
```



Kita juga dapat mengubah tampilan dari grafik dengan berbagai cara.

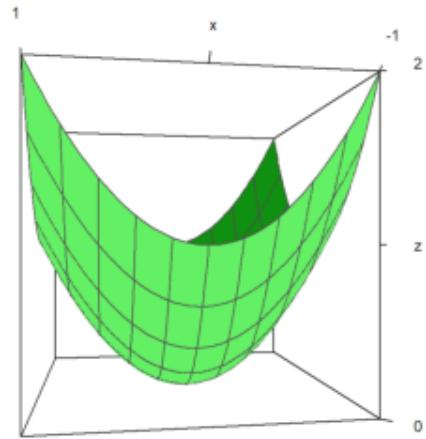
- distance: jarak pandang ke plot.
- zoom: nilai zoom.
- angle: sudut terhadap sumbu y negatif dalam radian.
- height: ketinggian pandangan dalam radian.

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

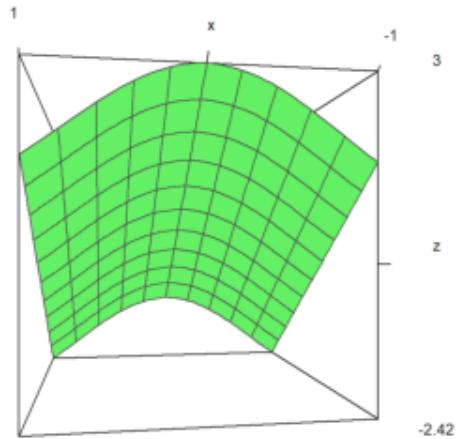
## Contoh

---

```
>plot3d("x^2+y^2",distance=4,zoom=1.5,angle=3,height=0):
```



```
>plot3d("2*y+cos(2*x)",distance=4,zoom=1.5,angle=3,height=0):
```



## **2. Menyimpan Rumus Fungsi Dua Variabel dalam Variabel Ekspresi**

Apa Itu Grafik Fungsi Dua Variabel?

---

Grafik fungsi dua variabel adalah representasi visual dari hubungan antara sebuah fungsi matematika dengan dua variabel independen (biasanya disebut sebagai "x" dan "y") dan variabel dependen (biasanya disebut sebagai "z" atau "f(x, y)").

Dalam grafik ini, sumbu x dan sumbu y digunakan untuk menggambarkan nilai-nilai dua variabel independen, sementara permukaan atau grafik 3D digunakan untuk menggambarkan nilai-nilai variabel dependen yang dihasilkan oleh fungsi tersebut.

Grafik fungsi dua variabel membantu memvisualisasikan bagaimana nilai variabel dependen (z) berubah seiring perubahan kedua variabel independen (x dan y) sesuai dengan aturan fungsi tersebut

Fungsi matematika yg terlibat dalam Menggambar Grafik Fungsi Dua Variabel

---

### 1. Fungsi Linear

Bentuk umum

$f(x, y) = ax + by + c$ , di mana a, b, dan c adalah konstanta. Grafiknya adalah bidang datar.

### 2. Fungsi Kuadratik

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

Grafiknya dapat berupa permukaan yang berbentuk paraboloid, baik terbuka ke atas atau ke bawah.

### 3. Fungsi Trigonometri

**sinus dan kosinus**

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

kan menghasilkan permukaan yang berulang-ulang naik dan turun.

#### 4.Fungsi Pecahan

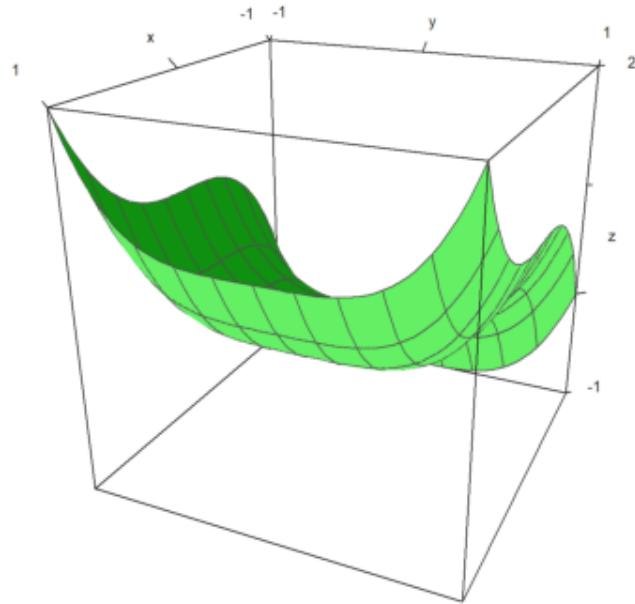
Bentuk umum  $f(x, y) = g(x, y) / h(x, y)$ , di mana  $g(x, y)$  dan  $h(x, y)$  adalah fungsi-fungsi lain. rafiknya dapat menghasilkan berbagai pola yang tergantung pada sifat fungsi  $g$  dan  $h$ .

#### CONTOH SOAL 1

---

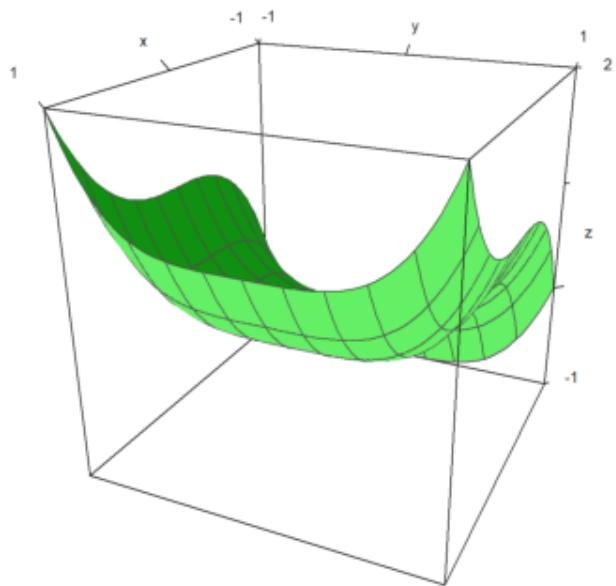
$$f(x, y) = x^3 + y^4$$

```
>function a(x,y):= x^3+y^4  
>plot3d("a"):
```

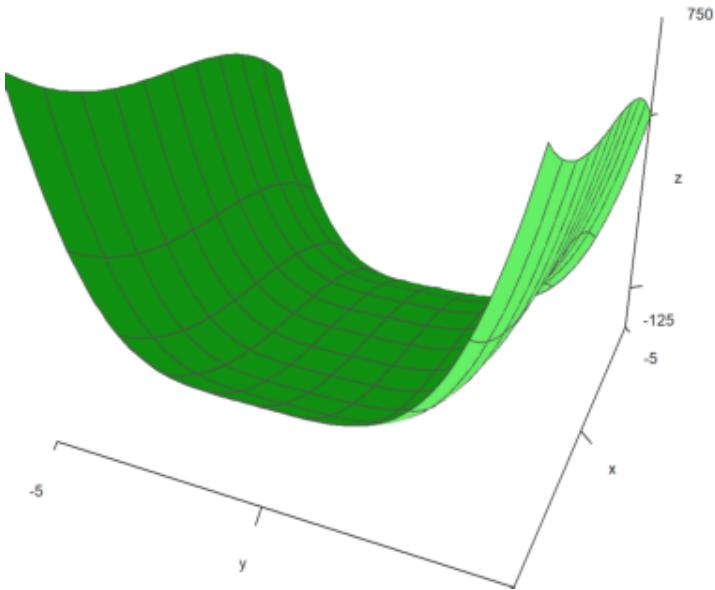


```
>plot3d("a",>user, ...
> title="Turn with the vector keys (press return to finish)":
```

Turn with the vector keys (press return to finish)



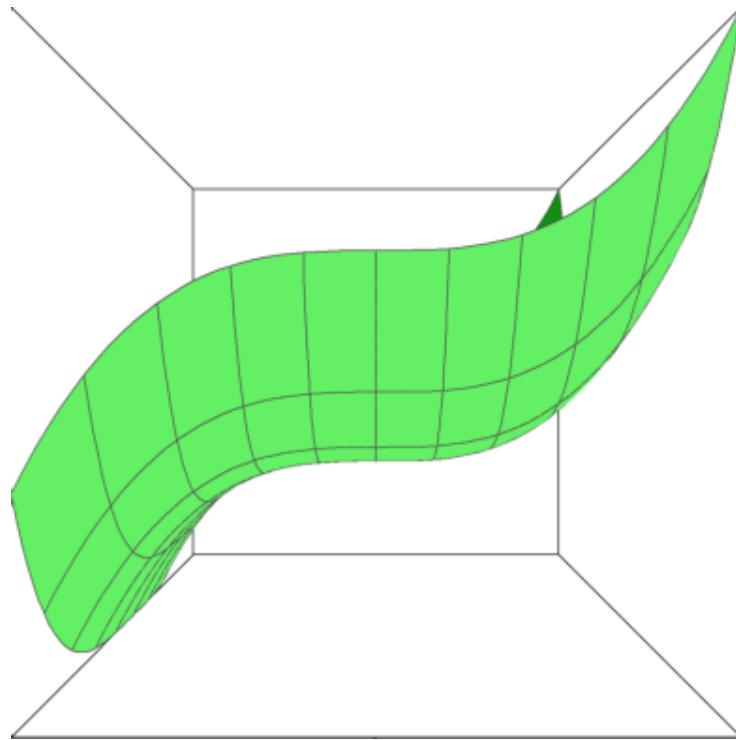
```
>plot3d("a",r=5,n=80,fscale=4,scale=1.2,frame=3):
```



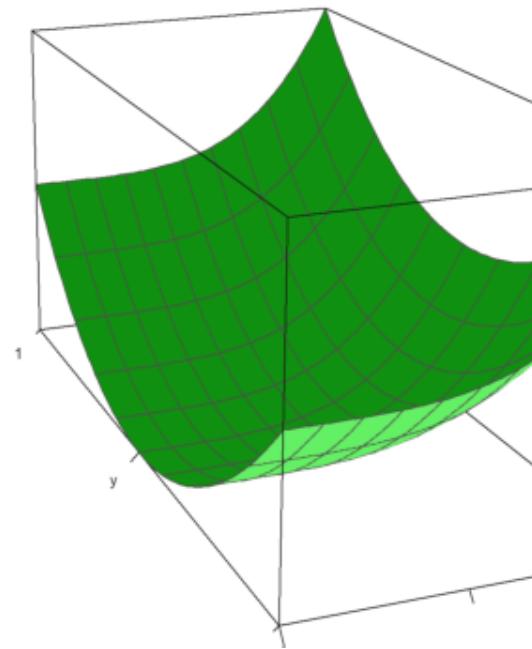
```
>view
```

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

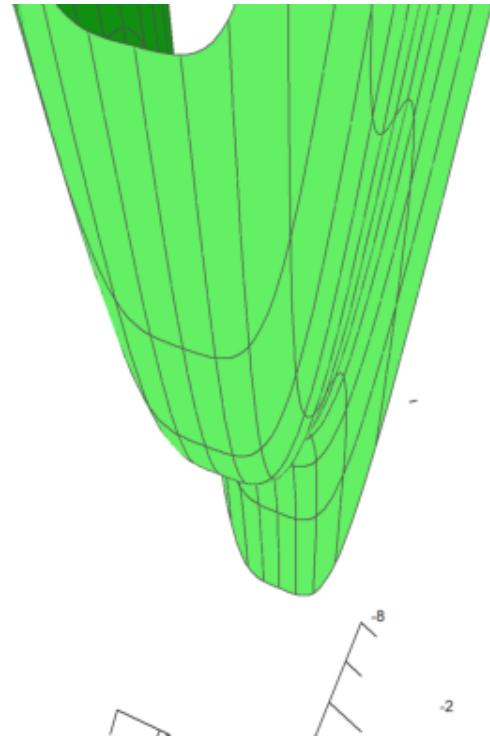
```
>plot3d("a",distance=3,zoom=2,angle=0,height=0):
```



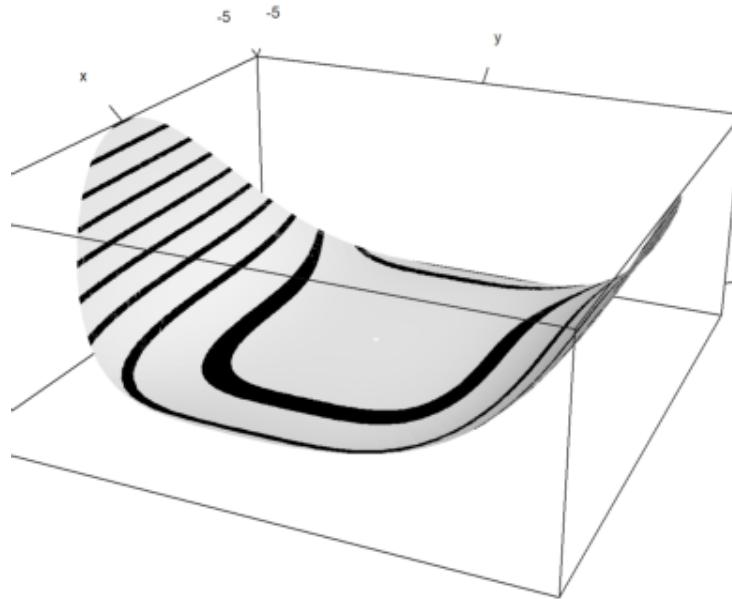
```
>plot3d("x^4+y^2",a=0,b=1,c=-1,d=1,angle=-20°,height=20°, ...
>center=[0.4,0,0],zoom=5):
```



```
>plot3d("a",r=2,<fscale,<scale,distance=13,height=50°, ...
> center=[0,0,-2],frame=3):
```



```
>plot3d("a",r=5,>polar, ...
>fscale=2,>hue,n=100,zoom=4,>contour,color=gray):
```



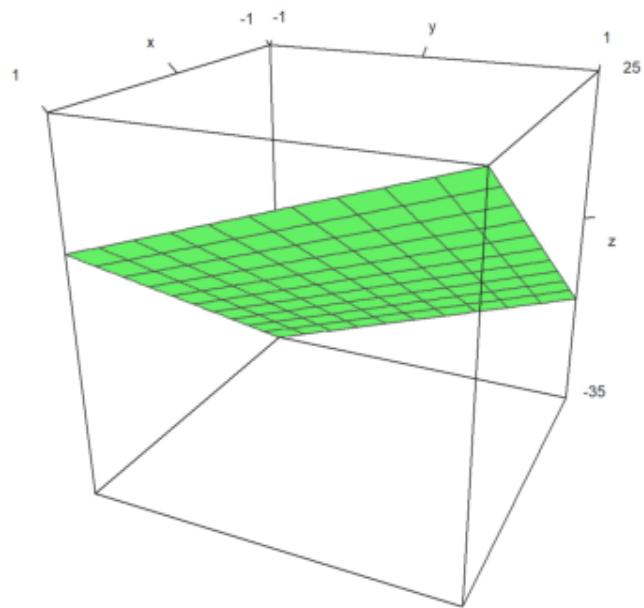
```
>plot3d("x", "a", "y", r=2, zoom=3.5, frame=3):
```



FUNGSI LINEAR

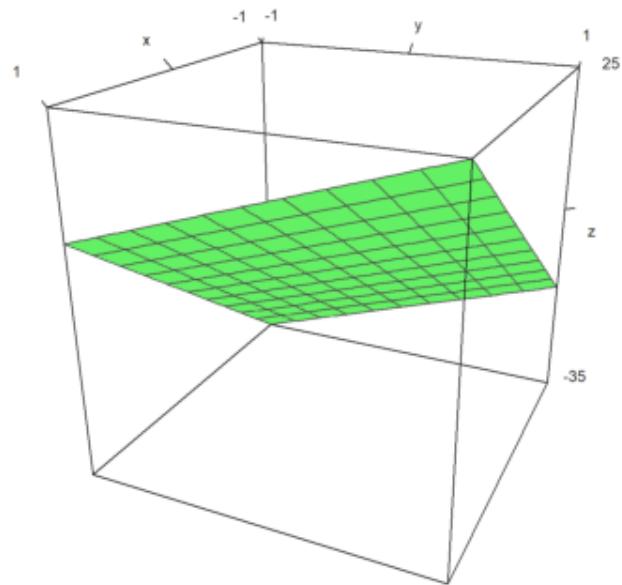
---

```
>function e(x,y):= 20x+10y-5  
>plot3d("e"):
```

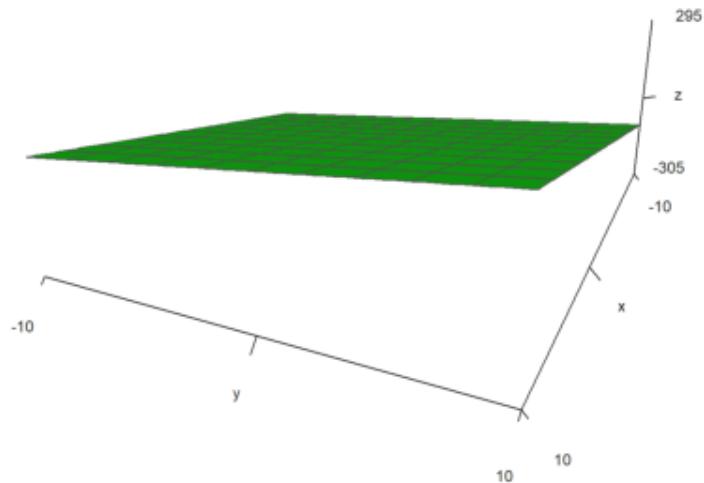


```
>plot3d("e",>user, ...
>title="Turn with the vector keys (press return to finish)":
```

Turn with the vector keys (press return to finish)



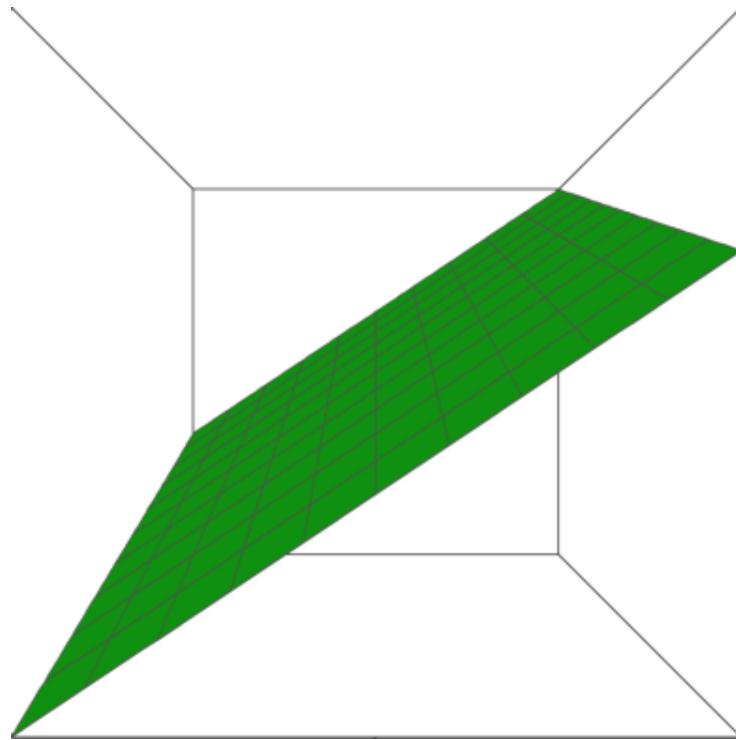
```
>plot3d("e",r=10,n=80,fscale=4,scale=1.2,frame=3):
```



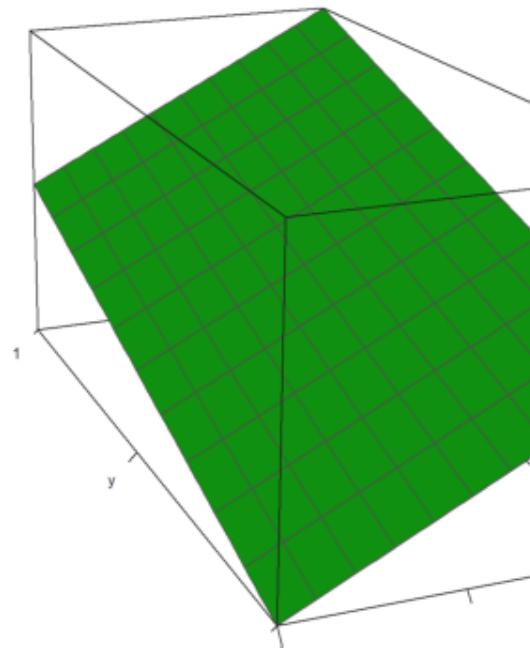
```
>view
```

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

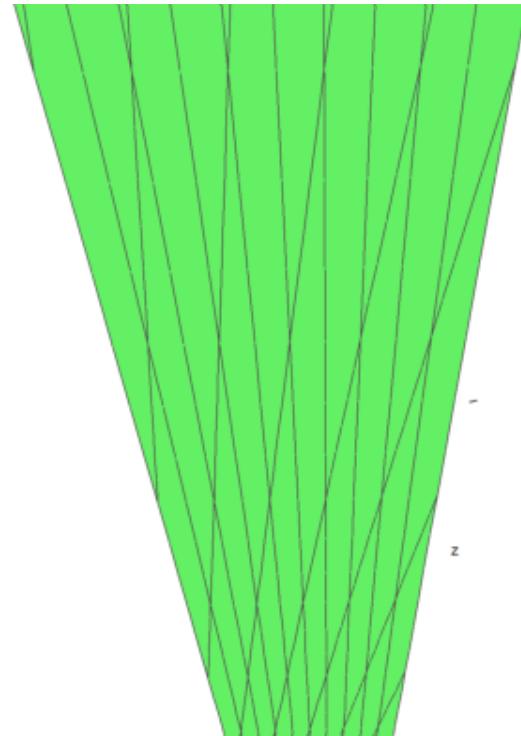
```
>plot3d("e",distance=3,zoom=2,angle=0,height=0):
```



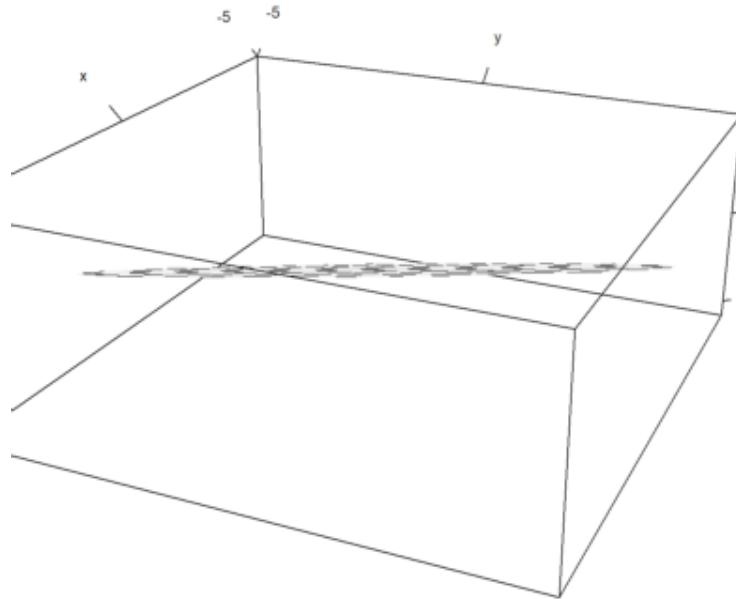
```
>plot3d("e",a=0,b=1,c=-1,d=1,angle=-20°,height=20°, ...
>center=[0.4,0,0],zoom=5):
```



```
>plot3d("e",r=2,<fscale,<scale,distance=13,height=50°, ...
>center=[0,0,-2],frame=3):
```



```
>plot3d("e",r=5,>polar, ...
>fscale=2,>hue,n=100,zoom=4,>contour,color=gray):
```



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

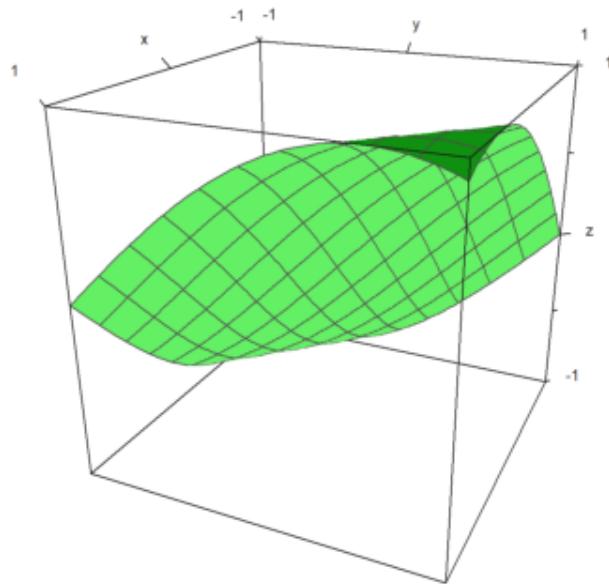


FUNGSI TRIGONOMETRI

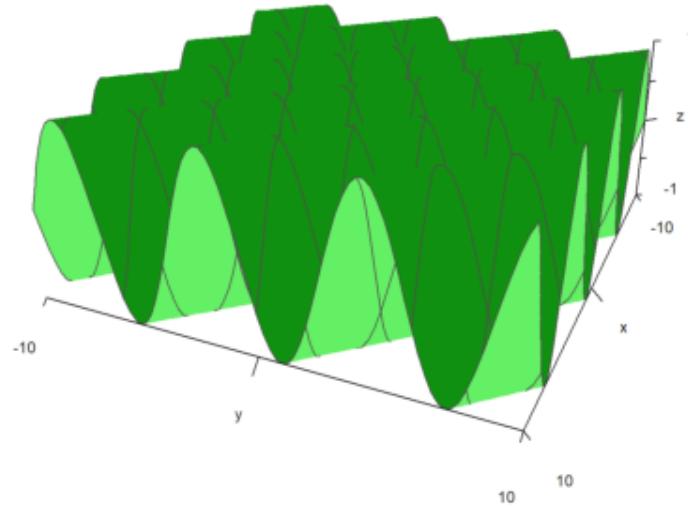
---

