

Neyfa Khalisha Amaluna (23030130014)

Menggambar Plot 3D dengan EMT

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

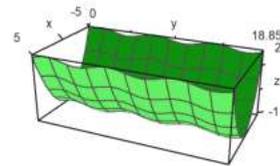
Euler menggambar fungsi tersebut menggunakan algoritma pengurutan untuk menyembunyikan bagian-bagian di latar belakang. Secara umum, Euler menggunakan proyeksi pusat. Standarnya adalah dari kuadran x-y positif ke arah titik asal $x=y=z=0$, tetapi sudut $=0^\circ$ terlihat dari arah sumbu y. Sudut pandang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

Euler dapat memetakan

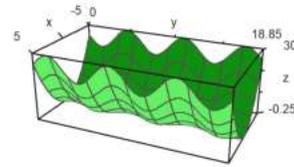
- permukaan dengan bayangan dan garis level atau rentang level,
- awan titik-titik,
- kurva parametrik,
- permukaan implisit.

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

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