```
>function f(x,y):= sin(x+y)
>plot3d("f")
>plot3d("f",>user, ...
>title="Turn with the vector keys (press return to finish)":
```

Turn with the vector keys (press return to finish)



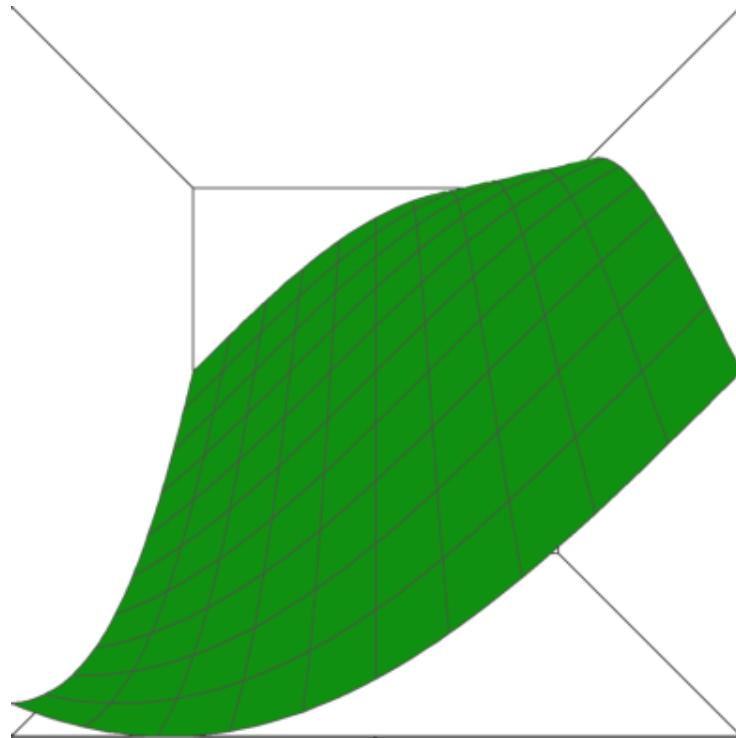
```
>plot3d("f",r=10,n=80,fscale=4,scale=1.2,frame=3):
```



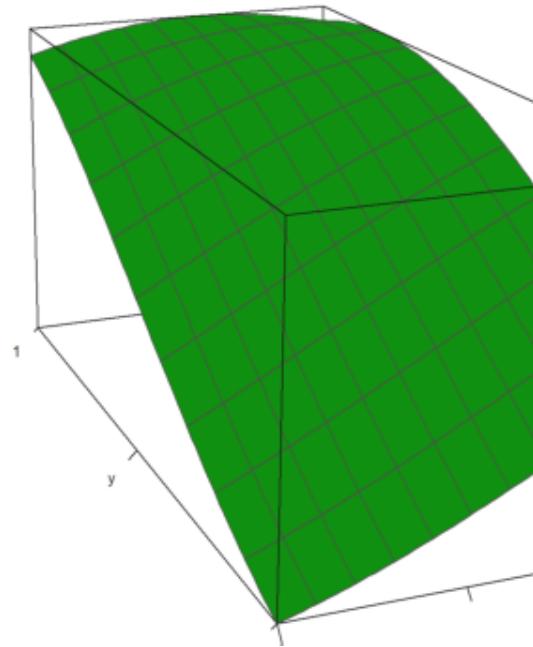
```
>view
```

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

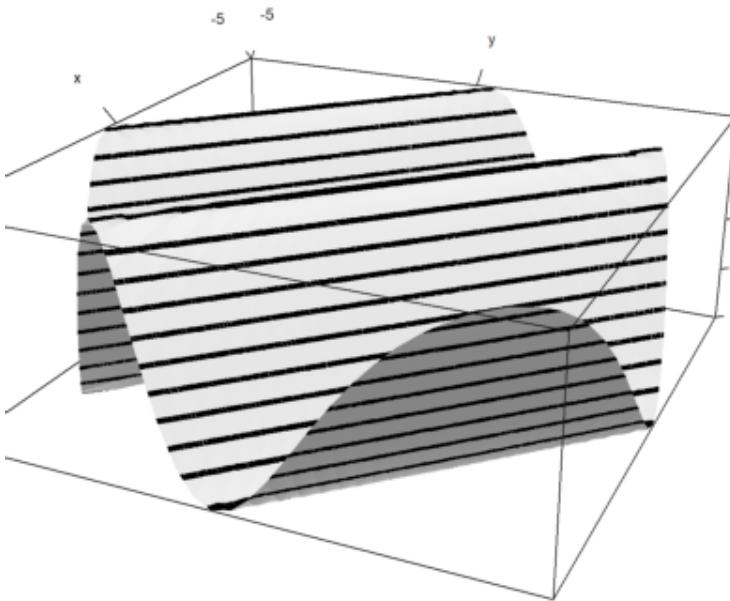
```
>plot3d("f",distance=3,zoom=2,angle=0,height=0):
```



```
>plot3d("f",a=0,b=1,c=-1,d=1,angle=-20°,height=20°, ...  
>center=[0.4,0,0],zoom=5):
```



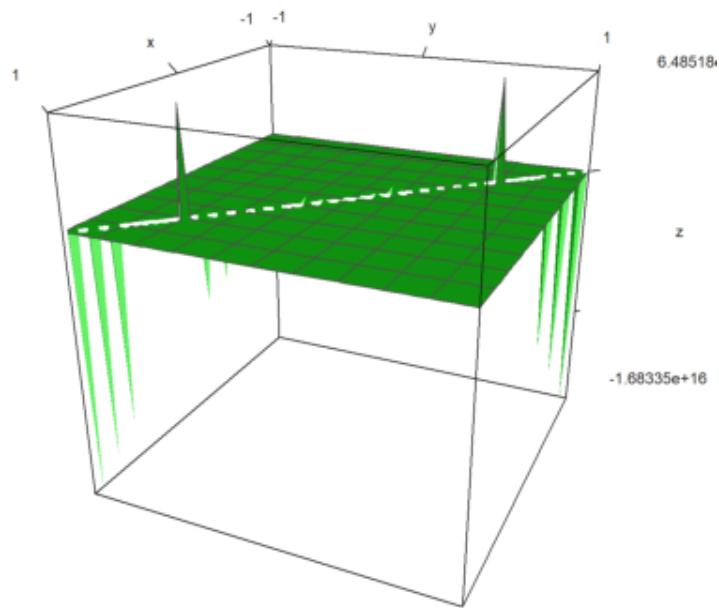
```
>plot3d("f",r=5,>polar, ...
>fscale=2,>hue,n=100,zoom=4,>contour,color=gray):
```



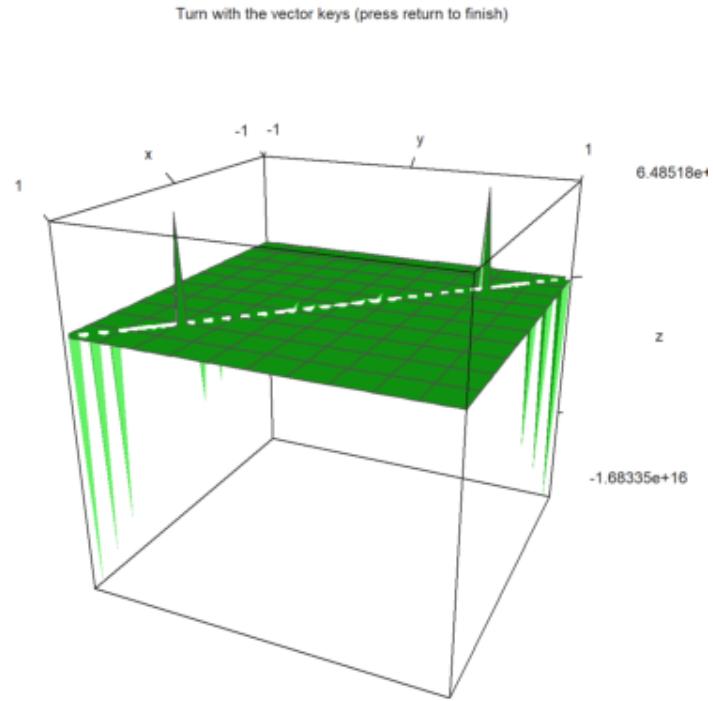
FUNGSI PECAHAN

---

```
>function h(x,y):= (x^2 + y^2) / (x + y)
>plot3d("h"):
```



```
>plot3d("h",>user, ...
>title="Turn with the vector keys (press return to finish)":
```



---

### 3. Menggambar Grafik Fungsi Dua Variabel yang Fungsinya

### Didefinisikan sebagai Fungsi Numerik

Fungsi numerik adalah sebuah fungsi dengan himpunan bilangan cacah sebagai domain dan himpunan bilangan real sebagai kodomain. Fungsi numerik merupakan konsep matematika yang mendasar yang melibatkan hubungan matematis antara bilangan yang menjadi domain dan bilangan sebagai kodomain.

Fungsi numerik memiliki 1 atau lebih variabel terikat,dalam pembahasan ini hanya 2 variabel terikat yang sering dilambangkan sebagai "X" dan "Y". Variabel X dan Y adalah

nilai atau parameter yang dapat berubah, dan fungsi numerik menggambarkan bagaimana variabel ini memengaruhi variabel dependen. Variabel dependen adalah hasil perhitungan atau keluaran dari fungsi numerik yang bergantung pada nilai atau perubahan dalam variabel independen.

Dalam euler math toolbox cara mendefinisikan fungsi menggunakan sintak function. untuk mendefinisikan fungsi numerik menggunakan tanda " $:=$ ".Fungsi numerik menjelaskan cara bilangan dalam domain berhubungan dengan bilangan sebagai kodomain, biasanya diberikan dalam bentuk rumus matematik(persamaan) atau aturan yang memetakan setiap domain kedalam kodomain yang sesuai.contoh:

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

(x)(variabel terikat) adalah fungsi yang memetakan setiap nilai x(variabel independen)kedalam nilai  $2x+1$ . Terdapat berbagai jenis fungsi yang termasuk ke dalam fungsi numerik, diantaranya berikut bentuk-bentuk umumnya :

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

$$f(x, y) = ax^2 + bx + c + dy^2 + ey + f$$

$$f(x, y) = a^x + y$$

$$f(x, y) = \log_a(x + y)$$

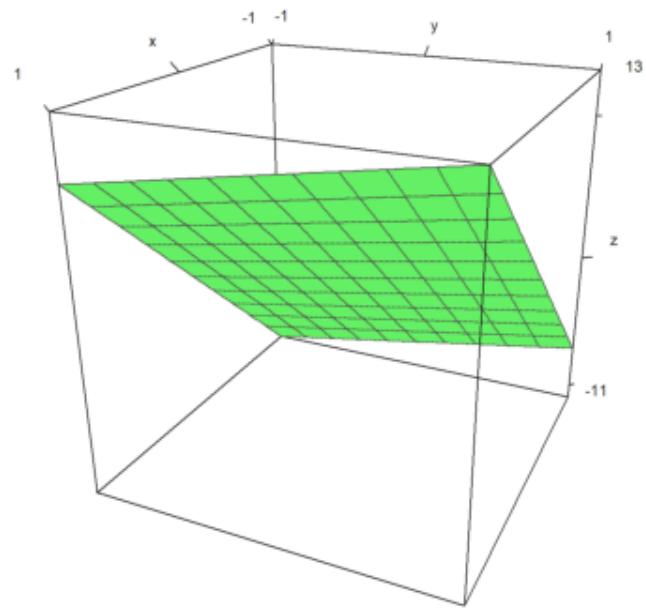
dan lain-lain.

Salah satu cara yang umum digunakan untuk memvisualisasikan fungsi numerik adalah dengan menggambarkan grafiknya. Grafik ini menggambarkan bagaimana variabel dependen berubah seiring perubahan variabel independen dan membantu dalam memahami sifat-sifat fungsi, seperti titik ekstrem Dll.

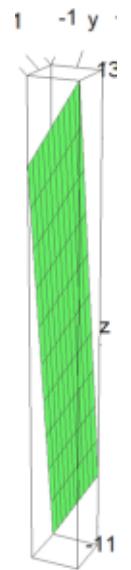
Contoh Soal

---

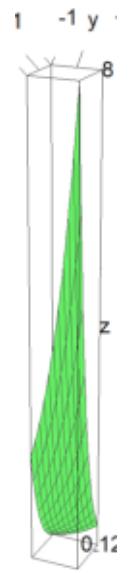
```
>function f(x,y):= 10*x+2*y+1  
>plot3d("f"):
```



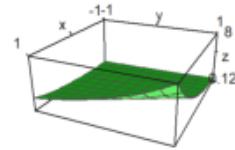
```
>aspect(0.1); plot3d("f"):
```



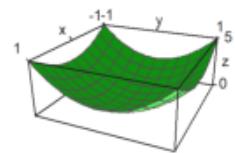
```
>function g(x,y):= 2^(2*x+y)
>plot3d("g"):
```



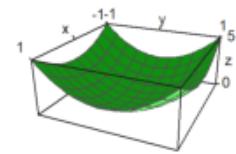
```
>aspect(2.5); plot3d("g"):
```



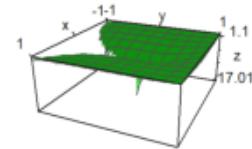
```
>function h(x,y):= 2*x^2+3*y^2  
>plot3d("h"):
```



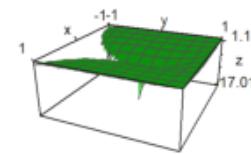
```
>aspect(2.5); plot3d("h"):
```



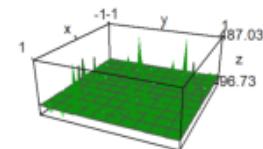
```
>function i(x,y):= log(2*x^2+y^5)
>plot3d("i"):
```



```
>aspect(2.5); plot3d("i"):
```



```
>function j(x,y):= sin(x^2+y)+sec(100*x+y)
>plot3d("j"):
```



```
>aspect(4.5); plot3d("j"):
```



## 4. Menggambar Grafik Dua Variabel yang Fungsinya Didefinisikan

---

sebagai Fungsi Simbolik

Fungsi simbolik adalah fungsi matematika yang dinyatakan dengan menggunakan simbol-simbol aljabar atau simbol-simbol matematika pada umumnya, tanpa menyertakan nilai numerik spesifik. Dengan kata lain, variabel-variabel dan operasi-operasi matematika diwakili oleh simbol-simbol aljabar, sehingga ekspresi fungsi tersebut tetap bersifat umum dan tidak bergantung pada nilai-nilai numerik tertentu.

Contoh sederhana dari fungsi simbolik adalah

$$f(x) = x^2$$

di mana  $x$  adalah variabel simbolik dan  $f(x)$  adalah fungsi kuadrat umum. Sebagai perbandingan, jika kita memiliki nilai spesifik untuk  $x$ , seperti  $x=3$ , kita dapat menggantikan  $x$  dalam fungsi tersebut dan mendapatkan nilai numerik spesifik

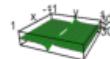
$$f(3) = 3^2 = 9.$$

Fungsi simbolik sering digunakan dalam berbagai bidang matematika, fisika, dan komputasi simbolik. Mereka memungkinkan manipulasi matematika yang umum, seperti penyederhanaan ekspresi, integrasi, dan diferensiasi, tanpa harus menetapkan nilai numerik pada awalnya. Pemrosesan simbolik ini dapat dilakukan menggunakan perangkat lunak atau pustaka yang mendukung komputasi simbolik, seperti SymPy dalam Python.

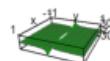
Contoh Soal

---

```
>function f(x,y) := x^2/y; // define a function  
>plot3d("f"):
```



```
>plot3d("f"):
```

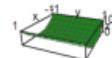


```
>plot3d({{"f",0.2}},0,1,2)
```

Function f needs only 2 arguments (got 3)!  
Use: f (x, y)  
%ploteval2:  
if maps then return %mapexpression2(x,y,f\$;args());

```
f3dplot:  
      z=%ploteval2(f$,x,y,maps;args());  
Try "trace errors" to inspect local variables after errors.  
plot3d:  
=cp,=cplevel,=cpcolor,=cpdelta);
```

```
>plot3d({{"f(x,b)",b=0.1}},0,1):
```



```
>function f(x) := x^3+y^2y;  
>function f(x,y) &= diff(x^2+3*y,x,y)
```

$$\frac{d^2}{dx^2} (3y^2 + x^3)$$

```
>expr &= sin(x)+cos(y)*exp(-x)
```

$$E = -x \cos(y) + \sin(x)$$

```
>plot3d(expr,0,2,3pi):
```



## 5. Menggambar Data $x, y, z$ pada ruang Tiga Dimensi (3D)

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

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

Karena x,y,z adalah matriks, kita asumsikan bahwa (t,s) melewati grid persegi. Hasilnya membawa kita untuk bisa memplot gambar .

```
>splot3d("x*y"):
```

```
Function splot3d not found.  
Try list ... to find functions!  
Error in:  
splot3d("x*y"): ...  
^
```

Penjelasan sintaks dari plot

- plot3d = membawa euler untuk mengetahui perintah apa yang harus dilakukan
- (" ...") = tempat kita untuk memasukkan perintah yang kita inginkan

## CONTOH

---

1. Kita akan membuat plot untuk

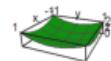
$$x^2 + y^2$$

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



Selanjutnya, jika kita ingin menambahkan garis pada plot bisa menggunakan grid

```
>plot3d("x^2+y^2",grid=4):
```



Jika kita ingin memodifikasi plot dengan menambahkan warna pada plot, bisa menggunakan fillcolor. Fillcolor dapat diisi dengan 1 warna yang sama atau 2 warna yang berbeda

```
>plot3d("x^2+y^2",grid= 3,fillcolor=[black,green]):
```



Selanjutnya, jika kita ingin menetapkan batasan untuk grafik kita. Kita dapat menambahkan perintah xmax, xmin, ymax, dan ymin.

```
>plot3d("x^2+y^2",xmin=-1,xmax=3,ymin=-1,ymax=2,>user):
```



## CONTOH

---

2. Selanjutnya, diberikan contoh ketika kita memiliki nilai x, y, dan z. Kemudian kita ingin memvisualisasikannya pada plot 3D.

```
>x=-1:0.1:1
```

```
[-1, -0.9, -0.8, -0.7, -0.6, -0.5, -0.4, -0.3, -0.2, -0.1, 0,  
0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]
```

```
>y=x'
```

```
Real 21 x 1 matrix
```

```
-1  
-0.9  
-0.8  
-0.7  
-0.6  
-0.5  
-0.4  
-0.3  
-0.2  
-0.1  
0  
0.1  
0.2
```

0.3  
0.4  
0.5  
0.6  
0.7  
0.8  
0.9

...

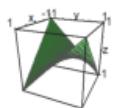
>z=x\*y

Real 21 x 21 matrix

1	0.9	0.8	0.7	...
0.9	0.81	0.72	0.63	...
0.8	0.72	0.64	0.56	...
0.7	0.63	0.56	0.49	...
0.6	0.54	0.48	0.42	...
0.5	0.45	0.4	0.35	...
0.4	0.36	0.32	0.28	...
0.3	0.27	0.24	0.21	...
0.2	0.18	0.16	0.14	...
0.1	0.09	0.08	0.07	...
0	0	0	0	...
-0.1	-0.09	-0.08	-0.07	...
-0.2	-0.18	-0.16	-0.14	...
-0.3	-0.27	-0.24	-0.21	...
-0.4	-0.36	-0.32	-0.28	...
-0.5	-0.45	-0.4	-0.35	...
-0.6	-0.54	-0.48	-0.42	...
-0.7	-0.63	-0.56	-0.49	...
-0.8	-0.72	-0.64	-0.56	...
-0.9	-0.81	-0.72	-0.63	...

• • •

```
>plot3d(x,y,z, >user):
```



## CONTOH

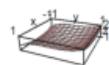
---

3. Ketika kita ingin membuat plot 3D pada fungsi

$$x^2 + y^3$$

dengan hasil warna yang seperti ini, kita bisa menggunakan perintah ini.

```
>plot3d("x^2+y^3",grid=10,>hue, color=red, >user):
```



Perintah ini juga bisa digunakan, tetapi warna yang dihasilkan nantinya akan berbeda

```
>plot3d("x^2+y^3",grid=10,fillcolor=[red,red], >user):
```



## CONTOH

---

4. Pada contoh keempat, kita menggunakan vektor nilai t dan vektor kolom nilai s untuk membuat parameter permukaan bola. Dalam gambar kita dapat menandai wilayah, dalam kasus kita wilayah polar/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=red,<frame,grid=[10,30], ...
>values=s,contourcolor=red,level=[90°-24°;90°-22°], ...
>scale=1.2,height=50°):
```



```
>t=linspace(0,2pi,180)
```

```
[0, 0.0349066, 0.0698132, 0.10472, 0.139626, 0.174533, 0.20944,
0.244346, 0.279253, 0.314159, 0.349066, 0.383972, 0.418879,
0.453786, 0.488692, 0.523599, 0.558505, 0.593412, 0.628319,
0.663225, 0.698132, 0.733038, 0.767945, 0.802851, 0.837758,
0.872665, 0.907571, 0.942478, 0.977384, 1.01229, 1.0472, 1.0821,
1.11701, 1.15192, 1.18682, 1.22173, 1.25664, 1.29154, 1.32645,
```

```

1.36136,  1.39626,  1.43117,  1.46608,  1.50098,  1.53589,  1.5708,
1.6057,  1.64061,  1.67552,  1.71042,  1.74533,  1.78024,  1.81514,
1.85005,  1.88496,  1.91986,  1.95477,  1.98968,  2.02458,  2.05949,
2.0944,  2.1293,  2.16421,  2.19911,  2.23402,  2.26893,  2.30383,
2.33874,  2.37365,  2.40855,  2.44346,  2.47837,  2.51327,  2.54818,
2.58309,  2.61799,  2.6529,  2.68781,  2.72271,  2.75762,  2.79253,
2.82743,  2.86234,  2.89725,  2.93215,  2.96706,  3.00197,  3.03687,
3.07178,  3.10669,  3.14159,  3.1765,  3.21141,  3.24631,  3.28122,
3.31613,  3.35103,  3.38594,  3.42085,  3.45575,  3.49066,  3.52557,
3.56047,  3.59538,  3.63028,  3.66519,  3.7001,  3.735,  3.76991,
3.80482,  3.83972,  3.87463,  3.90954,  3.94444,  3.97935,  4.01426,
4.04916,  4.08407,  4.11898,  4.15388,  4.18879,  4.2237,  4.2586,
4.29351,  4.32842,  4.36332,  4.39823,  4.43314,  4.46804,  4.50295,
4.53786,  4.57276,  4.60767,  4.64258,  4.67748,  4.71239,  4.7473,
...
]

```

```
>s=linspace(-pi/2,pi/2,90)
```

```

[-1.5708, -1.53589, -1.50098, -1.46608, -1.43117, -1.39626,
-1.36136, -1.32645, -1.29154, -1.25664, -1.22173, -1.18682,
-1.15192, -1.11701, -1.0821, -1.0472, -1.01229, -0.977384,
-0.942478, -0.907571, -0.872665, -0.837758, -0.802851, -0.767945,
-0.733038, -0.698132, -0.663225, -0.628319, -0.593412, -0.558505,
-0.523599, -0.488692, -0.453786, -0.418879, -0.383972, -0.349066,
-0.314159, -0.279253, -0.244346, -0.20944, -0.174533, -0.139626,
-0.10472, -0.0698132, -0.0349066, 0, 0.0349066, 0.0698132,
0.10472, 0.139626, 0.174533, 0.20944, 0.244346, 0.279253,
0.314159, 0.349066, 0.383972, 0.418879, 0.453786, 0.488692,
0.523599, 0.558505, 0.593412, 0.628319, 0.663225, 0.698132,
0.733038, 0.767945, 0.802851, 0.837758, 0.872665, 0.907571,
0.942478, 0.977384, 1.01229, 1.0472, 1.0821, 1.11701, 1.15192,
1.18682, 1.22173, 1.25664, 1.29154, 1.32645, 1.36136, 1.39626,
1.43117, 1.46608, 1.50098, 1.53589, 1.5708]

```

## CONTOH

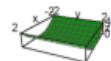
---

5. Selanjutnya, kita akan membuat plot dengan fungsi

$$x^2$$

dan dilengkapi dengan label atau keterangan

```
>plot3d("x^2",r=2, >user); label(["uri","(0,-1)"],0,-1):
```



## CONTOH

---

6. Selanjutnya, ketika kita mau memplotkan data yang kita punya.

Perintah hue dan light untuk mengatur pewarnaan dari plot kita

Perintah frame = 0 menunjukkan bahwa plot kita sebenarnya berada di dalam bangun ruang tetapi karena angkanya 0. jadi tidak terlihat. Zoom menunjukkan perbesaran dari plot data kita.

```
>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, >user):
```



```
>t=linspace(0,2pi,320)
```

```
[0,  0.019635,  0.0392699,  0.0589049,  0.0785398,  0.0981748,
0.11781,  0.137445,  0.15708,  0.176715,  0.19635,  0.215984,
0.235619,  0.255254,  0.274889,  0.294524,  0.314159,  0.333794,
0.353429,  0.373064,  0.392699,  0.412334,  0.431969,  0.451604,
0.471239,  0.490874,  0.510509,  0.530144,  0.549779,  0.569414,
0.589049,  0.608684,  0.628319,  0.647953,  0.667588,  0.687223,
0.706858,  0.726493,  0.746128,  0.765763,  0.785398,  0.805033,
```

```

0.824668,  0.844303,  0.863938,  0.883573,  0.903208,  0.922843,
0.942478,  0.962113,  0.981748,  1.00138,  1.02102,  1.04065,
1.06029,  1.07992,  1.09956,  1.11919,  1.13883,  1.15846,  1.1781,
1.19773,  1.21737,  1.237,  1.25664,  1.27627,  1.29591,  1.31554,
1.33518,  1.35481,  1.37445,  1.39408,  1.41372,  1.43335,  1.45299,
1.47262,  1.49226,  1.51189,  1.53153,  1.55116,  1.5708,  1.59043,
1.61007,  1.6297,  1.64934,  1.66897,  1.68861,  1.70824,  1.72788,
1.74751,  1.76715,  1.78678,  1.80642,  1.82605,  1.84569,  1.86532,
1.88496,  1.90459,  1.92423,  1.94386,  1.9635,  1.98313,  2.00277,
2.0224,  2.04204,  2.06167,  2.08131,  2.10094,  2.12058,  2.14021,
2.15984,  2.17948,  2.19911,  2.21875,  2.23838,  2.25802,  2.27765,
2.29729,  2.31692,  2.33656,  2.35619,  2.37583,  2.39546,  2.4151,
2.43473,  2.45437,  2.474,  2.49364,  2.51327,  2.53291,  2.55254,
... ]

```

```
>s=linspace(-pi/2,pi/2,160)
```

```

[-1.5708, -1.55116, -1.53153, -1.51189, -1.49226, -1.47262,
-1.45299, -1.43335, -1.41372, -1.39408, -1.37445, -1.35481,
-1.33518, -1.31554, -1.29591, -1.27627, -1.25664, -1.237,
-1.21737, -1.19773, -1.1781, -1.15846, -1.13883, -1.11919,
-1.09956, -1.07992, -1.06029, -1.04065, -1.02102, -1.00138,
-0.981748, -0.962113, -0.942478, -0.922843, -0.903208, -0.883573,
-0.863938, -0.844303, -0.824668, -0.805033, -0.785398, -0.765763,
-0.746128, -0.726493, -0.706858, -0.687223, -0.667588, -0.647953,
-0.628319, -0.608684, -0.589049, -0.569414, -0.549779, -0.530144,
-0.510509, -0.490874, -0.471239, -0.451604, -0.431969, -0.412334,
-0.392699, -0.373064, -0.353429, -0.333794, -0.314159, -0.294524,
-0.274889, -0.255254, -0.235619, -0.215984, -0.19635, -0.176715,
-0.15708, -0.137445, -0.11781, -0.0981748, -0.0785398,
-0.0589049, -0.0392699, -0.019635, 0, 0.019635, 0.0392699,
0.0589049, 0.0785398, 0.0981748, 0.11781, 0.137445, 0.15708,
```

```
0.176715,  0.19635,  0.215984,  0.235619,  0.255254,  0.274889,  
0.294524,  0.314159,  0.333794,  0.353429,  0.373064,  0.392699,  
0.412334,  0.431969,  0.451604,  0.471239,  0.490874,  0.510509,  
0.530144,  0.549779,  0.569414,  0.589049,  0.608684,  0.628319,  
0.647953,  0.667588,  0.687223,  0.706858,  0.726493,  0.746128,  
... ]
```

## **6. Membuat Gambar Grafik Tiga Dimensi (3D) 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():
```



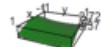
```
>function testplot () := plot3d("x^2+y",distance=3,zoom=1,angle=pi/2,height=0); ...
>rotate("testplot"); testplot():
```



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)

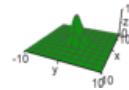


```
>plot3d("exp(x^2+y^2)",>user, ...
>title="Coba gerakan")
```

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. kembali: akhiri interaksi

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



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).

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



Tampilan dapat diubah dengan berbagai cara.

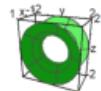
- distance: jarak pandang ke plot.
- zoom: nilai zoom.
- angle: sudut terhadap sumbu y negatif dalam radian.
- height: ketinggian tampilan dalam radian.

```
>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 center parameter.

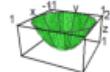
```
>plot3d("x^2+1",a=-1,b=1,rotate=true,grid=5):
```



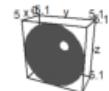
Parameter memutar memutar fungsi dalam x di sekitar sumbu x.

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

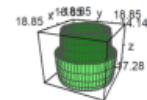
```
>plot3d("x^2+1",a=-1,b=1,rotate=2,grid=5):
```



```
>plot3d("sqrt(x^2+1)",a=0,b=5,rotate=1):
```



```
>plot3d("x*sin(x)",a=0,b=6pi,rotate=2):
```



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

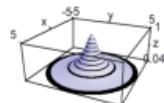
```



```

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

```

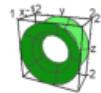


Plot kutub juga tersedia. Parameter polar=true menggambar plot kutub. Fungsi tersebut harus tetap merupakan fungsi dari x dan y. Itu parameter "fscale" menskalakan fungsi dengan skalanya sendiri. Kalau tidak, fungsinya akan diskalakan agar sesuai dengan kubus.

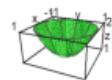
```

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

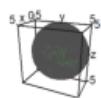
```



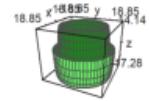
```
>function testplot () := plot3d("x^2+1",a=-1,b=1,rotate=2,grid=5); ...
>rotate("testplot"); testplot():
```



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



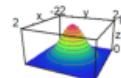
```
>function testplot () := plot3d("x*sin(x)",a=0,b=6pi,rotate=2); ...
>rotate("testplot"); testplot():
```



```
>function testplot () := plot3d("x","x^2+y^2","y",r=2,zoom=3.5,frame=3); ...
>rotate("testplot"); testplot():
```



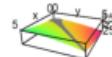
```
>function testplot () := plot3d("exp(-x^2-y^2)",r=2,n=100,level="thin", ...
>>contour,>spectral,fscale=1,scale=1.1,height=20°); ...
>rotate("testplot"); testplot():
```



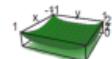
Untuk plotnya, Euler menambahkan garis grid. Sebaliknya dimungkinkan untuk menggunakan garis level dan satu warna atau warna spektral. Euler dapat menggambar ketinggian fungsi pada sebuah plot dengan bayangan. Di semua plot 3D, Euler dapat menghasilkan anaglyph merah/cyan.

- 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=redgreen); ...
>rotate("testplot"); testplot():
```



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



```
>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)



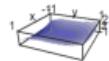
```
>function testplot () := plot3d("-x^2-y^2",color=rgb(0.2,0.2,0),hue=true,frame=false, ...
>zoom=3,contourcolor=red,level=-2:0.1:1,dl=0.01); ...
>rotate("testplot"); testplot():
```



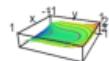
```
>function testplot () := plot3d("x^2+y^3",>contour); ...
>rotate("testplot"); testplot():
```



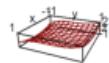
```
>function testplot () := plot3d("x^2+y^3",>contour,color=blue); ...
>rotate("testplot"); testplot():
```



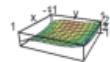
```
>function testplot () := plot3d("x^2+y^3",>contour,>spectral); ...
>rotate("testplot"); testplot():
```



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



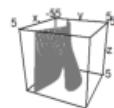
```
>function testplot () := plot3d("x^2+y^3",color=0,hue=true,grid=10); ...
>rotate("testplot"); testplot():
```



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



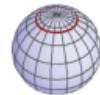
```
>function testplot () := plot3d("x^2+y^3+z*y-1",r=5,implicit=3); ...
>rotate("testplot"); testplot():
```



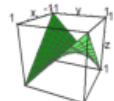
```
>function testplot () := plot3d("x^2+y^2+4*x*z+z^3",>implicit,r=2,zoom=2.5); ...
>rotate("testplot"); testplot():
```



```
>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); ...
>function testplot () := 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°); ...
>rotate("testplot"); testplot();
```



```
>t=-1:0.1:1; s=(-1:0.1:1)'; function testplot () := plot3d(t,s,t*s,grid=10); ...
>rotate("testplot"); testplot();
```



```

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

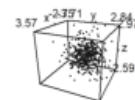
```



```

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

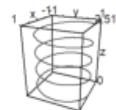
```



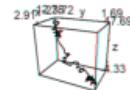
```

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

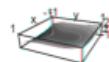
```



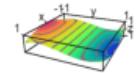
```
>X=cumsum(normal(3,100)); ...
>function testplot () := plot3d(X[1],X[2],X[3],>anaglyph,>wire); ...
>rotate("testplot"); testplot():
```



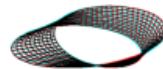
```
>function testplot () := plot3d("x^2+y^3",>anaglyph,>contour); ...
>rotate("testplot"); testplot():
```



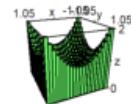
```
>function testplot () := plot3d("x^2*y^3-y",>spectral,>contour,zoom=3.2); ...
>rotate("testplot"); testplot():
```



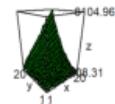
```
>u=linspace(-1,1,10); v=linspace(0,2*pi,50)'; ...
>X=(3+u*cos(v/2))*cos(v); Y=(3+u*cos(v/2))*sin(v); Z=u*sin(v/2); ...
>function testplot () := plot3d(X,Y,Z,>anaglyph,<frame,>wire,scale=2.3); ...
>rotate("testplot"); testplot():
```



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



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



```
>function f(x) := x^3-x+1; ...
>x=-1:0.01:1; t=linspace(0,2pi,50)';
>Z=x; X=cos(t)*f(x); Y=sin(t)*f(x); ...
>pov3d(X,Y,Z,angle=40°,look=povlook(red,0.1),height=50°,axis=0,zoom=4,light=[10,5,15]);
```

Function povlook not found.  
Try list ... to find functions!  
Error in:  
... f(x); pov3d(X,Y,Z,angle=40°,look=povlook(red,0.1),height=50°,a ...  
^

---

## 7. Menggambar Fungsi Parametrik Tiga Dimensi (3D)

Dikutip dari [https://id.wikipedia.org/wiki/Persamaan\\_parametrik](https://id.wikipedia.org/wiki/Persamaan_parametrik), persamaan parametrik merupakan sekelompok kuantitas sebagai fungsi dari satu atau lebih variabel independen yang disebut parameter. Persamaan parametrik biasanya digunakan untuk menyatakan koordinat dari titik-titik yang membentuk objek geometris seperti kurva atau permukaan, dalam hal ini persamaan representasi parametrik atau parameterization.

Sebuah parameter permukaan 3D dapat dikerjakan dengan tiga fungsi atau ekspresi dalam x dan y, atau dengan tiga matriks. Permukaan dimodelkan dengan pemetaan

$$f : Q \rightarrow R^3$$

dimana Q adalah persegi panjang pada bidang

$$R^2$$

```
>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=[1,0,1],<frame, ...
>n=90,grid=[18,36],wirecolor=darkgray,zoom=3):
```



Ekspresi dalam masukkan ini didefinisikan sebagai sebuah bola. Fungsi parameteranya diekspresikan dalam x dan y. Tentu saja fungsi dalam bentuk  $f(x; y)$  juga dapat digunakan.

Perintah allwindow menampilkan layar penuh untuk plot. Kalau tidak, plotnya menyisakan ruang untuk judul plot. Untuk plot tanpa grid seringkali lebih baik untuk mengatur zoomnya secara manual. Zoomnya cukup diatur sehingga bola memenuhi gambar. Untuk persegi panjang, parameteranya ditentukan oleh a, b, c, c.

Untuk plotnya, akan menggunakan model berbayang dengan sumber cahaya tertentu. Frame dibuat tidak utuh. Kemudian, akan dibuat 18 garis grid untuk garis lintang dan 36 untuk garis garis bujur. Jumlah garis grid harus merupakan pembagi jumlah permukaan elemen. Kedua benda tersebut dapat berbeda pada tiap arah persegi panjang.

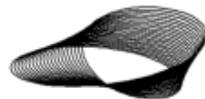
Plot berikut adalah strip Mobius. Kemudian akan dihitung tiga matriks untuk ketiga parameter tersebut. Parameter hue=2 mengaktifkan bayangan, namun hue/warna tidak bergantung pada sisi permukaannya. Nilai max menentukan kegelapan maksimal.

```
>aspect(16/9); allwindow; ...
> x:=linspace(0,2*pi,100); y:=(-1:0.1:1)'; ...
>plot3d(cos(x)*(1+y/2*cos(x/2)),sin(x)*(1+y/2*cos(x/2)),y/2*sin(x/2), ...
><frame,hue=2,max=0.9,scale=2.7):
```



Kurva 3D digambar dengan tiga fungsi parameter dan parameter opsional  
>lines. Jika mengaktifkan interaksi dengan >user, maka plotnya menunjukkan heliks.

```
>x:=linspace(0,2*pi,100); y:=(-1:0.1:1)'; ...
>plot3d(cos(x)*(1+y/2*cos(x/2)),sin(x)*(1+y/2*cos(x/2)),y/2*sin(x/2), ...
>>lines,<frame,xmin=0,xmax=10pi,n=100):
```

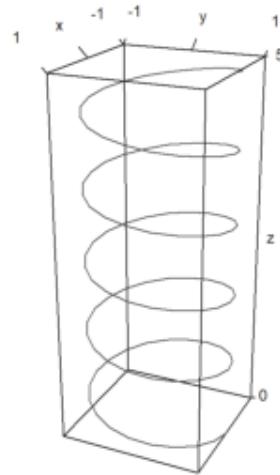


```
>x:=linspace(0,2*pi,100); y:=(-1:0.1:1)'; ...
>plot3d(cos(x)*(1+y/2*cos(x/2)),sin(x)*(1+y/2*cos(x/2)),y/2*sin(x/2), ...
>>lines,<frame,xmin=0,xmax=100,n=100,>user):
```



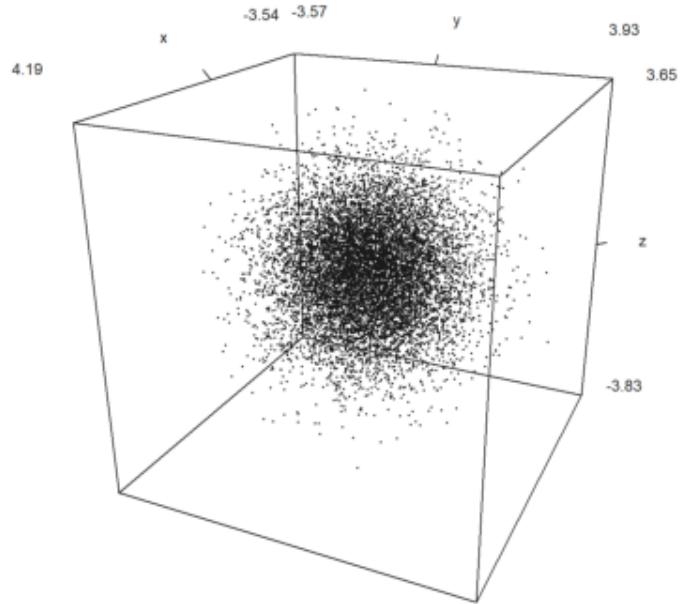
Sebagai alternatif, tiga vektor atau matriks dapat digunakan. Jika `>lines` disetel, plot yang terjadi hanya menampilkan garis dalam satu arah. Koordinatnya kemudian berada di baris matriks. Ini strip Möbius lagi dengan garis saja.

```
>reset; ...
>plot3d("sin(x)","cos(x)","x/2Pi",>lines,xmin=0,xmax=10pi,n=100,>user):
```



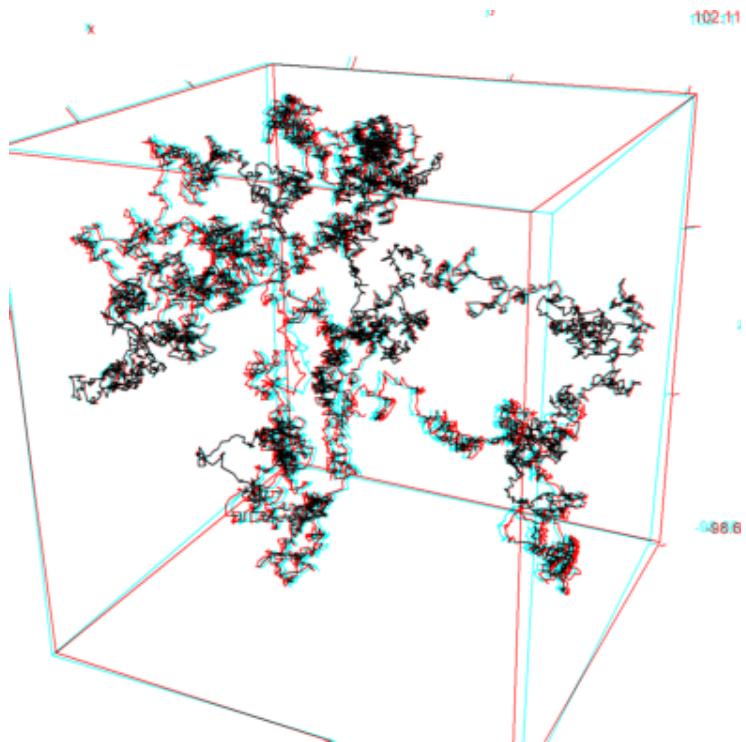
Titik dalam 3D digambar dengan parameter `>points`. Diperlukan tiga vektor untuk koordinat titik-titik tersebut. Perintah berikut menghasilkan cloud secara acak dari nilai terdistribusi normal dalam ruang.

```
>A:=normal(3,10000); plot3d(A[1],A[2],A[3],>points,style="."):
```



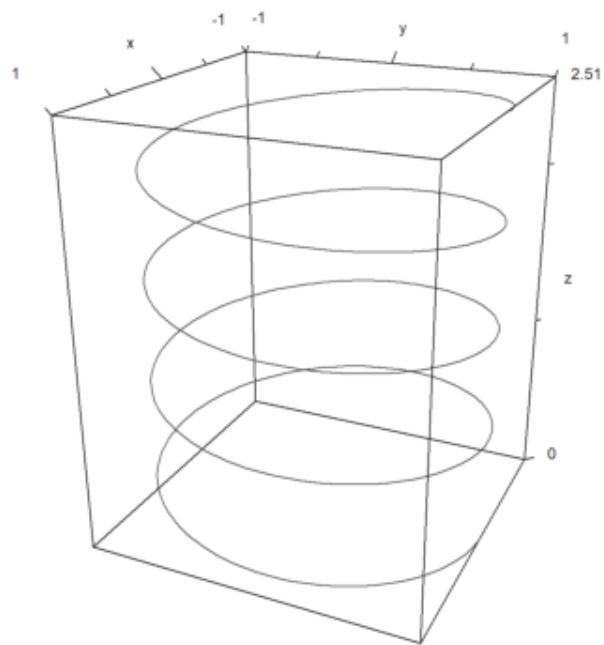
Plot berikut menunjukkan Brownian motion/gerak Brown yang diperoleh dengan jumlah kumulatif koordinatnya. Tampilan dari plot tersebut adalah sebagai plot anaglyph. Dalam hal ini, dibutuhkan kacamata berwarna merah/cyan untuk melihat efek 3D yang menakjubkan

```
>A:=normal(3,10000); B:=cumsum(normal(3,10000)); ...
>plot3d(B[1],B[2],B[3],>wire, ...
>lineWidth=1,>anaglyph,zoom=3.5):
```

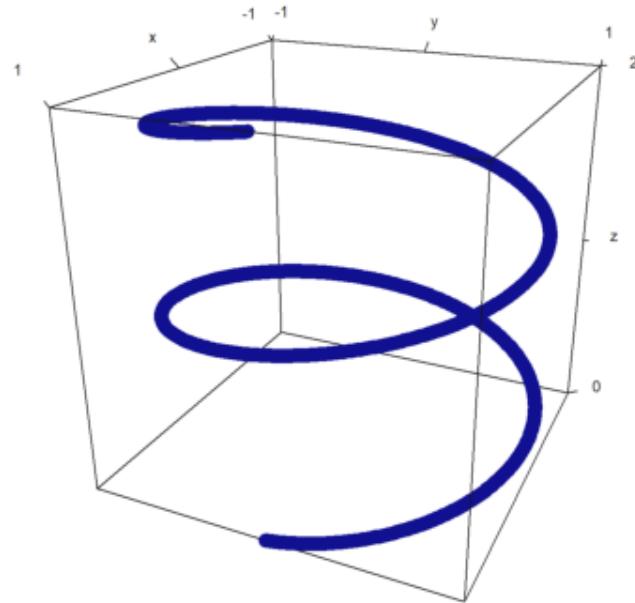


Dimungkinkan juga untuk memplot kurva dalam 3D dengan lebih mudah, yaitu menghitung terlebih dahulu titik-titik kurva. Untuk kurva pada bidang digunakan barisan koordinat dan parameter `wire=true`.

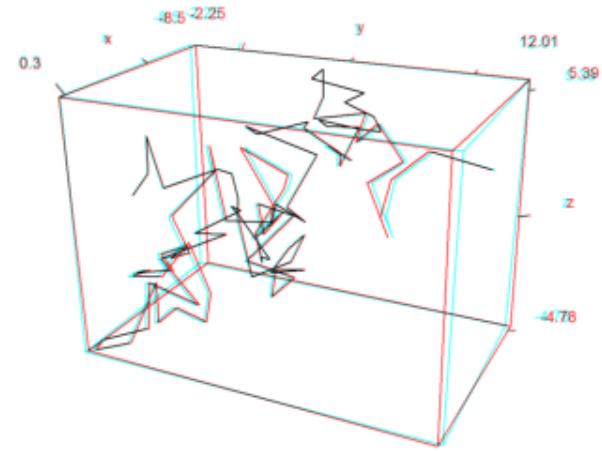
```
>t=linspace(0,8pi,500); ...
>plot3d(sin(t),cos(t),t/10,>wire,zoom=3):
```



```
>t=linspace(0,4pi,1000); plot3d(cos(t),sin(t),t/2pi,>wire, ...
>lineWidth=10, wireColor=blue):
```



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



---

#### 8. Menggambar Fungsi Implisit Tiga Dimensi (3D)

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

### Plot Implisit

Ada juga plot implisit dalam tiga dimensi. Euler menghasilkan pemotongan melalui objek. Fitur plot3d mencakup plot implisit. Plot ini menunjukkan himpunan nol suatu fungsi dalam tiga variabel.

Solusi dari

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

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

- implisit=1: dipotong sejajar bidang y-z
- implisit=2: dipotong sejajar dengan bidang x-z
- implisit=4: dipotong sejajar bidang x-y

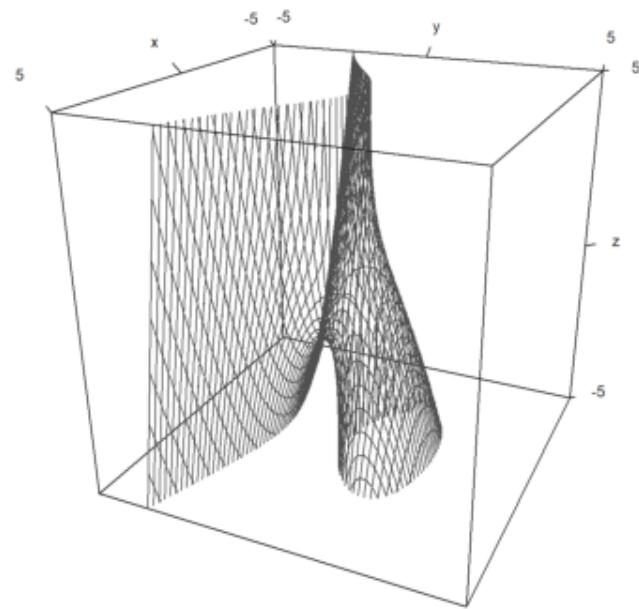
Tambahkan nilai-nilai ini, jika Anda mau. Dalam contoh kita memplot

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

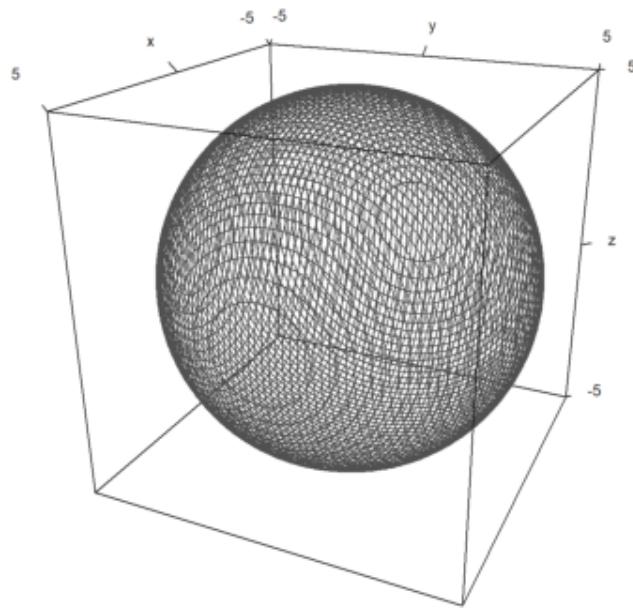
### Contoh Fungsi Implisit

---

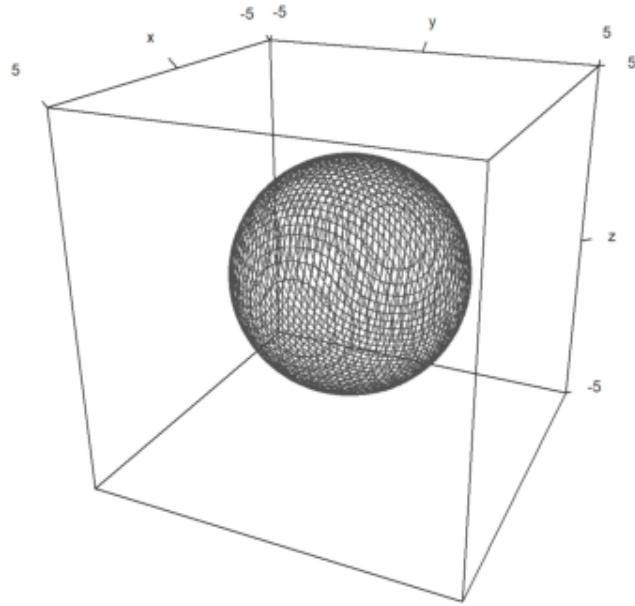
```
>plot3d("x^2+y^3+z*y-1",r=5,implicit=3):
```



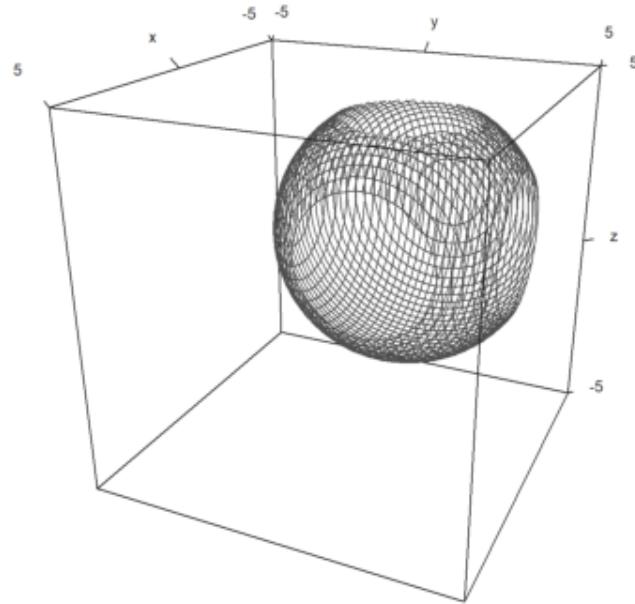
```
>plot3d("x^2 + y^2 + z^2 - 25",r=5,implicit=3):
```



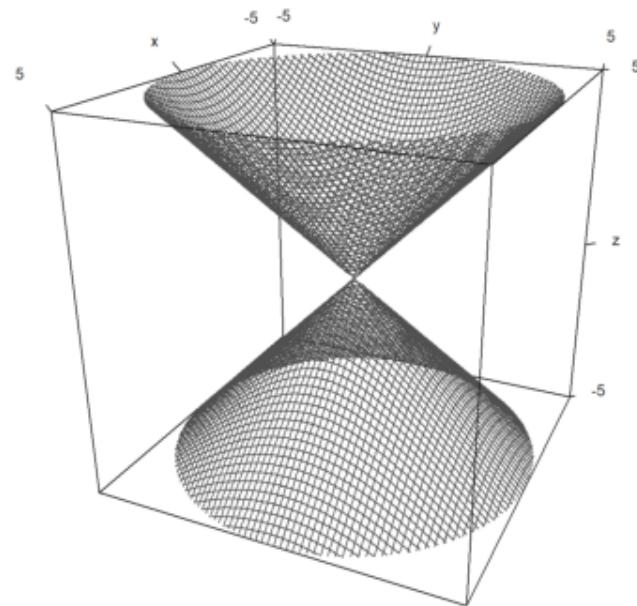
```
>plot3d("x^2 + y^2 + z^2 - 10",r=5,implicit=3):
```



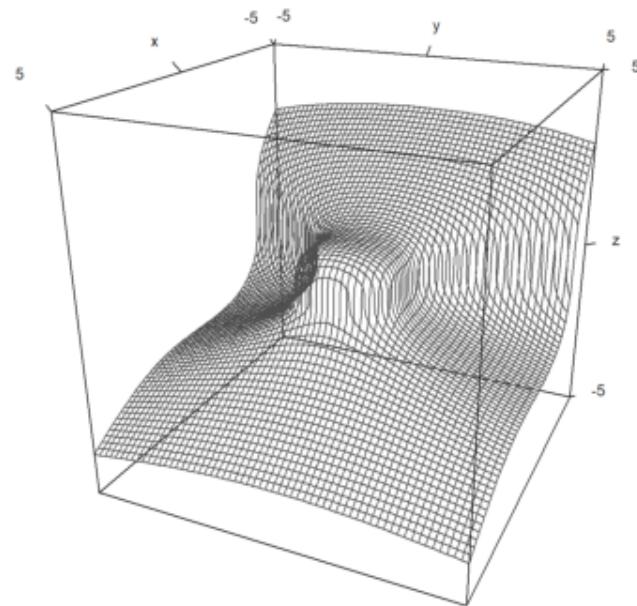
```
>plot3d("x^2 + y^2 + z^2 - 5x - 5y - 5z + 10",r=5,implicit=3):
```



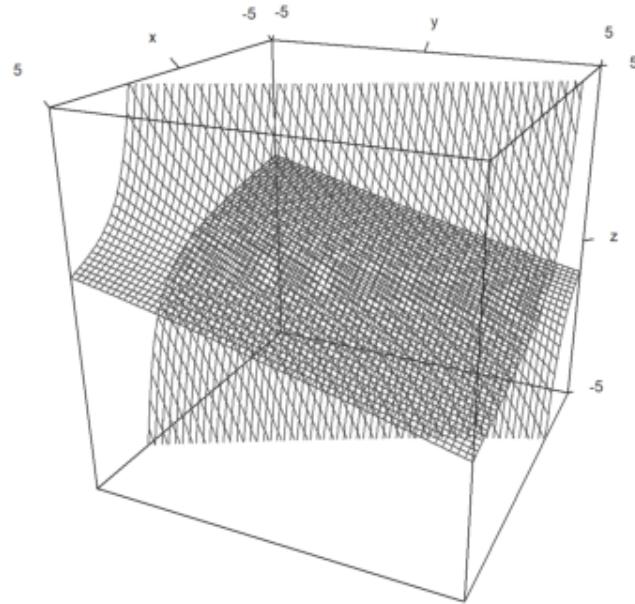
```
>plot3d("x^2 + y^2 - z^2",r=5,implicit=3):
```



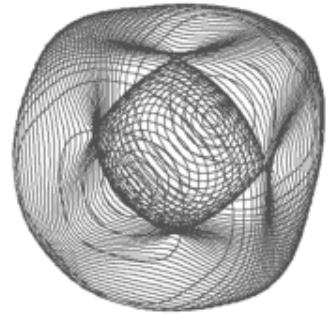
```
>plot3d("x^3 + 2*y^2 +3*z^3-4",r=5,implicit=3):
```



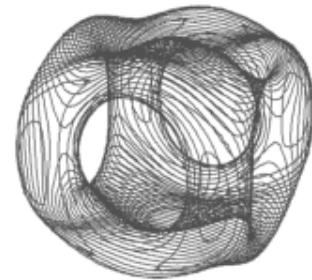
```
>plot3d("x^2+y^2+z^2+2*x*y+4*y*z+8*z*x-20",r=5,implicit=3):
```



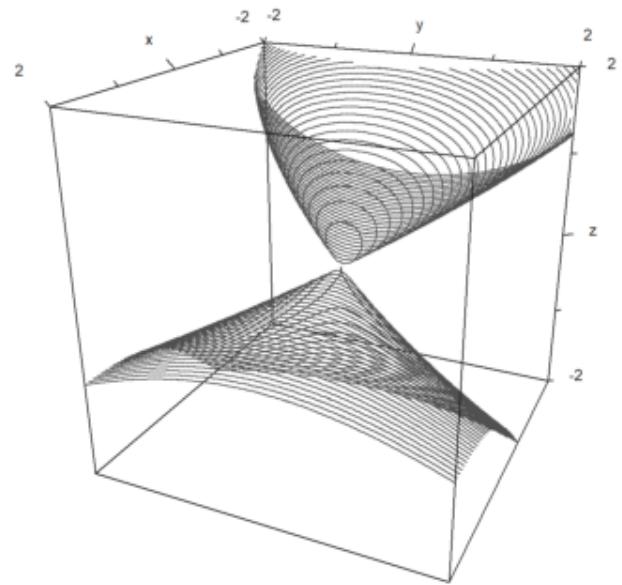
```
>c=1; d=1;  
>plot3d("((x^2+y^2-c^2)^2+(z^2-1)^2)*((y^2+z^2-c^2)^2+(x^2-1)^2)*((z^2+x^2-c^2)^2+(y^2-1)^2)-d",r=2,
```



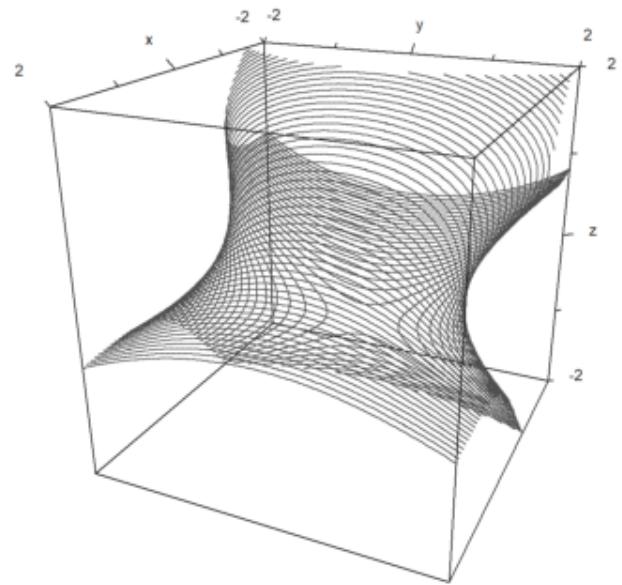
```
>c=1; d=1;  
>plot3d("((x^2+y^2+c^2)^2+(z^2-1)^2)*((y^2+z^2-c^2)^2+(x^2-1)^2)*((z^2+x^2-c^2)^2+(y^2-1)^2)-d",r=2,
```



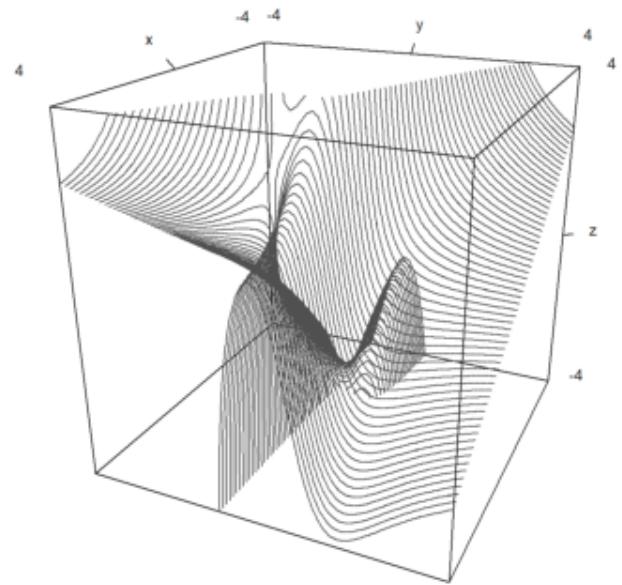
```
>plot3d("x^2+y^2+4*x*z+z^3",>implicit,r=2,zoom=2.5):
```



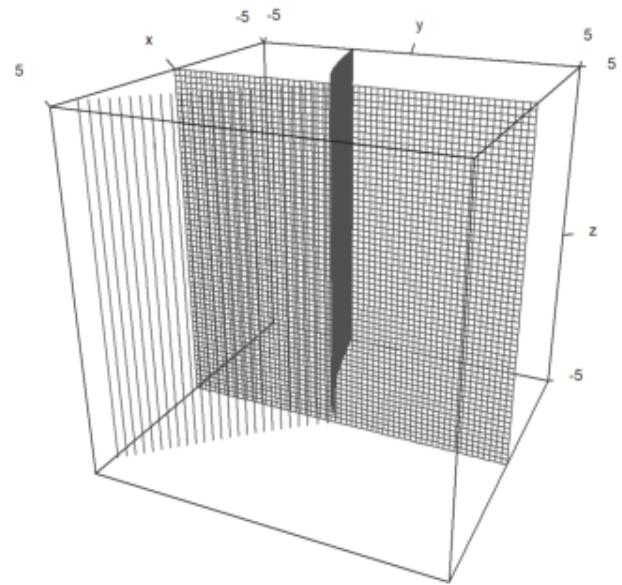
```
>plot3d("x^2+y^2+4*x*z+z^3-2",>implicit,r=2,zoom=2.5):
```



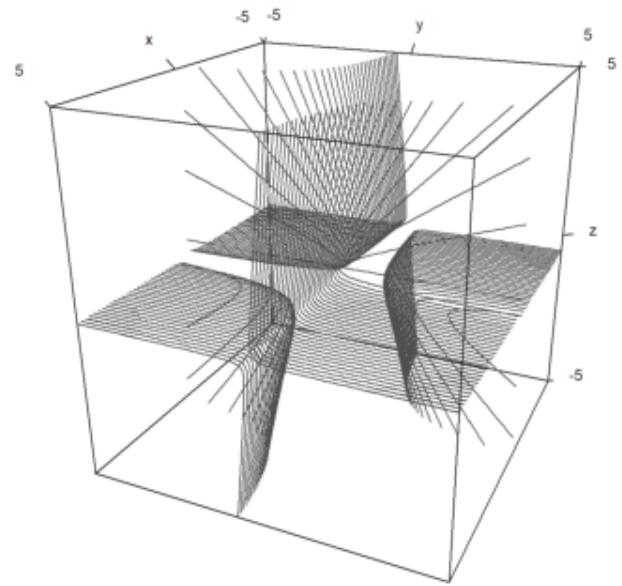
```
>plot3d("5*x*y^2+x*z+2*y^3*z-5",>implicit,r=4,zoom=2.5):
```



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



```
>plot3d("x*y-z^2+2*x*y*z-0",>implicit,r=5,zoom=2.5):
```



Fungsi Implisit menggunakan Povray

---

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 keluaran ekspresi Maxima atau Euler.

$$((x^2 + y^2 - c^2)^2 + (z^2 - 1)^2) * ((y^2 + z^2 - c^2)^2 + (x^2 - 1)^2) * ((z^2 + x^2 - c^2)^2 + (y^2 - 1)^2) = d$$

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

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

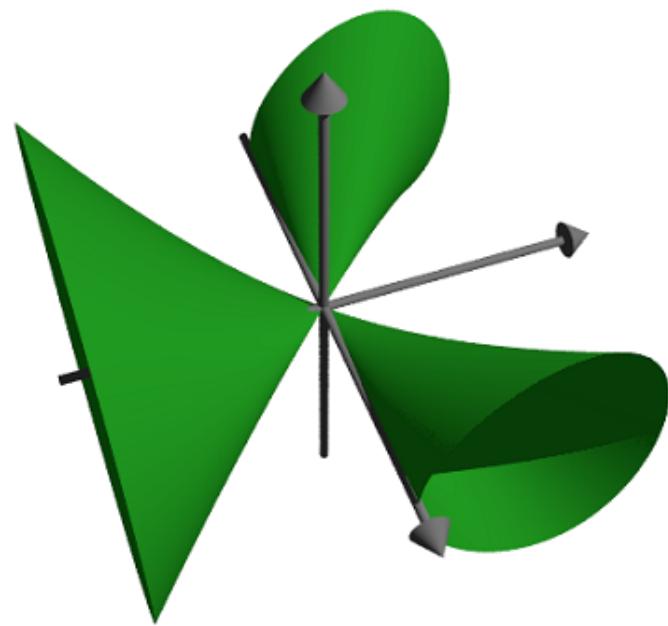
```
>povstart(angle=25°,height=10°);
>writeln(povsurface("pow(x,2)+pow(y,2)*pow(z,2)-1",povlook(blue),povbox(-2,2,"")));
>povend();
```



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

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

```
>povstart(angle=70°,height=50°,zoom=4);  
>writeln(povsurface("pow(x,2)*y-pow(y,3)-pow(z,2)",povlook(green))); ...  
>writeAxes(); ...  
>povend();
```



## 9. Menggambar Titik pada Ruang Tiga Dimensi

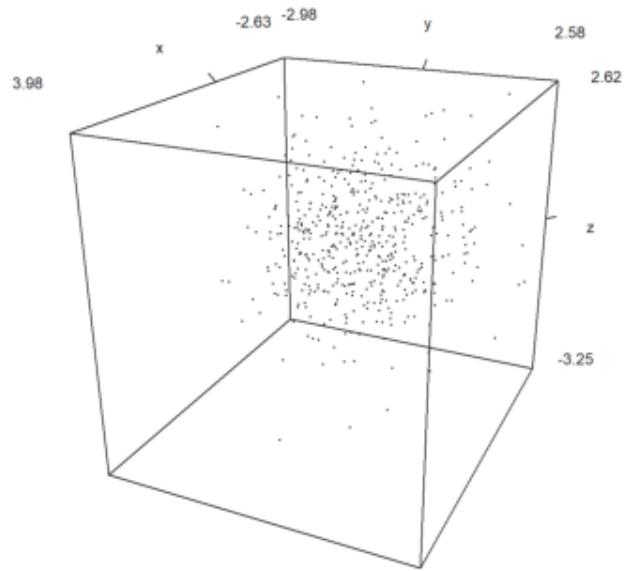
---

Pada sub materi kali ini, sesuai dengan judulnya, kita akan mempelajari bagaimana menggambar titik pada ruang tiga dimensi.

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="."):
```



## Contoh 1

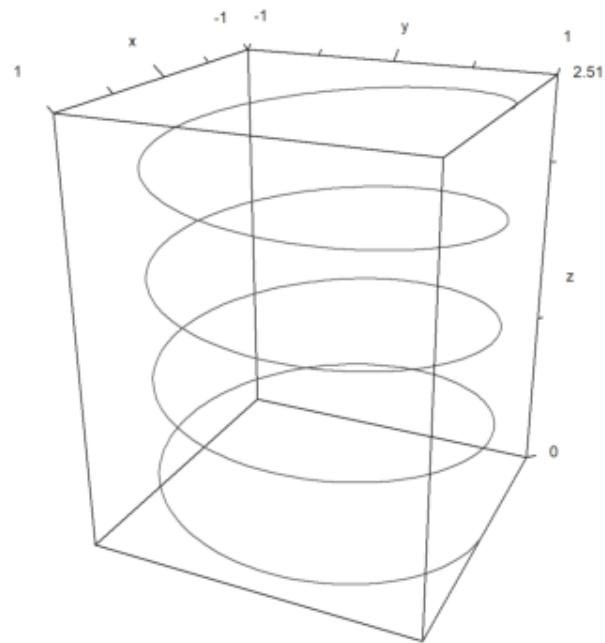
---

Dimungkinkan juga untuk memplot kurva dalam 3D. Dalam hal ini, lebih mudah untuk menghitung terlebih dahulu titik-titik kurva. Untuk kurva pada bidang kita menggunakan barisan koordinat dan parameter wire=true.

```
>t=linspace(0,8pi,500)
```

```
[0,  0.0502655,  0.100531,  0.150796,  0.201062,  0.251327,  0.301593,
0.351858,  0.402124,  0.452389,  0.502655,  0.55292,  0.603186,
0.653451,  0.703717,  0.753982,  0.804248,  0.854513,  0.904779,
0.955044,  1.00531,  1.05558,  1.10584,  1.15611,  1.20637,  1.25664,
1.3069,  1.35717,  1.40743,  1.4577,  1.50796,  1.55823,  1.6085,
1.65876,  1.70903,  1.75929,  1.80956,  1.85982,  1.91009,  1.96035,
2.01062,  2.06088,  2.11115,  2.16142,  2.21168,  2.26195,  2.31221,
2.36248,  2.41274,  2.46301,  2.51327,  2.56354,  2.61381,  2.66407,
2.71434,  2.7646,  2.81487,  2.86513,  2.9154,  2.96566,  3.01593,
3.06619,  3.11646,  3.16673,  3.21699,  3.26726,  3.31752,  3.36779,
3.41805,  3.46832,  3.51858,  3.56885,  3.61911,  3.66938,  3.71965,
3.76991,  3.82018,  3.87044,  3.92071,  3.97097,  4.02124,  4.0715,
4.12177,  4.17204,  4.2223,  4.27257,  4.32283,  4.3731,  4.42336,
4.47363,  4.52389,  4.57416,  4.62442,  4.67469,  4.72496,  4.77522,
4.82549,  4.87575,  4.92602,  4.97628,  5.02655,  5.07681,  5.12708,
5.17734,  5.22761,  5.27788,  5.32814,  5.37841,  5.42867,  5.47894,
5.5292,  5.57947,  5.62973,  5.68,  5.73027,  5.78053,  5.8308,
5.88106,  5.93133,  5.98159,  6.03186,  6.08212,  6.13239,  6.18265,
6.23292,  6.28319,  6.33345,  6.38372,  6.43398,  6.48425,  6.53451,
6.58478,  6.63504,  6.68531,  6.73557,  6.78584,  6.83611,  6.88637,
... ]
```

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



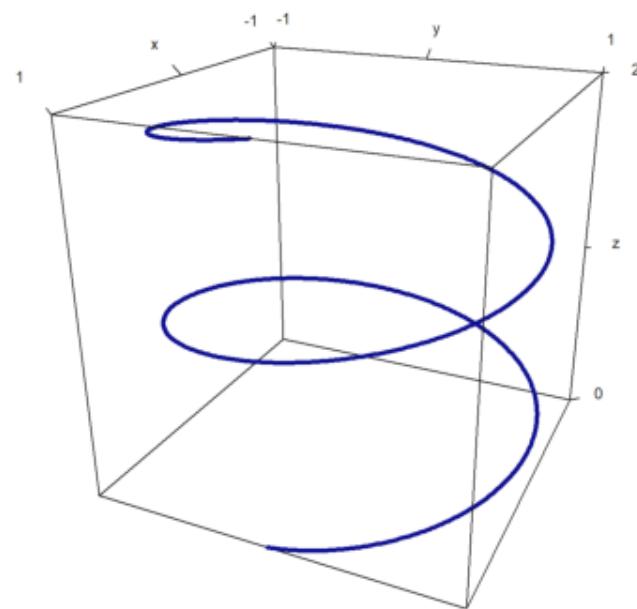
## Contoh 2

---

```
>t=linspace(0,4pi,1000)
```

```
[0,  0.0125664,  0.0251327,  0.0376991,  0.0502655,  0.0628319,
 0.0753982,  0.0879646,  0.100531,  0.113097,  0.125664,  0.13823,
 0.150796,  0.163363,  0.175929,  0.188496,  0.201062,  0.213628,
 0.226195,  0.238761,  0.251327,  0.263894,  0.27646,  0.289027,
 0.301593,  0.314159,  0.326726,  0.339292,  0.351858,  0.364425,
 0.376991,  0.389557,  0.402124,  0.41469,  0.427257,  0.439823,
 0.452389,  0.464956,  0.477522,  0.490088,  0.502655,  0.515221,
 0.527788,  0.540354,  0.55292,  0.565487,  0.578053,  0.590619,
 0.603186,  0.615752,  0.628319,  0.640885,  0.653451,  0.666018,
 0.678584,  0.69115,  0.703717,  0.716283,  0.728849,  0.741416,
 0.753982,  0.766549,  0.779115,  0.791681,  0.804248,  0.816814,
 0.82938,  0.841947,  0.854513,  0.86708,  0.879646,  0.892212,
 0.904779,  0.917345,  0.929911,  0.942478,  0.955044,  0.967611,
 0.980177,  0.992743,  1.00531,  1.01788,  1.03044,  1.04301,  1.05558,
 1.06814,  1.08071,  1.09327,  1.10584,  1.11841,  1.13097,  1.14354,
 1.15611,  1.16867,  1.18124,  1.19381,  1.20637,  1.21894,  1.2315,
 1.24407,  1.25664,  1.2692,  1.28177,  1.29434,  1.3069,  1.31947,
 1.33204,  1.3446,  1.35717,  1.36973,  1.3823,  1.39487,  1.40743,
 1.42,  1.43257,  1.44513,  1.4577,  1.47027,  1.48283,  1.4954,
 1.50796,  1.52053,  1.5331,  1.54566,  1.55823,  1.5708,  1.58336,
 ... ]
```

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



---

Selanjutnya, kita akan melihat contoh visualisasi titik pada 3D dengan povray.

Fungsi povray pada dasarnya memiliki maksud yang sama dengan plot3d, yakni menghasilkan grafik f (x,y) atau permukaan dengan koordinat X, Y, Z dalam matriks. Bedanya, fungsi ini memulai raytracker secara otomatis dan memuat adegan di dalam notebook euler.

Yang pertama, kita perlu memuat file povray. Perlu diingat bahwa povray memiliki sistem koordinat yang akan menerjemahkannya ke sistem povray.

```
>load povray;
```

Selanjutnya, kita akan memanggil povray kita sehingga berisi path ke povray dapat dieksekusi

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

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

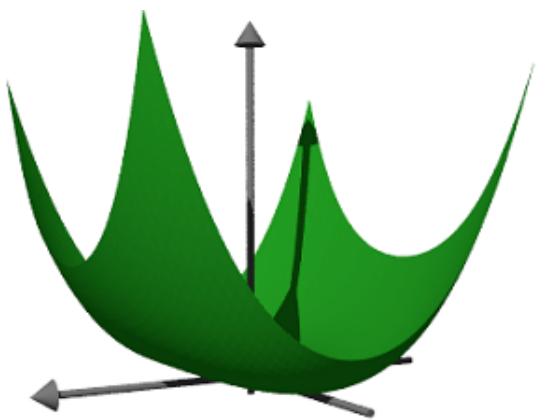
## Contoh 3

---

Selanjutnya, ketika kita ingin menuliskan fungsi secara manual.

Kita bisa memplot himpunan titik yang ada pada bidang kompleks. Yang mana, hasil kali dari jarak ke 1 dan -1 sama dengan 1.

```
>pov3d("((x-1)^2+y^2)*((x+1)^2+y^2)/40",r=2, ...
> angle=-120°,level=1/40,dlevel=0.005,light=[-1,1,1],height=10°,n=50, ...
> <fscale,zoom=3)
```



## Contoh 4

---

Kita dapat membuat visualisasi dengan memplot koordinat pada povray. Oleh karenanya, kita memerlukan tiga matriks untuk mendefinisikan objek kita.

Contoh di bawah ini akan membutar 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]);
```



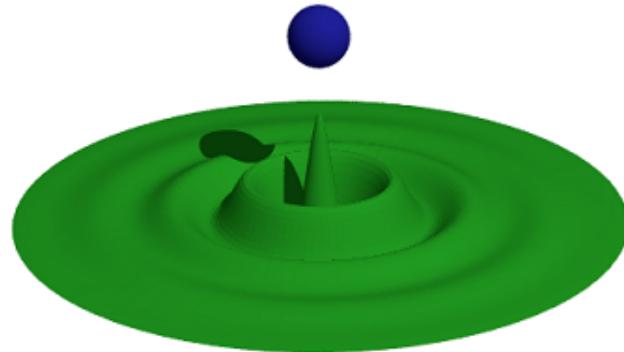
## Contoh 5

---

Selanjutnya, kita akan memplot gelombang teredam. Kita akan menghasilkan gelombang dengan bahasa matriks Euler.

Selanjutnya juga ditunjukkan 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(blue)), ...
> w=500,h=300);
```



## Contoh 6

---

Selanjutnya, kita mencoba menambahkan vektor normal di setiap titik matriks nantinya, dikarenakan sangat sedikit titik yang menghasilkan permukaan sangat halus.

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

$$\begin{matrix} 2 & 3 \\ x & y \end{matrix}$$

Persamaan permukaannya yakni  $[x,y,z]$ .

Selanjutnya, kita hitung kedua turunan dari  $x$  dan  $y$  dan mengambil produk silang sebagai normal.

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

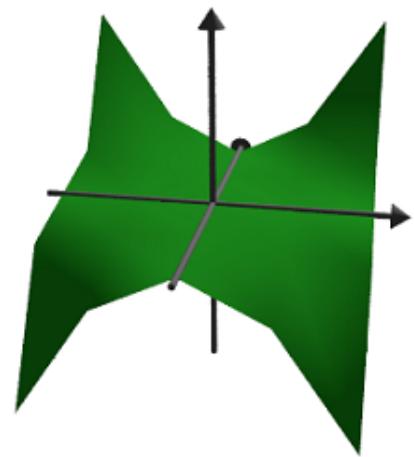
Lanjut, kita definiskan normal sebagai produk silang dari turunan dan mendefinisikan fungsi koordinat

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

$$[-\frac{3}{2}x^{\frac{3}{2}}y^{\frac{1}{2}}, -\frac{3}{2}x^{\frac{1}{2}}y^{\frac{3}{2}}, 1]$$

Kita gunakan 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>;
```



## Contoh 7

---

Contoh selanjutnya, kita akan mencoba simpul Trefoil yang diterapkan oleh A. Busser Povray.

Pada contoh ini, digunakan vektor normal dan maksima untuk menghitung nilai normal.

```
>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);
```

Selanjutnya, kita turunkan kedua vektor x dan y ke dalam dx dan dy

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

Diperoleh produk silang dari dua turunan yang ada

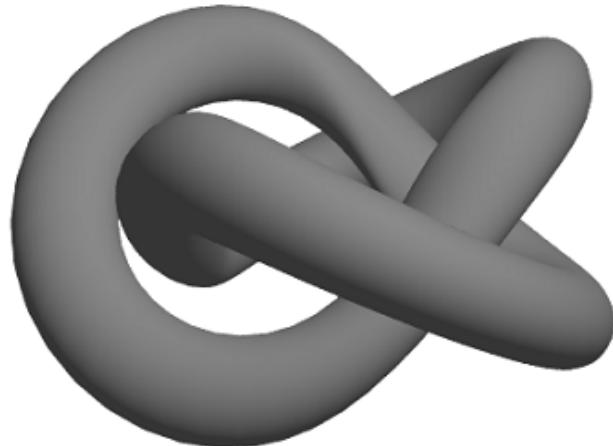
```
>dn &= crossproduct(dx,dy);
```

Selanjutnya, kita mengevaluasi semua rumusan di atas secara numerik

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

Karena vektor normal merupakan evaluasi dari ekspresi simbolik pada  $dn[i]$ dengan  $i = 1,2,3$ . Maka sintaks ini merupakan parameter yang didefinisikan terlebih dahulu ekspresi simbolik NX, NY, dan NZ nya.

```
>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));
```

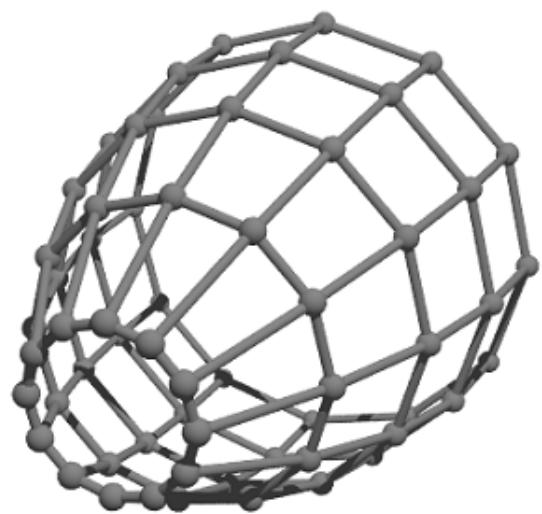


## Contoh 8

---

Selanjutnya, kita dapat membuat visualisasi kita dalam bentuk grid di 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();
```

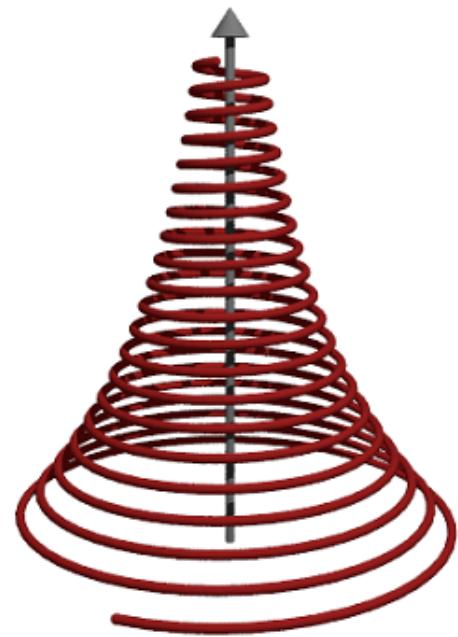


## Contoh 9

---

Dapat juga menggunakan povgrid(), sehingga diperoleh

```
>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();
```



10. Mengatur tampilan, warna dan sudut pandang gambar permukaan

---

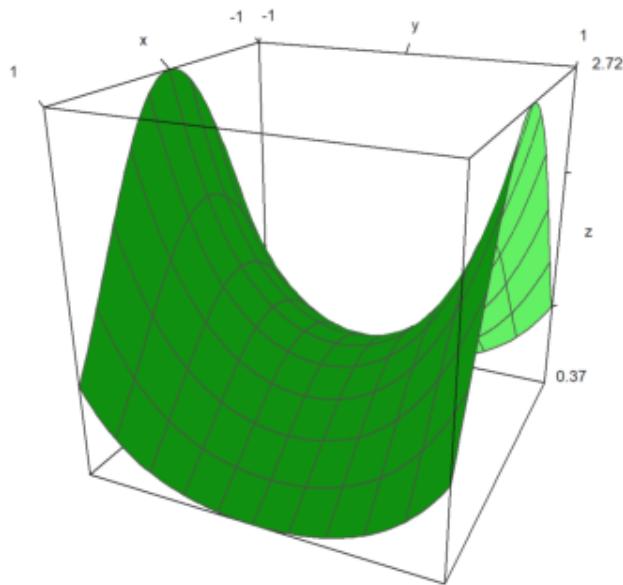
(3D)

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 interaks

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

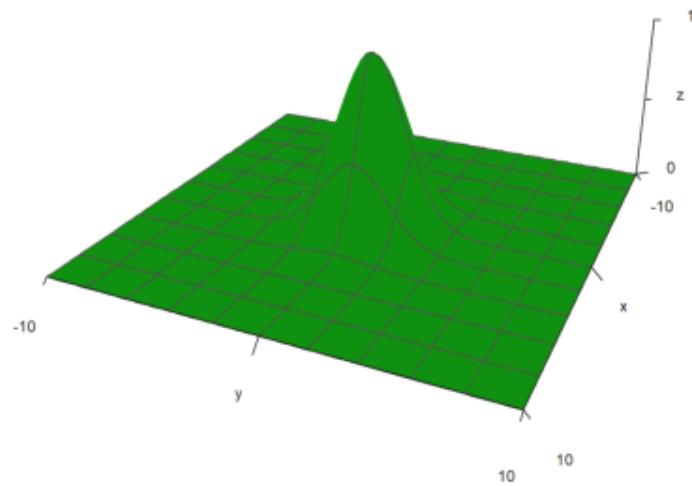
Turn with the vector keys (press return to finish)



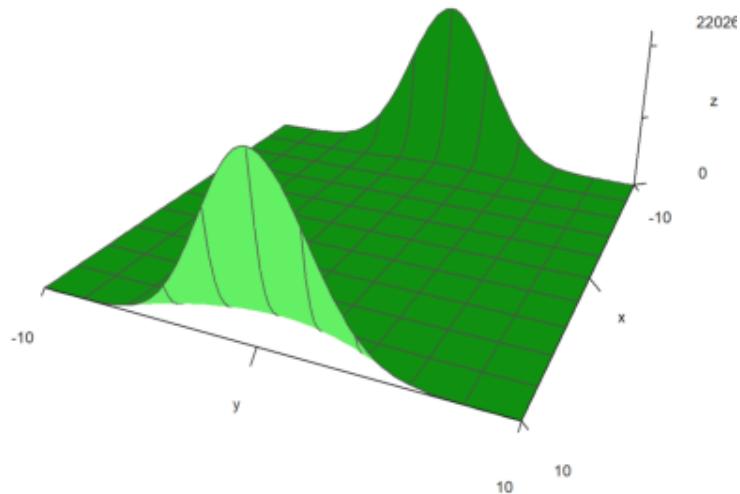
Ada beberapa parameter untuk menskalakan fungsi atau mengubah tampilan grafik.

- `fscale`: skala ke nilai fungsi (defaultnya adalah `<fscale>`). Jika nilai ‘`fscale`’ besar maka dapat mempersempit ”puncak” fungsi, membuatnya lebih tajam.
- `skala`: angka atau vektor  $1 \times 2$  untuk skala ke arah x dan y. Jika semakin besar nilai ‘`scale`’, semakin besar ukuran plotnya, dan sebaliknya.
- `bingkai`: jenis bingkai (default 1).

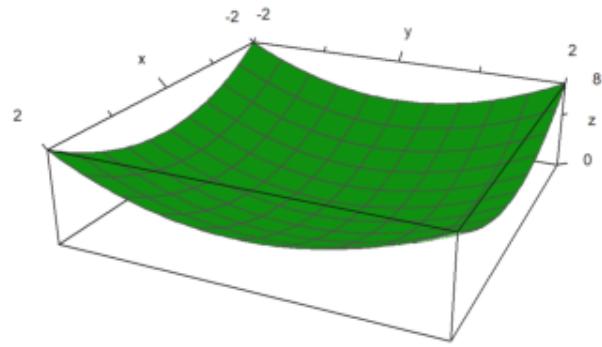
```
>plot3d("exp(-(x^2+y^2)/5)",r=10,n=80,fscale=4,scale=1.2,frame=3):
```



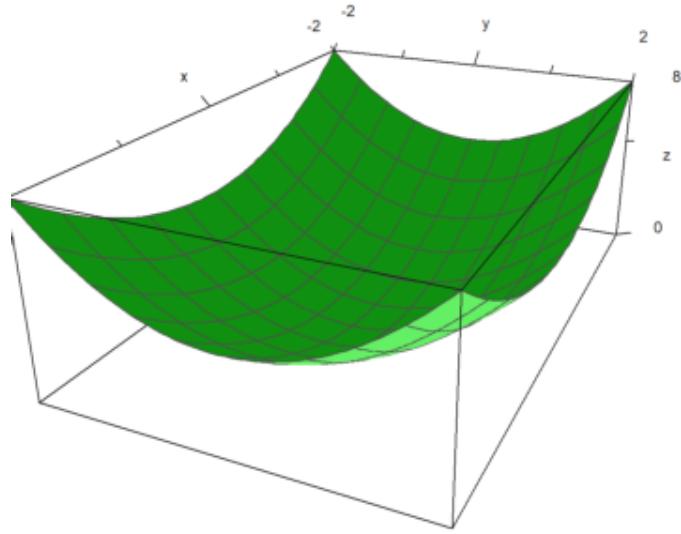
```
>plot3d("exp((x^2-y^2)/10)",r=10,n=80,fscale=4,scale=1.2,frame=3):
```



```
>plot3d("x^2+y^2",r=2,fscale=1/2):
```



```
>plot3d("x^2+y^2",r=2,scale=[1.5,1,0.5]):
```



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.

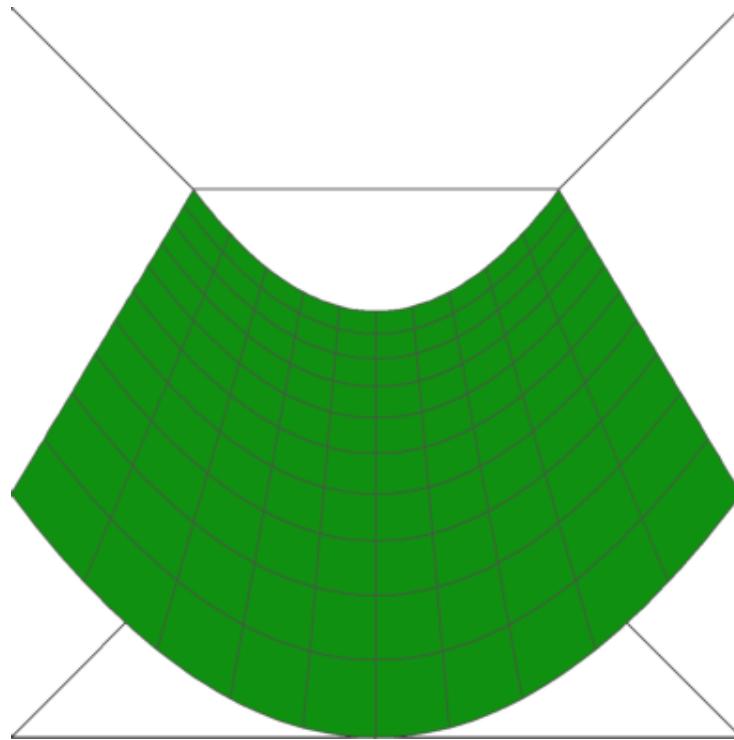
Nilai default dapat diperiksa atau diubah dengan fungsi `view()`. Ini mengembalikan parameter dalam urutan di atas.

```
>view
```

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

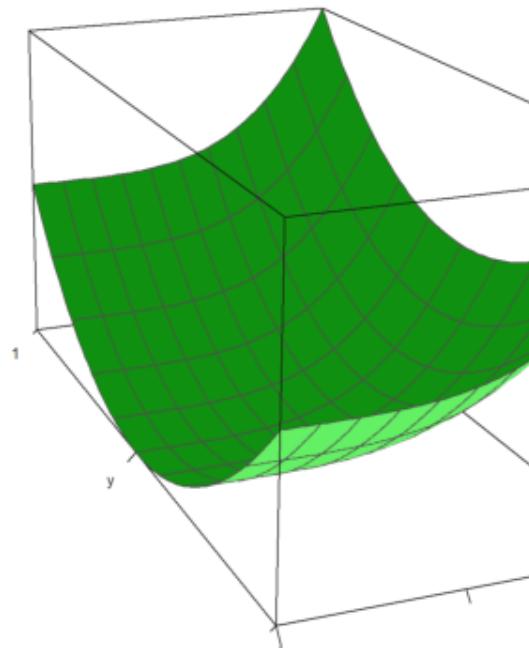
Jarak yang lebih dekat membutuhkan lebih sedikit zoom. Efeknya lebih seperti lensa sudut lebar. Dalam contoh berikut, sudut=0 dan tinggi=0 terlihat dari sumbu y negatif. Label sumbu untuk y disembunyikan dalam kasus ini.

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



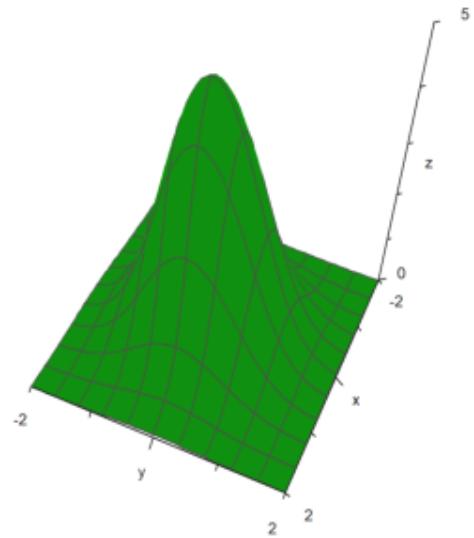
Plot terlihat selalu ke pusat kubus plot. Anda dapat memindahkan pusat dengan parameter tengah.

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



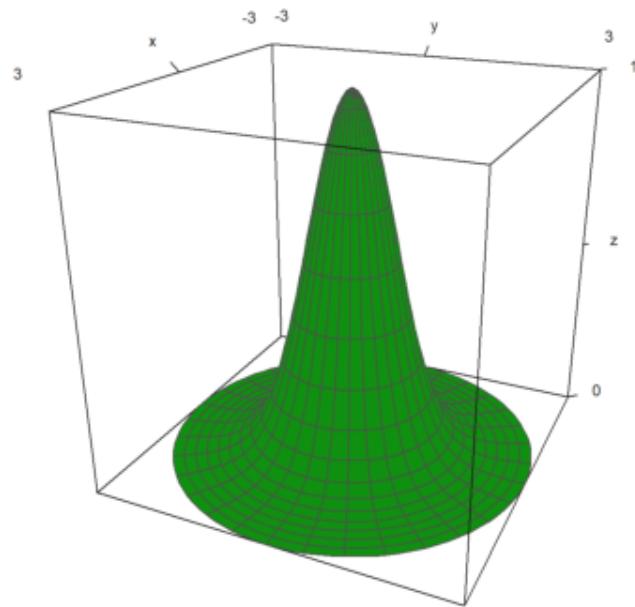
Plot diskalakan agar sesuai dengan kubus satuan untuk dilihat. Jadi tidak perlu mengubah jarak atau zoom tergantung pada ukuran plot. Namun, label mengacu pada ukuran sebenarnya. Jika Anda mematikannya dengan `scale=false`, Anda perlu berhati-hati, bahwa plot masih cocok dengan jendela plot, dengan mengubah jarak pandang atau zoom, dan memindahkan pusat.

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

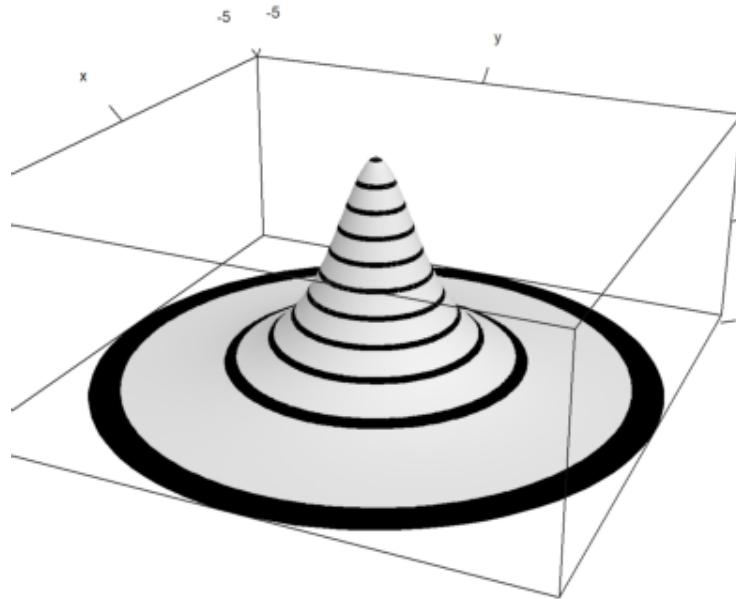


Sebuah plot kutub juga tersedia. Parameter polar=true menggambar plot polar. Fungsi tersebut harus tetap merupakan fungsi dari x dan y. Parameter "fscale" menskalakan fungsi dengan skala sendiri. Jika tidak, fungsi diskalakan agar sesuai dengan kubus.

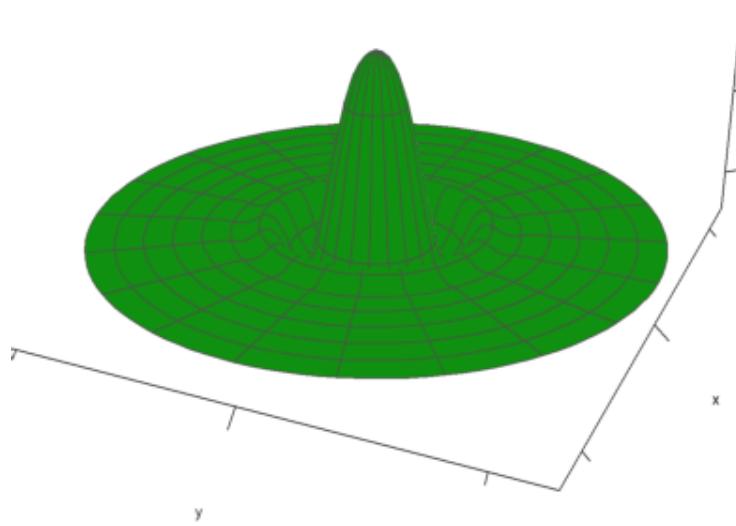
```
>plot3d("exp(-x^2-y^2)",r=3,n=50,grid=20,>polar):
```



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



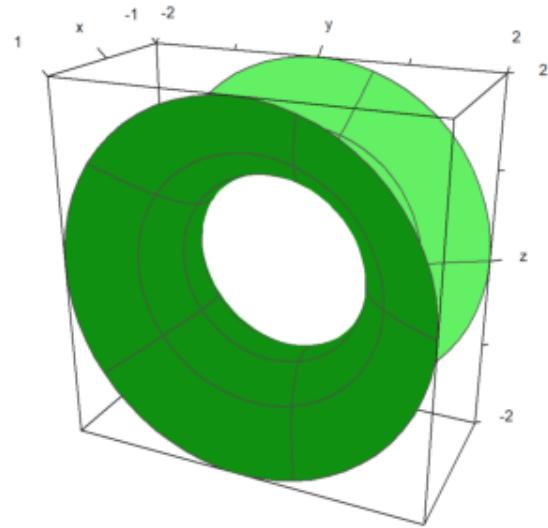
```
>function f(r) := exp(-r/2)*cos(r); ...
>plot3d("f(x^2+y^2)",>polar,scale=[1,1,0.4],r=2pi,frame=3,zoom=4):
```



Rotasi parameter memutar fungsi dalam x di sekitar sumbu x.

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

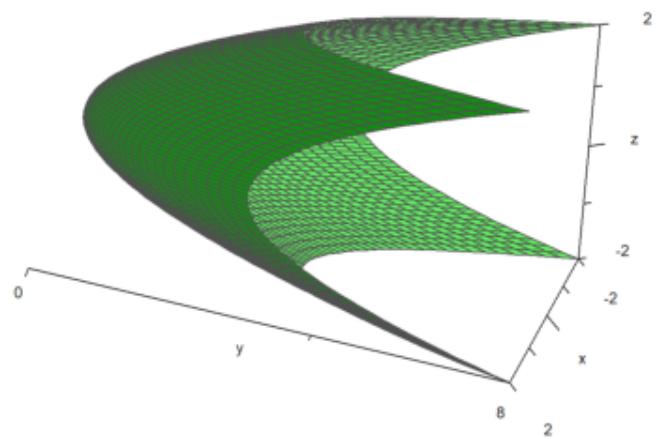
```
>plot3d("x^2+1",a=-1,b=1,rotate=true,grid=5):
```



Berikut adalah plot dengan tiga fungsi. i sekitar sumbu x.

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

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

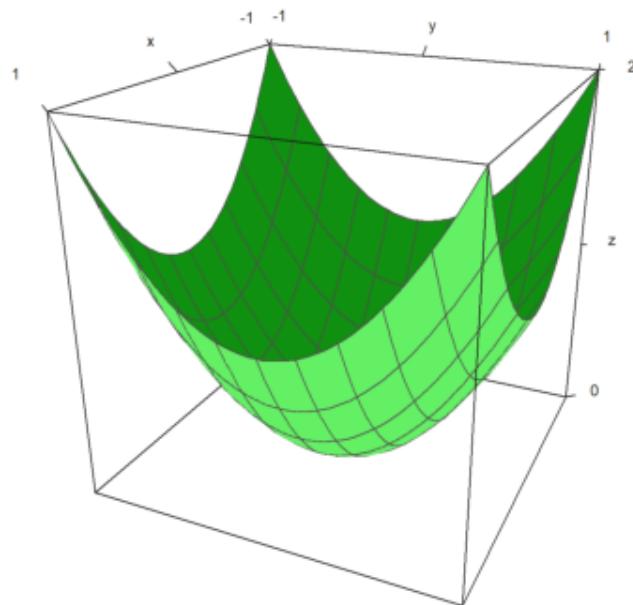


---

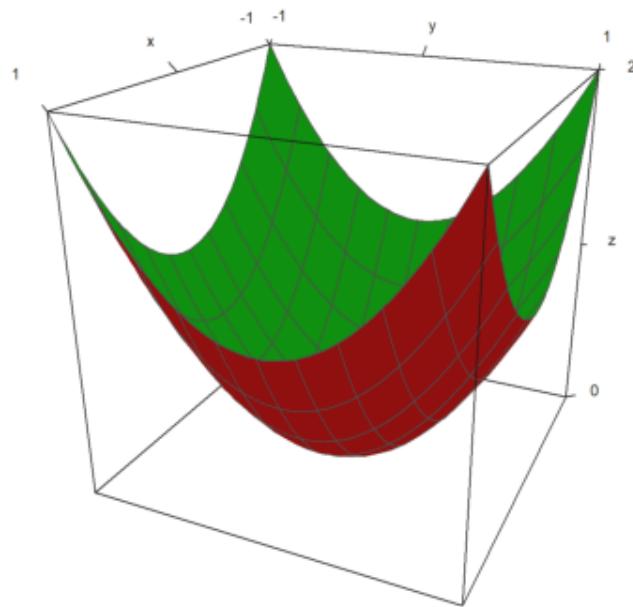
Standarnya adalah kotak kawat yang diisi dengan warna berbeda di kedua sisi. Perhatikan bahwa jumlah default interval grid adalah 10, tetapi plot menggunakan jumlah default 40x40 persegi panjang untuk membangun permukaan. Ini bisa diubah.

- n=40, n=[40,40]: jumlah garis grid di setiap arah
  - grid=10, grid=[10,10]: jumlah garis grid di setiap arah.
- Kami menggunakan default n=40 dan grid=10.

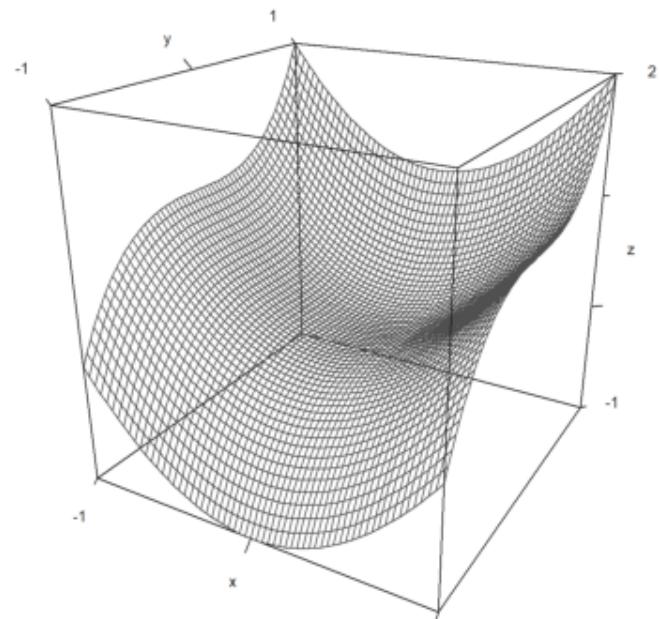
```
>plot3d("x^2+y^2"):
```



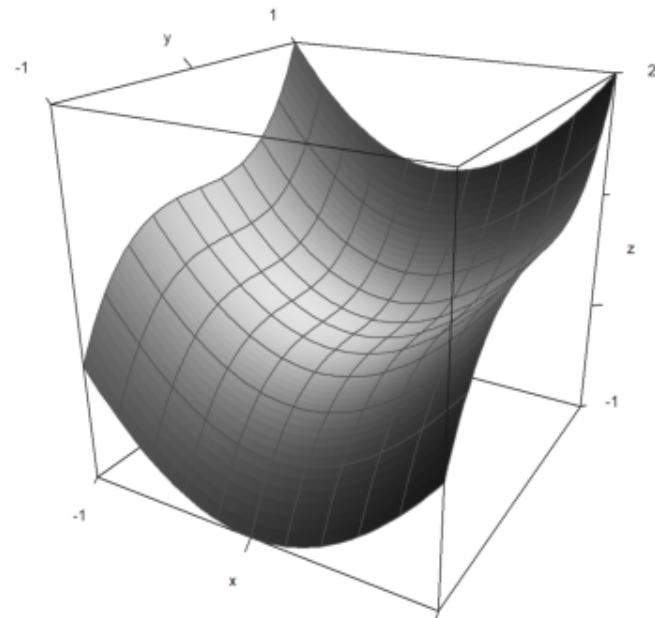
```
>plot3d("x^2+y^2",fillcolor=[red,green]):
```



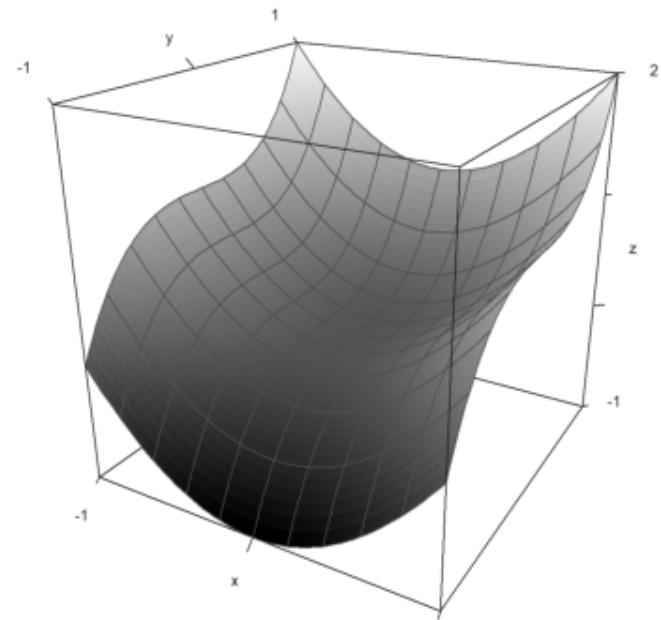
```
>plot3d("x^2+y^3",>wire,angle=30°):
```



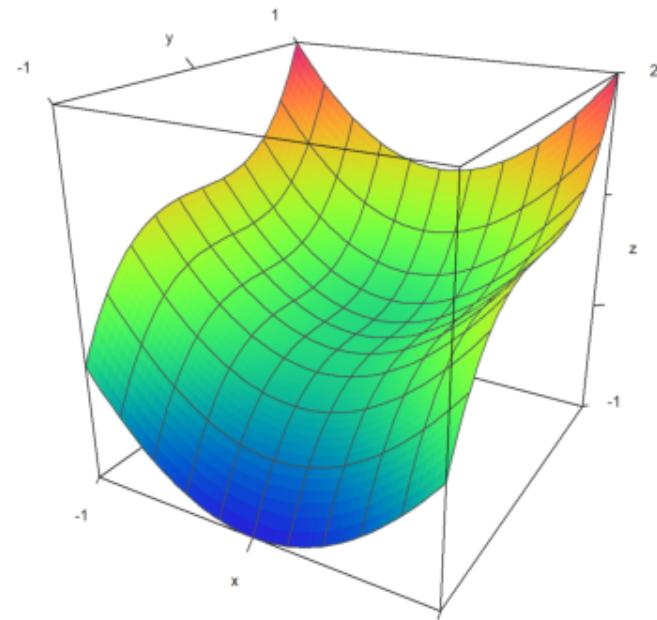
```
>plot3d("x^2+y^3",grid=10,>hue,angle=30°):
```



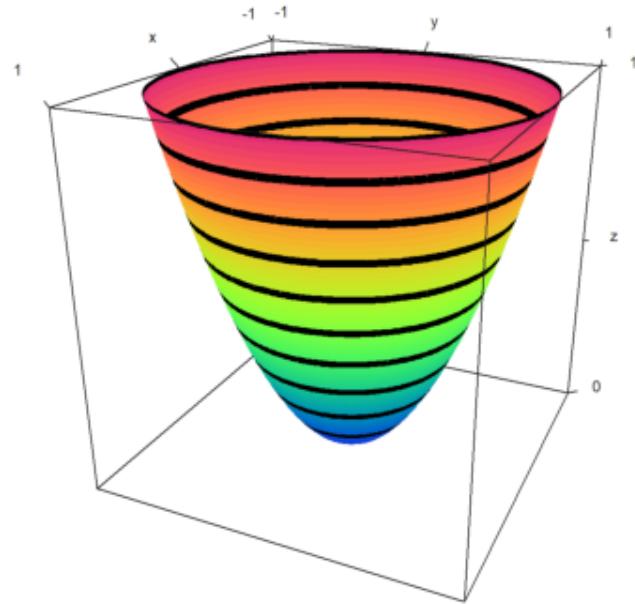
```
>plot3d("x^2+y^3",grid=10,>zhue,angle=30°):
```



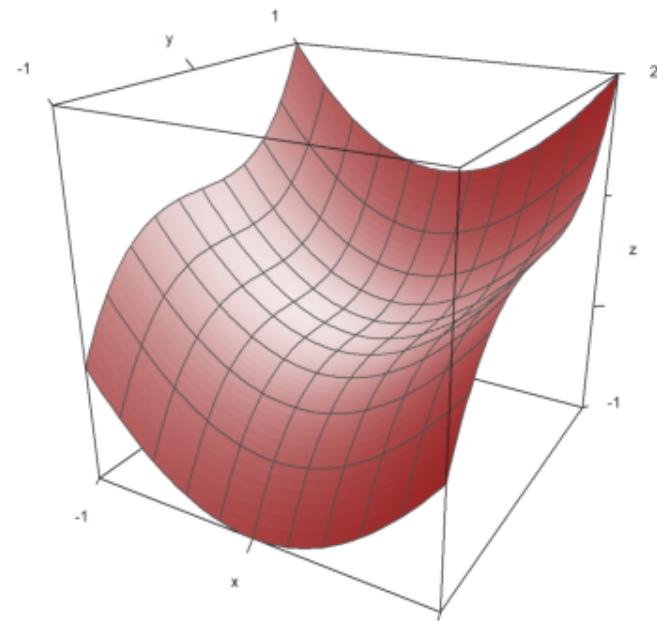
```
>plot3d("x^2+y^3",grid=10,>zhue,>spectral,angle=30°):
```



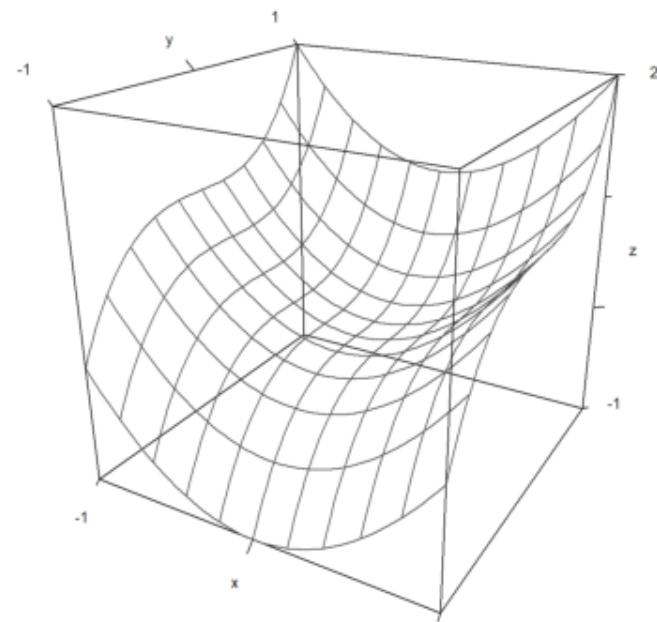
```
>plot3d("x^2+y^2",>spectral,>contour,n=100,>polar):
```



```
>plot3d("x^2+y^3",grid=10,>hue,color=red,angle=30°):
```



```
>plot3d("x^2+y^3",grid=10,>hue,color=transparent,angle=30°):
```



---

## 11. Menampilkan Kontur dan Bidang Kontur Permukaan 3D

Plot kontur adalah teknik grafis untuk merepresentasikan permukaan 3D dengan memplot irisan konstan, yang disebut kontur, dalam format 2D. Plot kontur banyak digunakan dalam kartografi , di mana garis kontur pada peta topografi menunjukkan ketinggian yang sama.

Plot kontur digunakan untuk menampilkan permukaan 3D dengan memplot z-slide pada permukaan 2D. Dalam kontur, kita memiliki 3 variabel x, y, z. Variabel x,y digunakan untuk memberikan nilai z, ( $z=f(x, y)$ ). Variabel x dan y biasanya berada dalam grid.

Untuk membuat plotnya, Euler menambahkan garis grid. Sebaliknya dimungkinkan untuk menggunakan garis datar dan rona satu warna atau rona warna spektral. Euler dapat menggambar ketinggian fungsi pada plot dengan arsiran. 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 levels): Vektor nilai garis kontur.

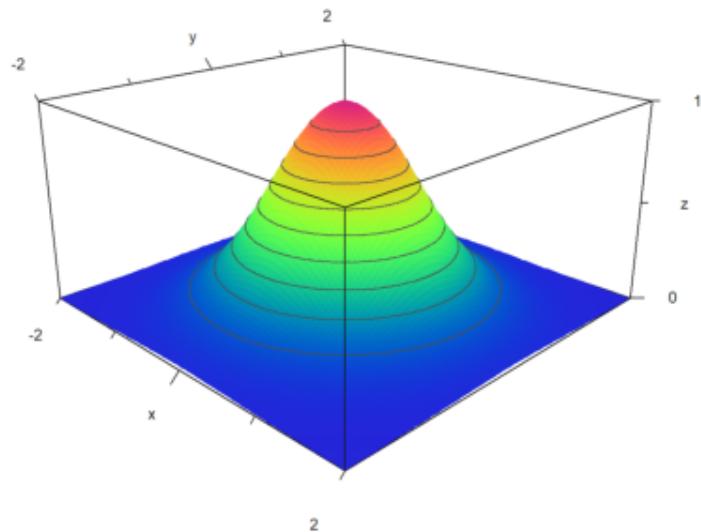
Standarnya adalah level="auto", yang menghitung beberapa garis level secara otomatis. Seperti yang Anda lihat di plot, level sebenarnya adalah rentang level.

Gaya default dapat diubah. Untuk plot kontur berikut, kita menggunakan grid yang lebih halus berukuran 100x100, menskalakan fungsi dan plot, dan menggunakan sudut pandang yang berbeda.

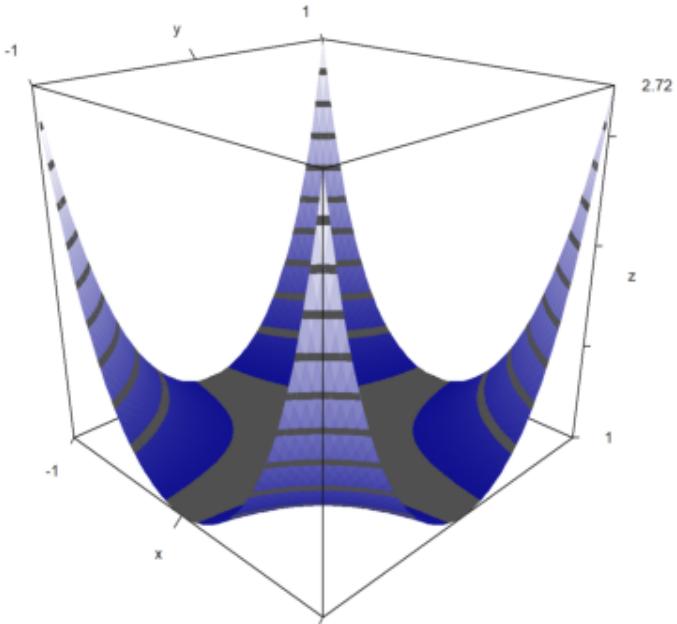
## Contoh

---

```
>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)", angle=45°, >contour, color=blue):
```

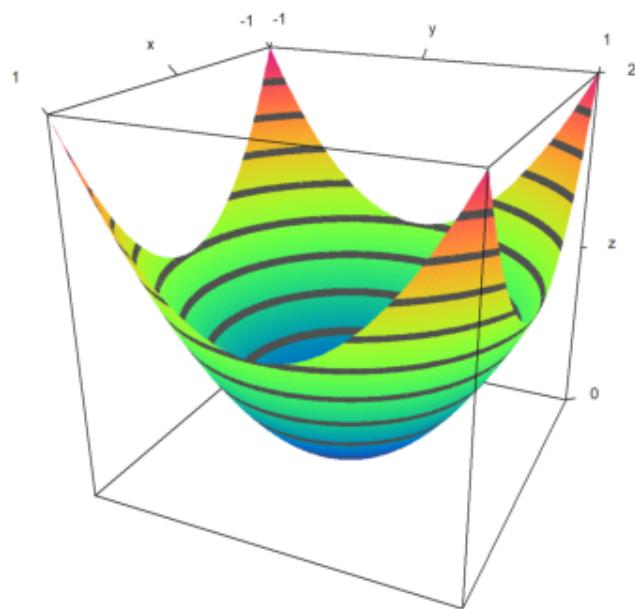


Pada contoh diatas kita menggunakan warna abu-abu untuk bayangan defaultnya. Kita juga bisa memberikan rentang warna untuk spektral.

- >spectral: 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.

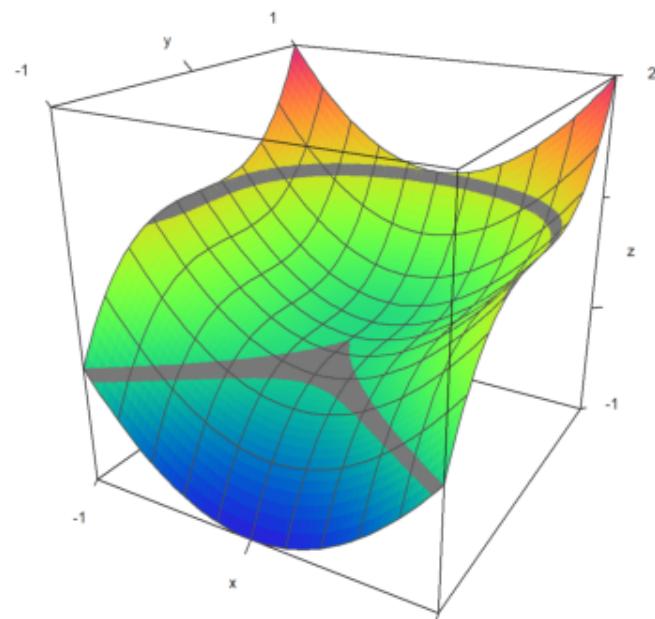
```
>plot3d("x^2+y^2",>spectral,>contour,n=100):
```



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

Selain itu, kita juga melapisi grid 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):
```

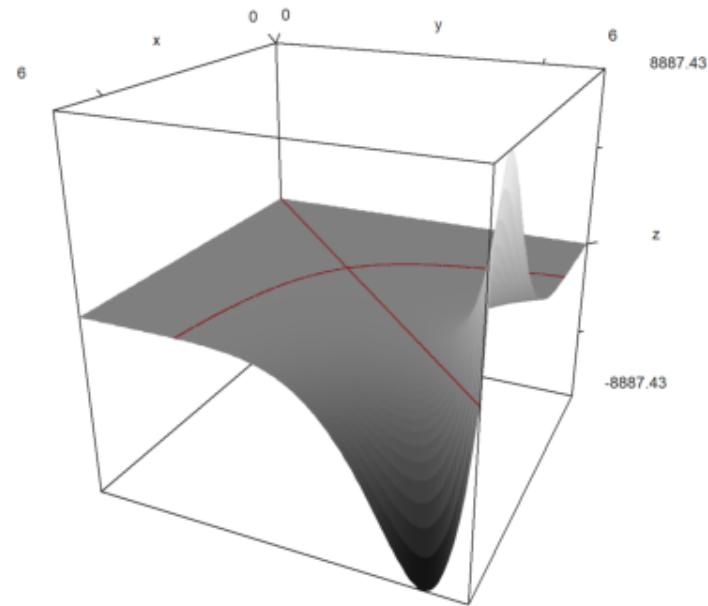


Pada contoh berikut, kita memplot himpunan, di mana

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

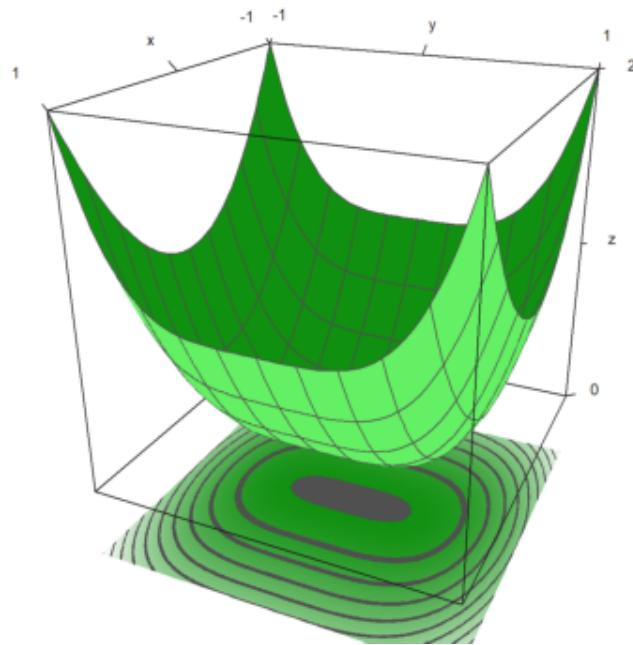
Kami menggunakan satu garis tipis untuk garis level.

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



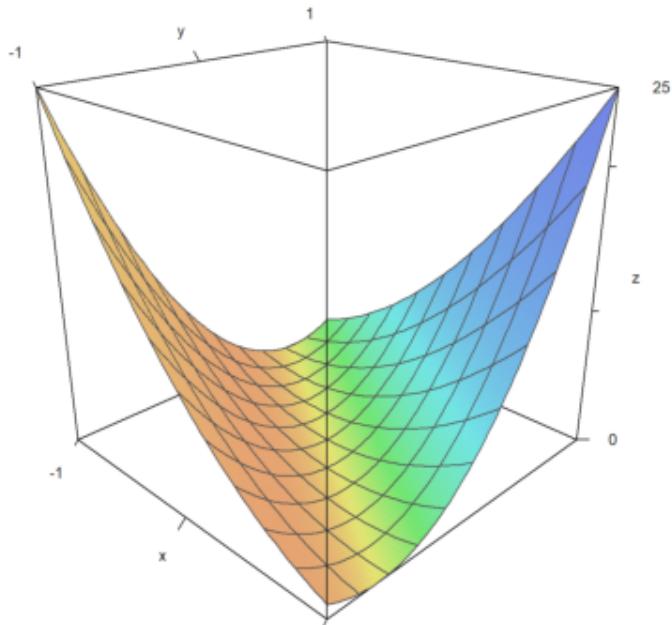
Dimungkinkan untuk menampilkan bidang kontur di bawah plot. Warna dan jarak ke plot juga dapat kita tentukan.

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

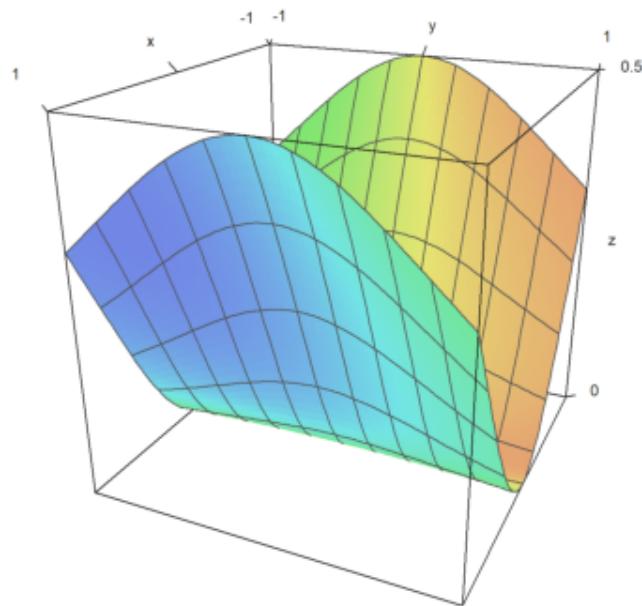


Pada tampilan warna, jika kita menggunakan warna 0 maka akan menghasilkan efek pelangi yang isitimewa.

```
>plot3d("(2*x+3*y)^2",angle=45°,color=0,hue=true,grid=10):
```

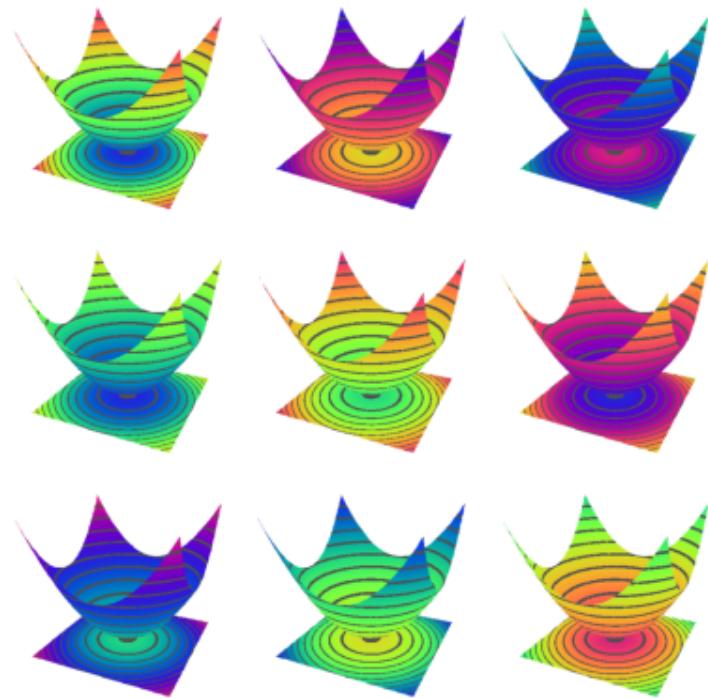


```
>plot3d("x^2/(x^2+y^2+1)",color=0,hue=true,grid=10):
```



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

```
>figure(0):
```



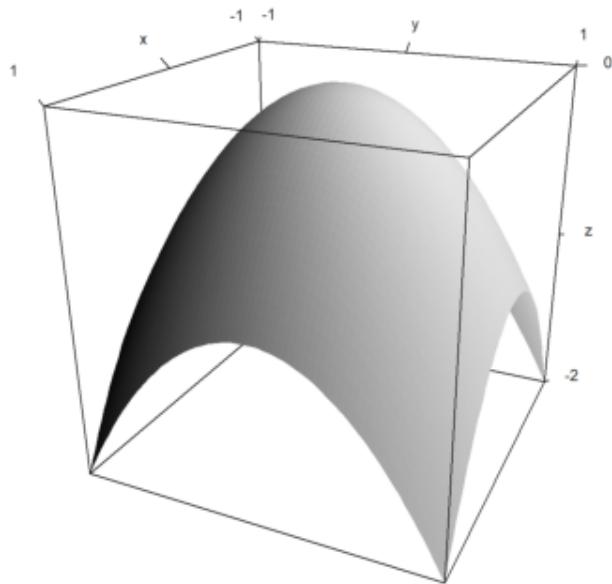
Sumber cahaya dapat diubah dengan 1 dan tombol kursor selama interaksi pengguna. Itu juga dapat diatur dengan parameter.

- light : arah datangnya cahaya
- amb: cahaya sekitar antara 0 dan 1

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



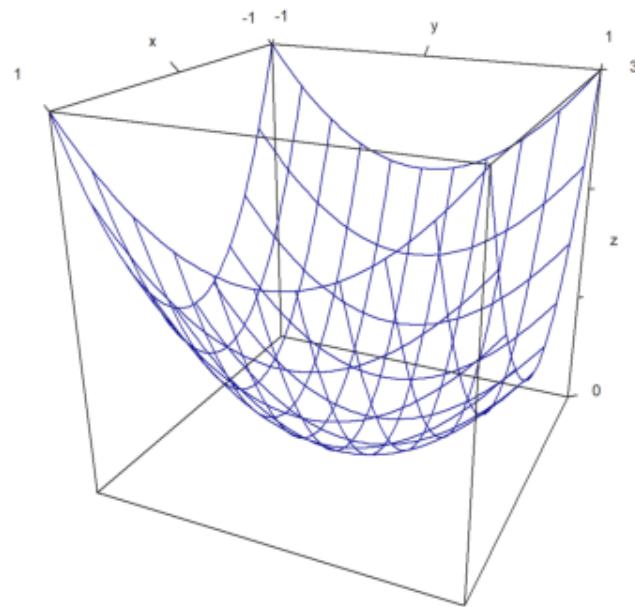
Parameter warna mengubah warna permukaan. Disini kita juga dapat mengubah warna garis levelnya.

```
>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,dl=0.01):
```

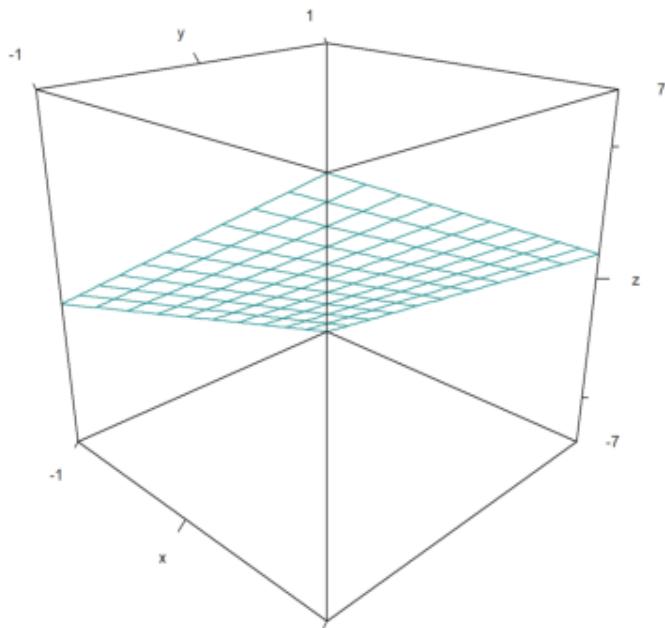


Permukaannya juga bisa dibuat menjadi transparan.

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



```
>plot3d("4*x-3*y",angle=45°,>transparent,grid=10,wirecolor=cyan):
```



---

**12. Menggambar Grafik Tiga Dimensi alam modus anaglif**

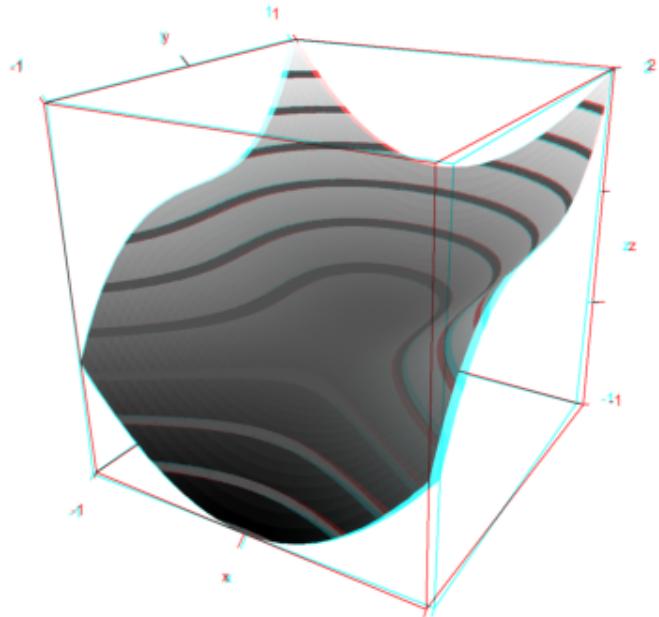
Modus anaglif adalah salah satu teknik yang digunakan untuk menciptakan efek tiga dimensi pada gambar atau video dengan menggunakan dua warna yang berbeda. Teknik ini memanfaatkan persepsi mata manusia yang dapat membedakan warna merah dan biru (hijau/cyan) untuk menciptakan efek kedalaman.

Kacamata anaglif biasanya memiliki satu lensa merah di satu sisi dan satu lensa biru (kadang-kadang hijau/cyan) di sisi lain. Warna lensa merah dan biru ini digunakan untuk memfilter gambar yang mewakili pandangan mata kiri dan mata kanan dalam grafik anaglif.

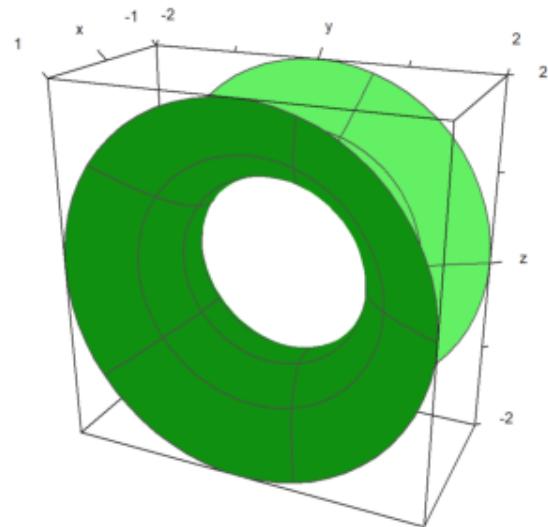
Namun, terdapat beberapa variasi dalam penggunaan warna pada kacamata anaglif, terutama dalam penggunaan lensa merah dan biru. Misalnya, ada juga kacamata anaglif yang menggunakan lensa merah dan cyan (biru muda) sebagai gantinya. Warna lensa yang digunakan dapat bervariasi tergantung pada preferensi pembuat grafik 3D atau produsen kacamata anaglif tertentu.

Jadi, pada umumnya, kacamata anaglif akan memiliki satu lensa yang memungkinkan cahaya merah melewati dan satu lensa yang memungkinkan cahaya biru (atau warna sejenis) melewati. Kedua lensa ini digunakan untuk memisahkan pandangan mata kiri dan mata kanan ketika melihat gambar anaglif, sehingga menciptakan efek tiga dimensi.

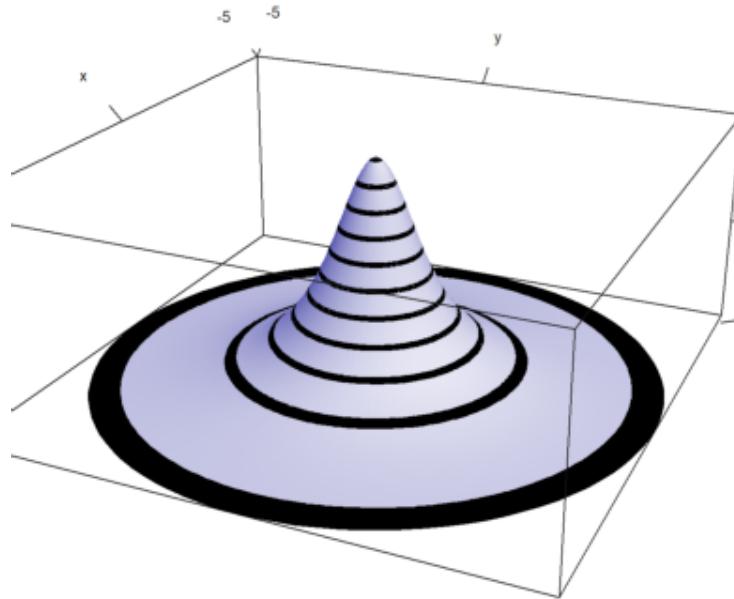
```
>  
>plot3d("x^2+y^3",>anaglyph,>contour,angle=30°):
```



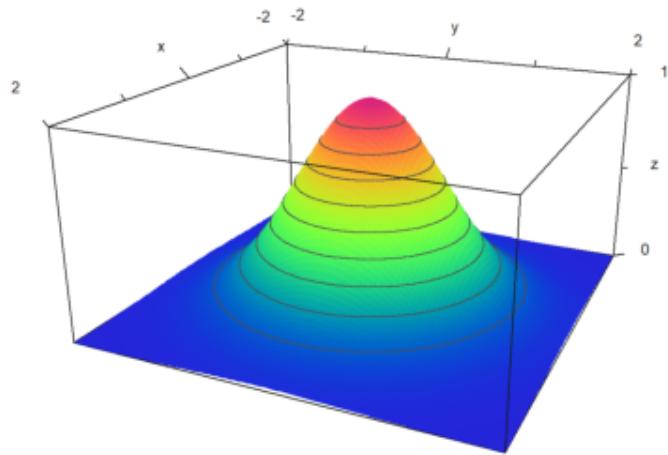
```
>plot3d("x^2+1",a=-1,b=1,rotate=true,grid=5,>user):
```



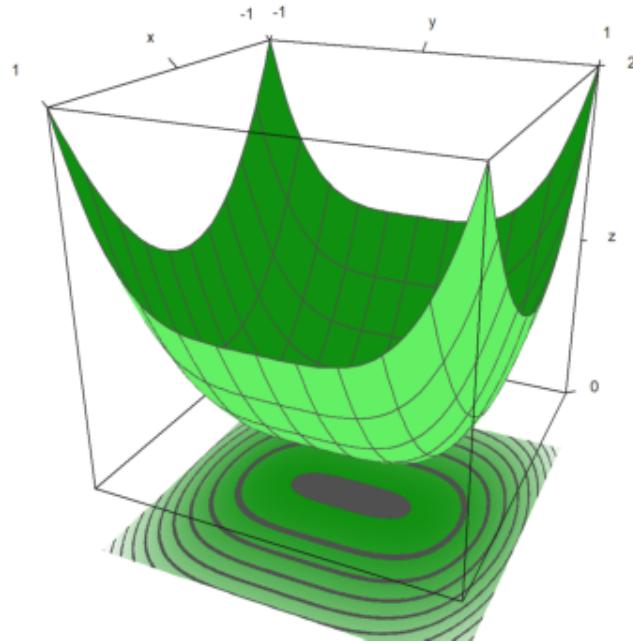
```
>plot3d("1/(x^2+y^2+1)",r=5,>polar, ...
>fscale=2,>hue,n=100,zoom=4,>contour,color=blue,>user):
```



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

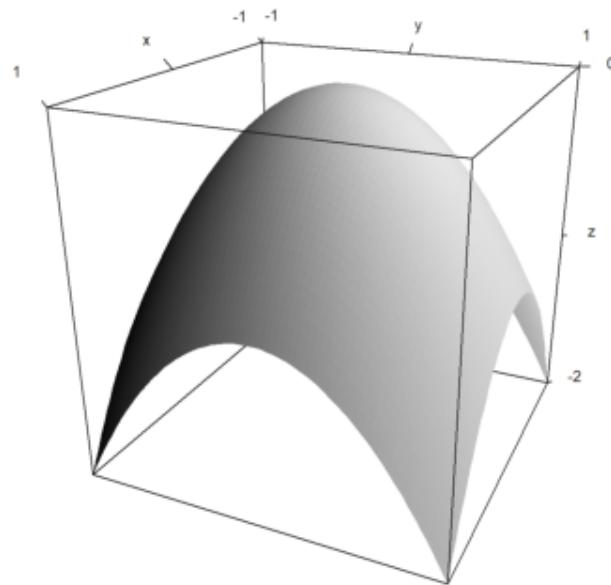


```
>plot3d("x^2+y^4",>cp,cpcolor=green,cpdelta=0.2,>user):
```



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

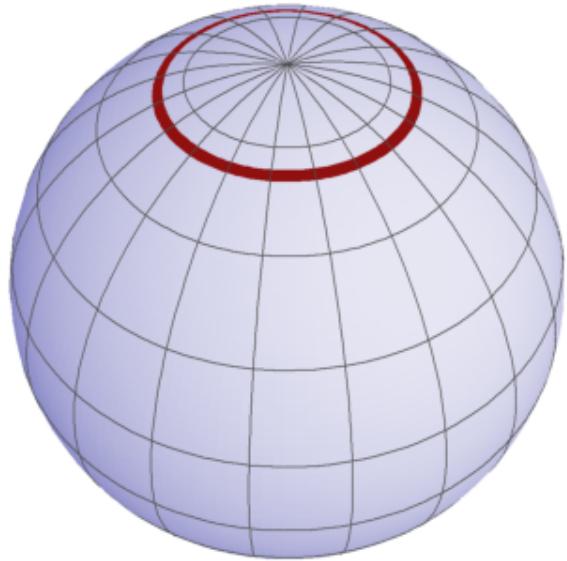
Press I and cursor keys (return to exit)



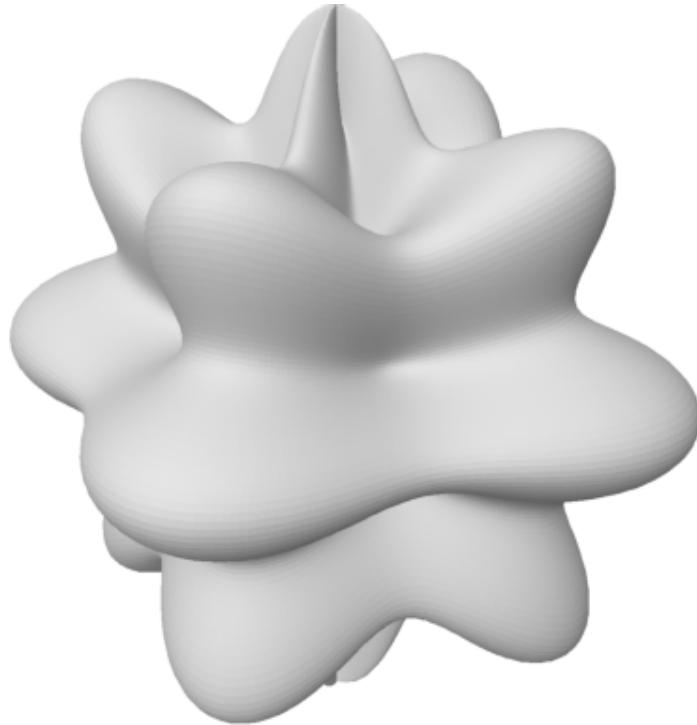
```
>plot3d("-x^2-y^2",color=rgb(0.2,0.2,0),hue=true,frame=false, ...
> zoom=3,contourcolor=red,level=-2:0.1:1,dl=0.01,>user):
```



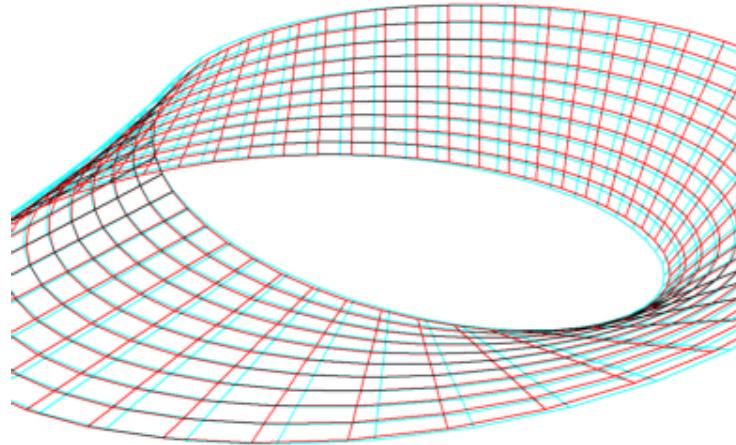
```
>t=linspace(0,2pi,180); s=linspace(-pi/2,pi/2,90)'; ...
>x=cos(s)*cos(t); y=cos(s)*sin(t); z=sin(s); ...
>plot3d(x,y,z,>hue, ...
>color=blue,<frame,grid=[10,20], ...
>values=s,contourcolor=red,level=[90°-24°;90°-22°], ...
>scale=1.4,height=50°,>user):
```



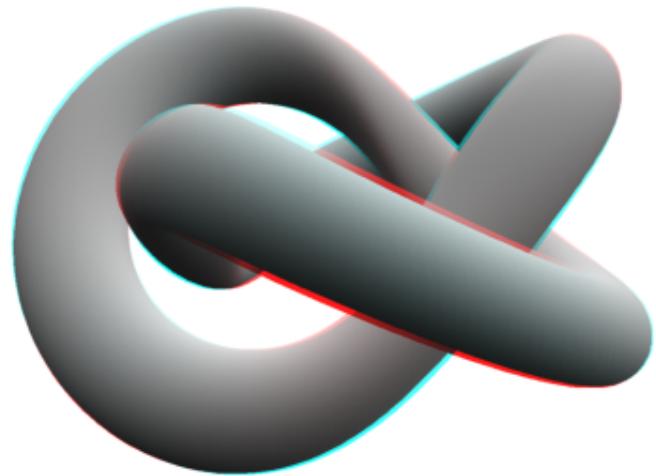
```
>t=linspace(0,2pi,320); s=linspace(-pi/2,pi/2,160)'; ...
>d=1+0.2*(cos(4*t)+cos(8*s)); ...
>plot3d(cos(t)*cos(s)*d,sin(t)*cos(s)*d,sin(s)*d,hue=1, ...
> light=[1,0,1],frame=0,zoom=5,>user):
```



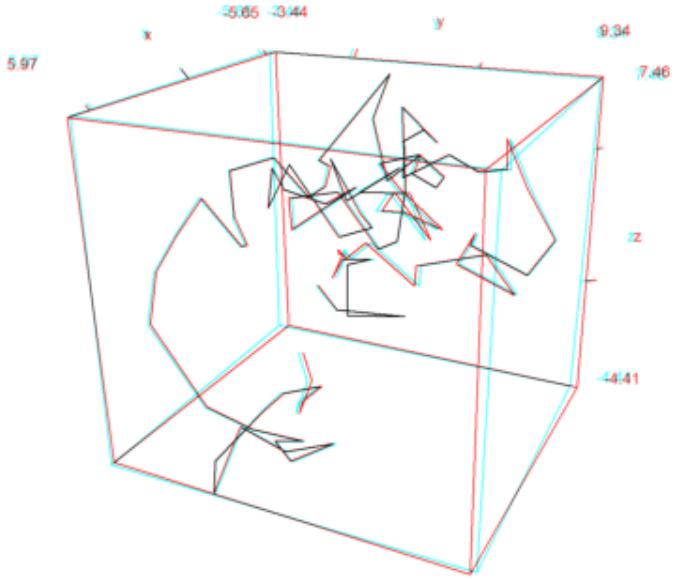
```
>u=linspace(-1,1,10); v=linspace(0,2*pi,50)'; ...
>X=(3+u*cos(v/2))*cos(v); Y=(3+u*cos(v/2))*sin(v); Z=u*sin(v/2); ...
>plot3d(X,Y,Z,>anaglyph,<frame,>wire,scale=2.3):
```



```
>u:=linspace(-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):
```

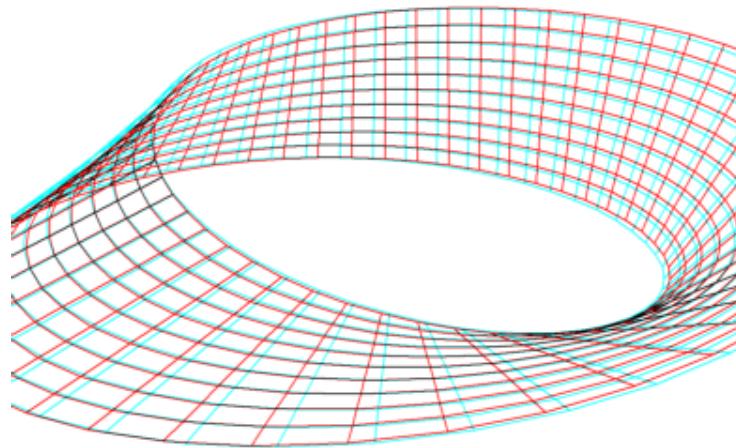


```
>X=cumsum(normal(3,100)); ...
> plot3d(X[1],X[2],X[3],>anaglyph,>wire,>user):
```



```
>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); ...
>function testplot () := plot3d(X,Y,Z,>anaglyph,<frame,>wire,scale=2.3); ...
>rotate("testplot"); testplot():
```

Press space to stop, return to end



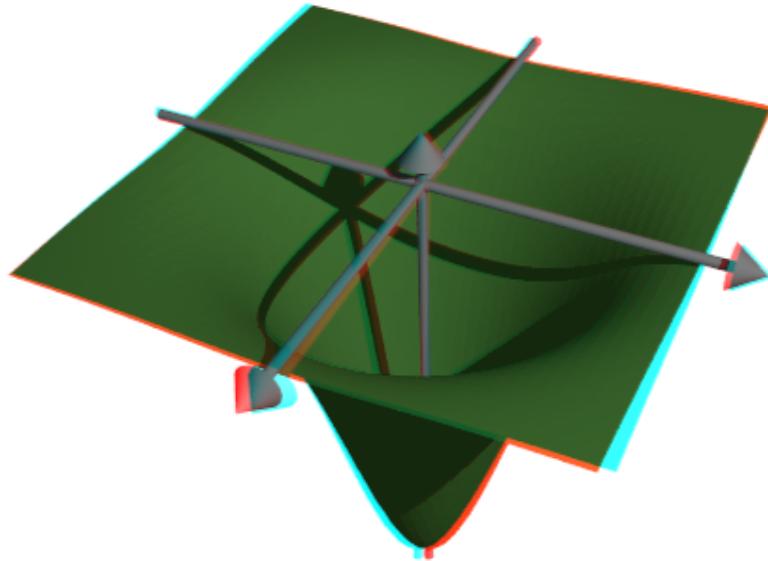
```
>load povray;
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

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

Untuk menghasilkan anaglyph untuk kacamata merah/cyan, Povray harus dijalankan 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 baik. Fungsi pov3d() memiliki saklar sederhana untuk menghasilkan anaglyph.

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



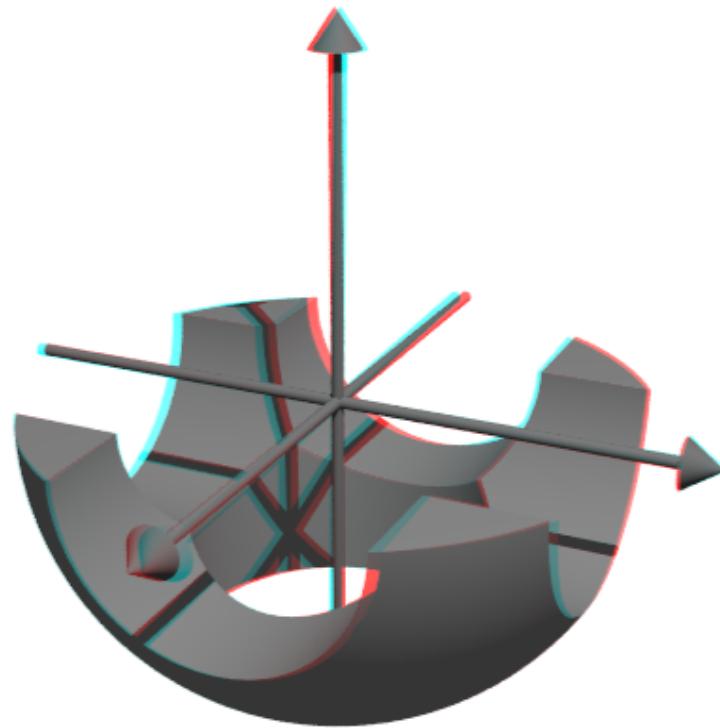
Jika Anda membuat adegan dengan objek, Anda perlu memasukkan generasinya adegan menjadi suatu fungsi, dan jalankan dua kali dengan nilai yang berbeda parameter anaglif.

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

```
s=povsphere(povc,1);
cl=povcylinder(-povz,povz,0.5);
clk=povobject(cl,rotate=xrotate(90°));
cly=povobject(cl,rotate=yrotate(90°));
c=povbox([-1,-1,0],1);
un=povunion([cl,clk,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);
```



---

### 13. Menggambar Diagram Batang Tiga Dimensi

Bar plots/plot batang juga dimungkinkan. Untuk itu, kita harus menyediakannya

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

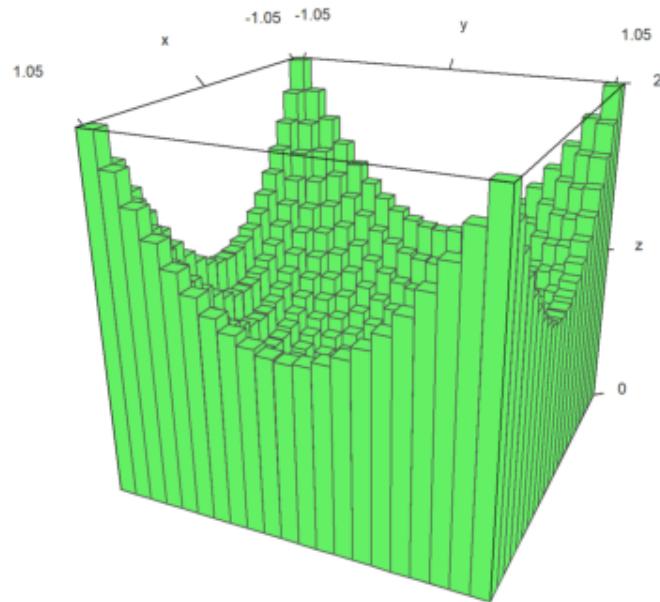
z bisa lebih besar, tetapi hanya nilai nxn yang akan digunakan.

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

### Contoh 1

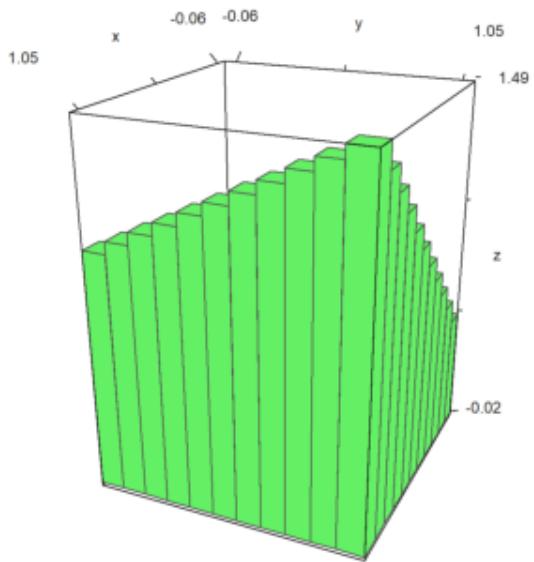
---

```
>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);
```



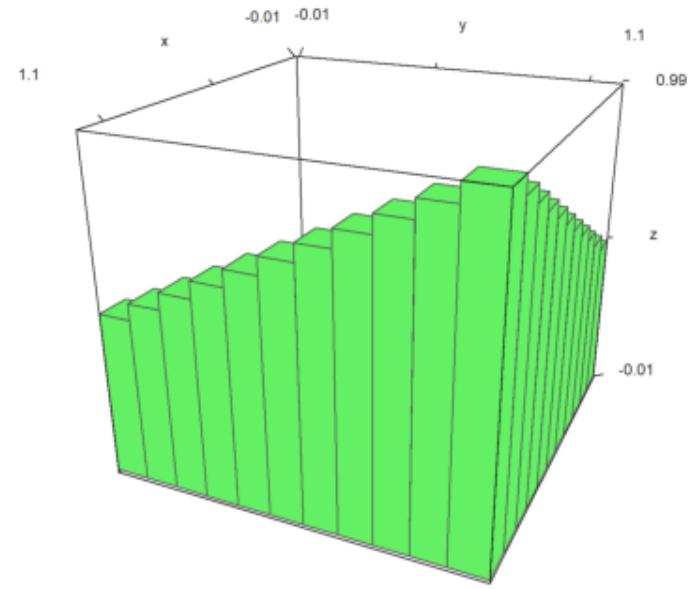
## Contoh 2

```
>x=-0.01:0.1:1; y=x'; z=x+1/2*y; ...
>xa=(x|1.1)-0.05; ya=(y_1.1)-0.05; ...
>plot3d(xa,ya,z,bar=true):
```



### Contoh 3

```
>x=-0.01:0.1:1; y=x'; z=1/2*x+1/2*y; ...
>xa=(x|1.1); ya=(y_1.1); ...
>plot3d(xa,ya,z,bar=true):
```

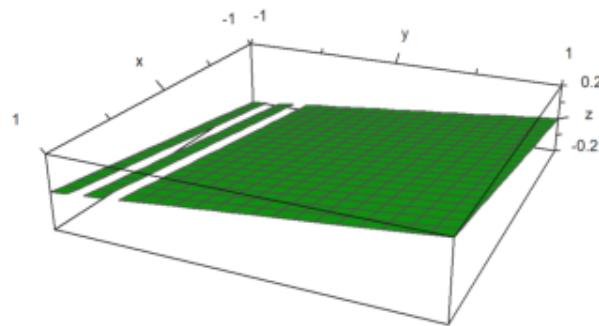


**Contoh 4**

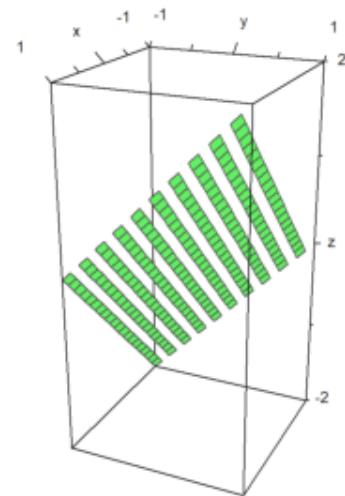
---

Dimungkinkan untuk membagi plot suatu permukaan menjadi dua bagian atau lebih.

```
>x=-1:0.1:1; y=x'; z=1/10*x+1/10*y; d=zeros(size(x)); ...
>plot3d(x,y,z,disconnect=2:2:5);
```



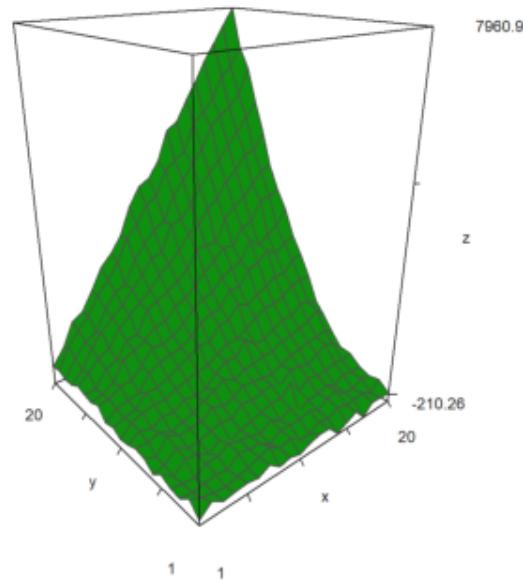
```
>x=-1:0.1:1; y=x'; z=x+y; d=zeros(size(x)); ...
>plot3d(x,y,z,disconnect=2:2:20);
```



Contoh 5

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

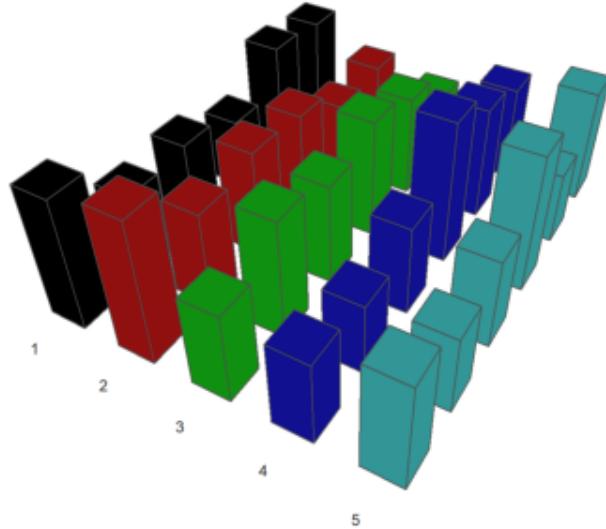
```
>i=1:20; j=i'; ...
>plot3d(i*j^2+100*normal(20,20),>zscale,scale=[1,1,1.5],angle=-40°,zoom=1.8):
```



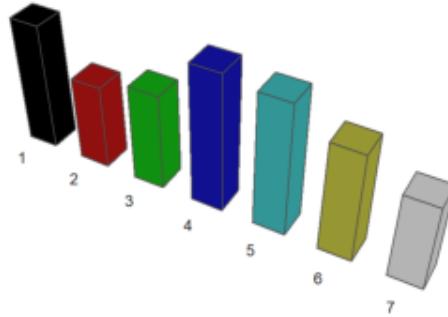
## Contoh 6

---

```
>Z=intrandom(5,100,6); v=zeros(5,6); ...
>loop 1 to 5; v[#]=getmultiplicities(1:6,Z[#]); end; ...
>columnsplot3d(v',scols=1:5,ccols=[1:5]):
```

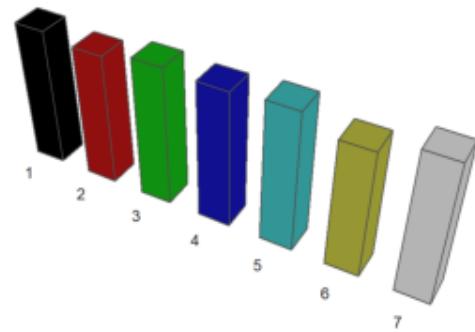


```
>Z=intrandom(7,100,6); v=zeros(7,1); ...
>loop 1 to 7; v[#]=getmultiplicities(1:1,Z[#]); end; ...
>columnsplot3d(v',scols=1:7,ccols=[1:7]):
```



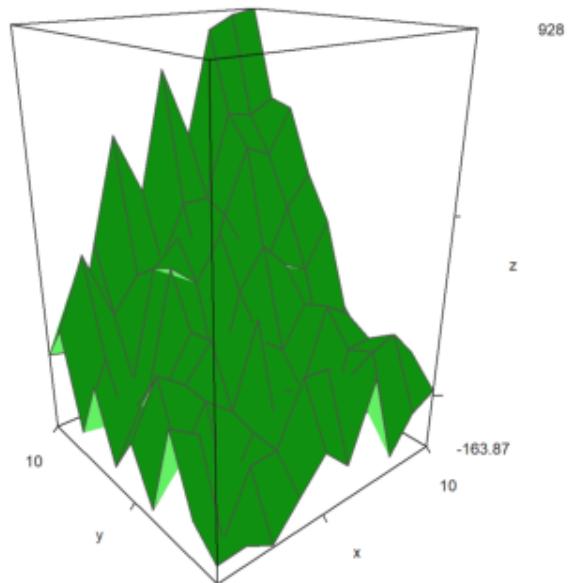
```
>Z=intrandom(7,1000,6); v=zeros(7,1); ...
>loop 1 to 7; v[#]=getmultiplicities(1:1,Z[#]); end; ...
```

```
>columnsplot3d(v',scols=1:7,ccols=[1:7]):
```



**Contoh 7**

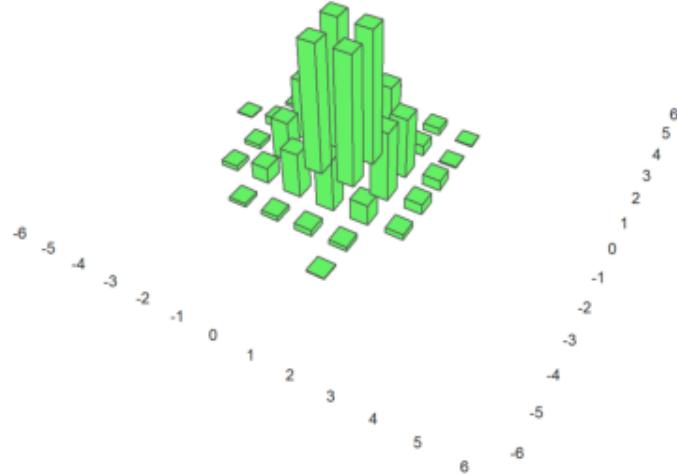
```
>i=1:10; j=i'; ...
>plot3d(i*j^2+100*normal(10,10),>zscale,scale=[1,1,1.5],angle=-40°,zoom=2):
```



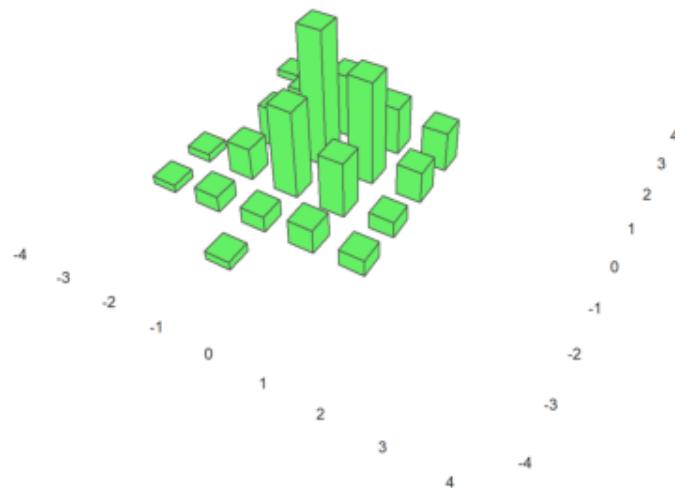
## Contoh 8

---

```
>x=normal(1,1000); y=normal(1,1000); ...
>v=-6:6; z=find2(x,y,v,v); ...
>columnsplot3d(z,v,v,>positive):
```



```
>x=normal(1,100); y=normal(1,100); ...
>v=-4:4; z=find2(x,y,v,v); ...
>columnsplot3d(z,v,v,>positive):
```



## 14. Menggambar Permukaan Benda Putar

---

Permukaan benda putar adalah bagian luar atau lapisan luar dari benda yang berputar atau berotasi. Ketika suatu benda berputar atau berotasi, titik-titik di permukaannya bergerak dalam lintasan melingkar sekitar sumbu rotasi. Permukaan benda putar ini dapat memiliki berbagai bentuk dan sifat tergantung pada jenis benda tersebut.

Misalnya, jika kita membayangkan sebuah bola yang berputar, permukaan bola adalah seluruh lapisan luar yang mengelilingi bola. Permukaan bola ini umumnya berbentuk sferis atau bulat sempurna. Sebaliknya, jika kita memikirkan tentang sebuah benda berbentuk silinder yang berputar, seperti sebuah tabung, permukaan silinder tersebut adalah seluruh bagian luar silinder yang melingkari tubuh silinder tersebut.

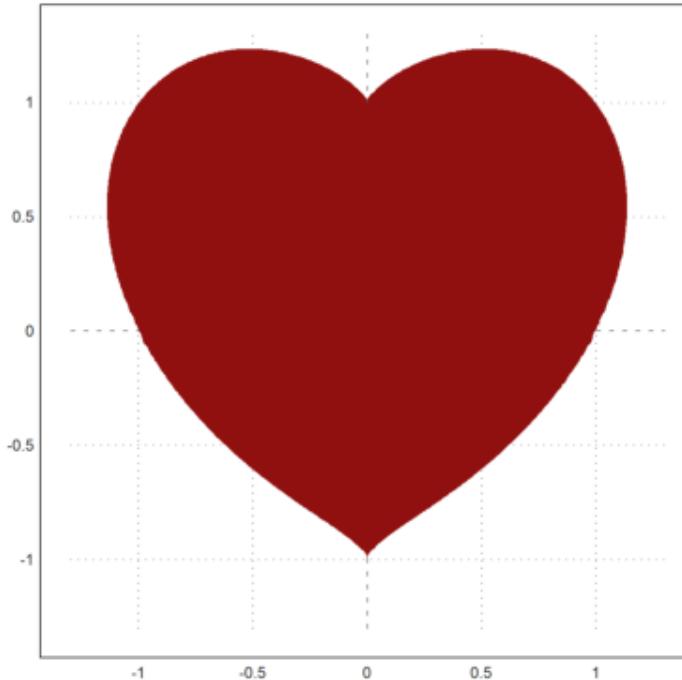
Permukaan benda putar dapat memiliki berbagai sifat, seperti kasar atau halus, berwarna atau transparan, dan bisa terbuat dari berbagai jenis material, seperti logam, plastik, atau kaca. Sifat-sifat ini dapat memengaruhi bagaimana benda berputar berinteraksi dengan lingkungannya, seperti ketika benda tersebut bersentuhan dengan benda lain atau saat dipengaruhi oleh gaya-gaya seperti gaya gesekan.

Dalam ilmu fisika, pemahaman tentang permukaan benda putar dapat menjadi penting dalam konteks berbagai aplikasi, seperti dalam dinamika benda berputar, ilmu material, dan teknik mesin.

Contoh

---

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



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

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

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

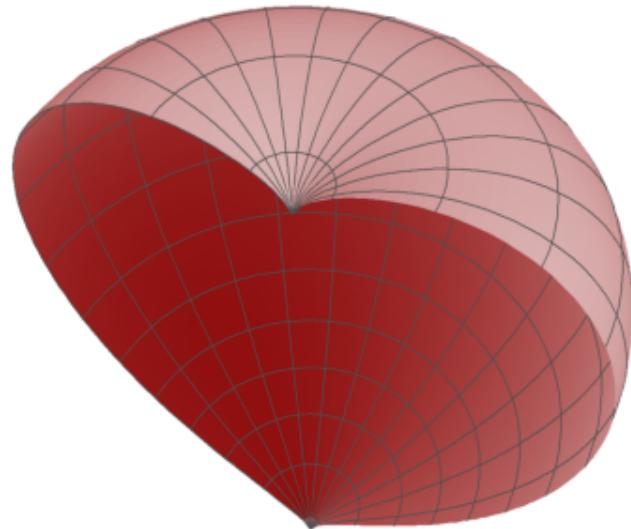
$$x = r \cdot \cos(a), y = r \cdot \sin(a)$$

```
>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.

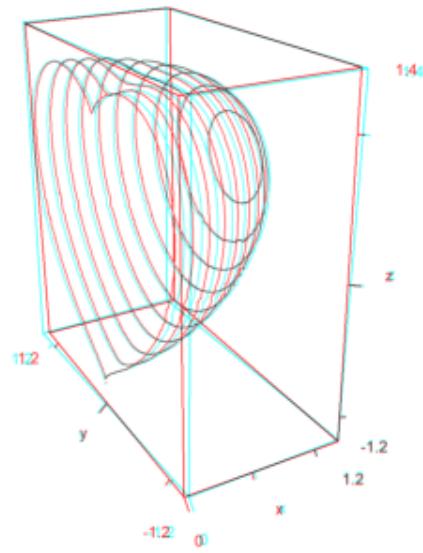
```
>function map f(a) := bisect("fr",0,2;a); ...
>t=linspace(-pi/2,pi/2,100); r=f(t); ...
>s=linspace(pi,2pi,100)'; ...
>plot3d(r*cos(t)*sin(s),r*cos(t)*cos(s),r*sin(t), ...
>>hue,<frame,color=red,zoom=4,amb=0,max=0.7,grid=12,height=50°):
```



```
>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):
```

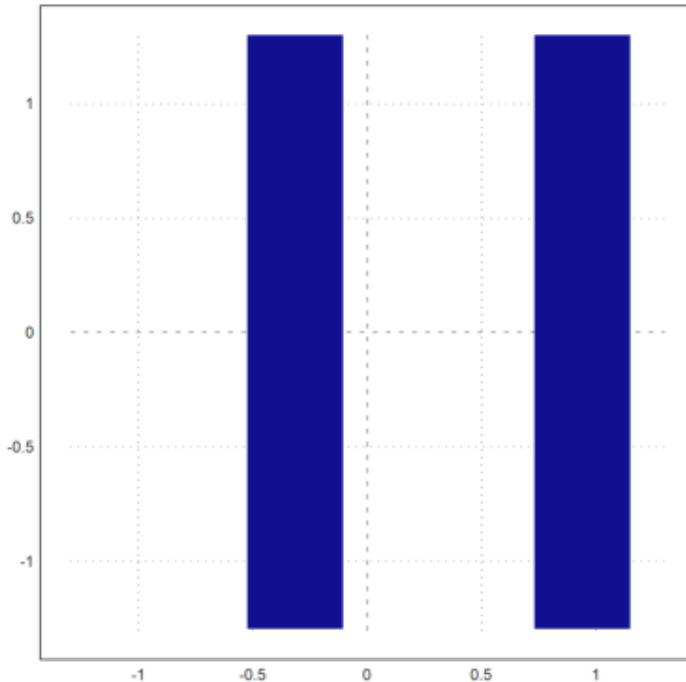


---

Contoh 2

---

```
>plot2d("sin(5x) * 2 + 1",r=1.3, ...
>style="#",color=blue,<outline, ...
>level=[-2;0],n=100):
```



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

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

## 15. Menggambar Grafik 3D dengan Povray di EMT

---

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 subdirektori "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 jalur. 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.  
fault 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

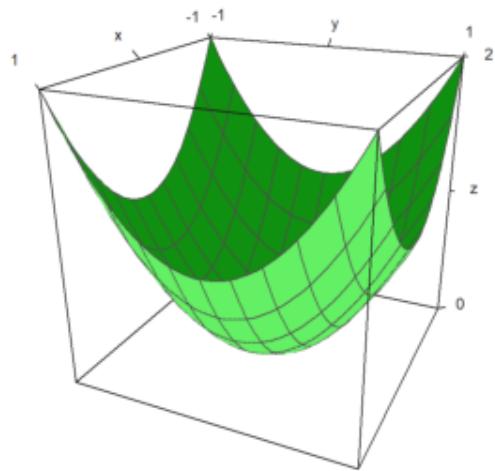
povstart(). Kemudian gunakan writeln(...) untuk menulis objek ke file adegan. Terakhir, akhiri file dengan povend(). Secara default, raytracer akan dimulai, dan PNG akan dimasukkan ke dalam notebook Euler.

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.

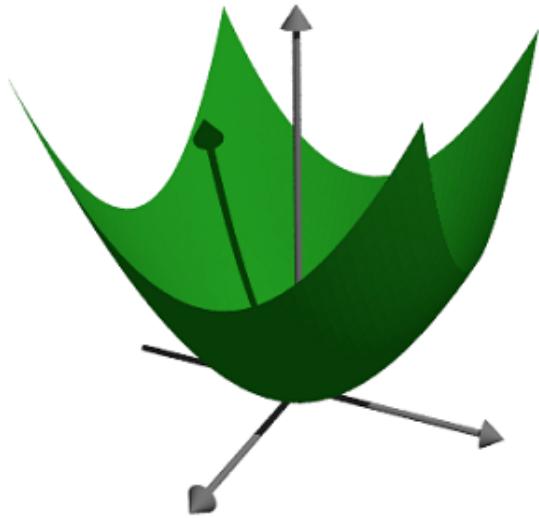
Perhatikan bahwa alam semesta Povray memiliki sistem koordinat lain. Antarmuka ini menerjemahkan semua koordinat ke sistem Povray. Jadi Anda dapat terus berpikir dalam sistem koordinat Euler dengan z menunjuk vertikal ke atas, dan x,y,z sumbu dalam arti tangan kanan.

Anda perlu memuat file povray.

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

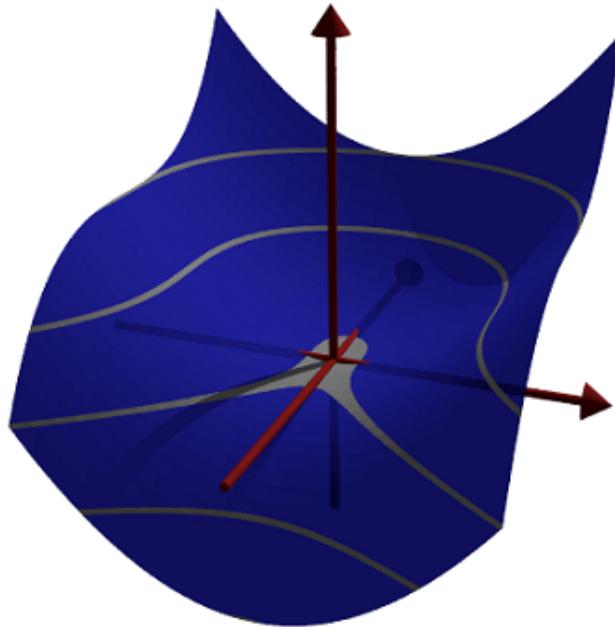


```
>pov3d("x^2+y^2",zoom=3);
```



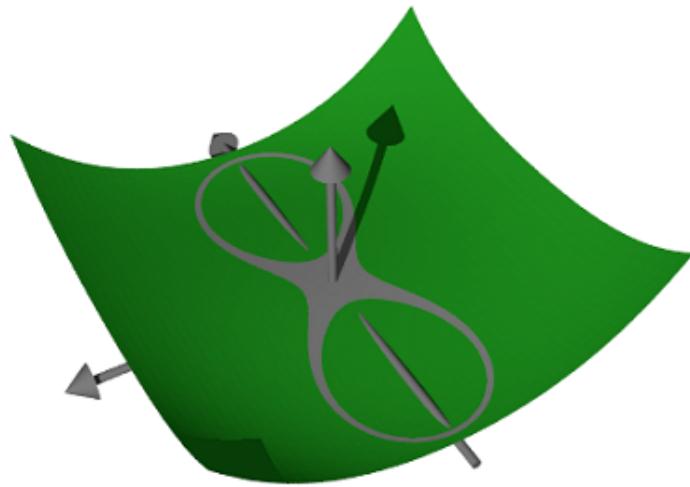
Kita dapat membuat fungsi menjadi transparan dan menambahkan hasil akhir lainnya. Kami juga dapat menambahkan garis level ke plot fungsi.

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



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);
```

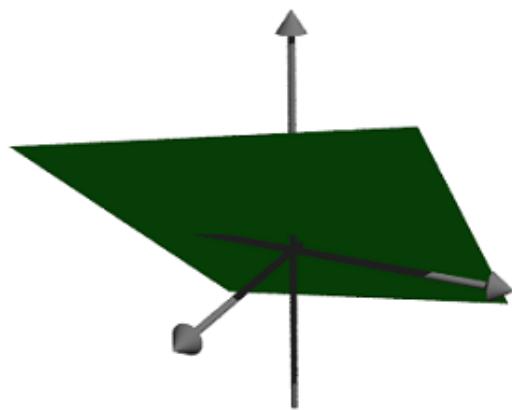


CONTOH

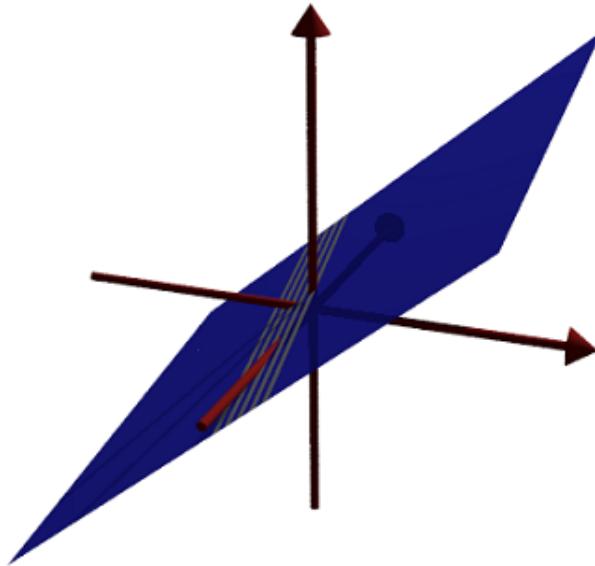
## PART 1

---

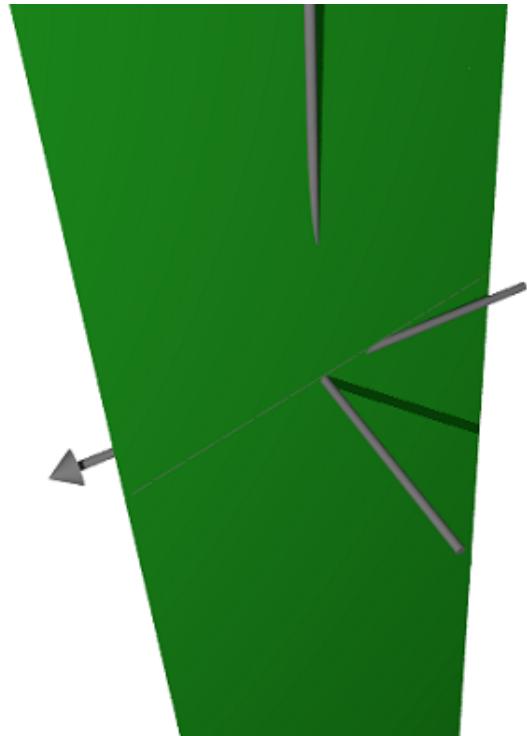
```
>pov3d("10*x+2*y+1",zoom=3);
```



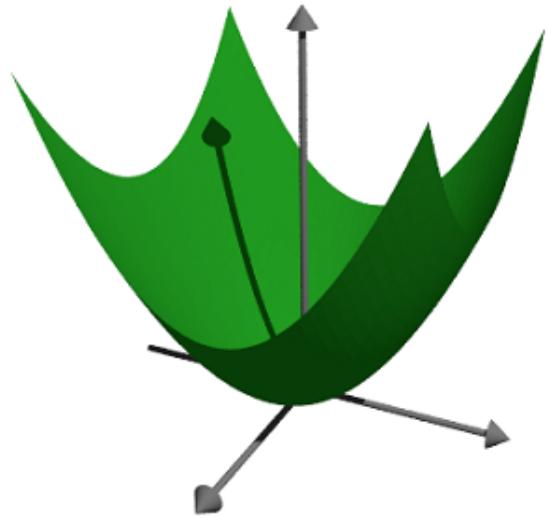
```
>pov3d("10*x+2*y+1",axiscolor=red,angle=20°, ...
>look=povlook(blue,0.2),level=-1:0.5:1,zoom=3.8);
```



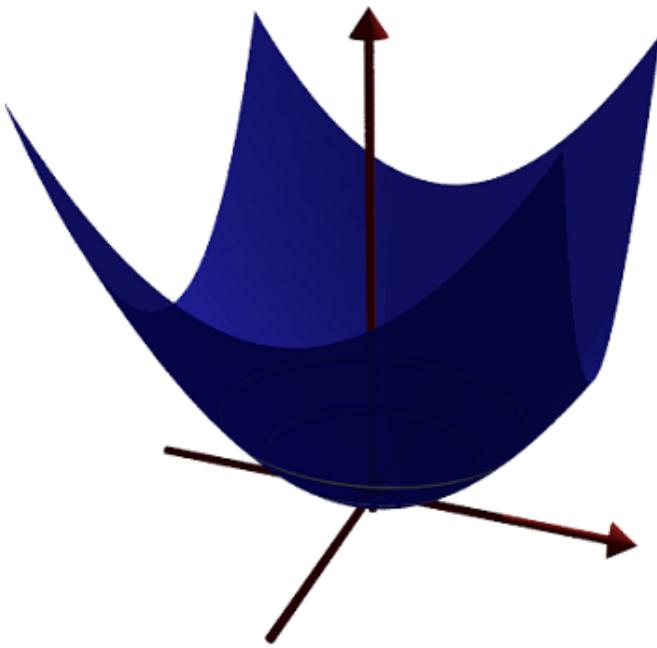
```
>pov3d("10*x+2*y+1",r=1.5, ...
>angle=-120°,level=1/40,dlevel=0.005,light=[-1,1,1],height=45°,n=50, ...
><fscale,zoom=3.8);
```



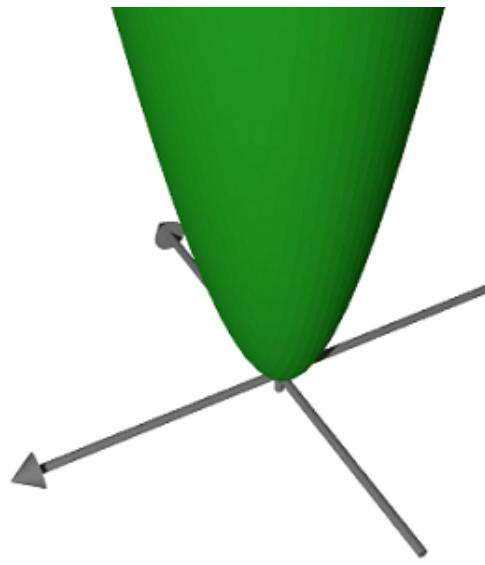
```
>pov3d("2*x^2+3*y^2",zoom=3);
```



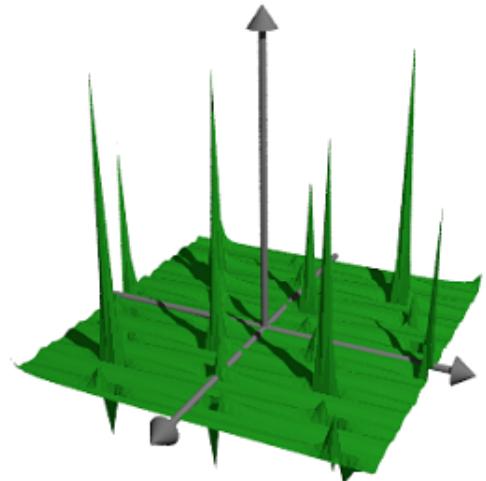
```
>pov3d("2*x^2+3*y^2",axiscolor=red,angle=20°, ...
>look=povlook(blue,0.2),level=-1:0.5:1,zoom=3.8);
```



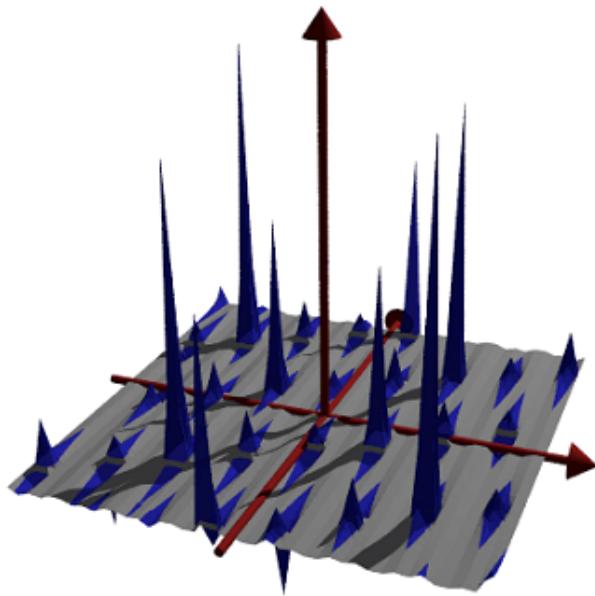
```
>pov3d("2*x^2+3*y^2",r=1.5, ...
>angle=-120°,level=1/40,dlevel=0.005,light=[-1,1,1],height=45°,n=50, ...
><fscale,zoom=3.8);
```



```
>pov3d("sin(x^2+y)+sec(100*x+y)" ,zoom=3);
```



```
>pov3d("sin(x^2+y)+sec(100*x+y)",axiscolor=red,angle=20°, ...
>look=povlook(blue,0.2),level=-1:0.5:1,zoom=3.8);
```



```
>pov3d("sin(x^2+y)+sec(100*x+y)",r=1.5, ...
>angle=-120°,level=1/40,dlevel=0.005,light=[-1,1,1],height=45°,n=50, ...
><fscale,zoom=3.8);
```

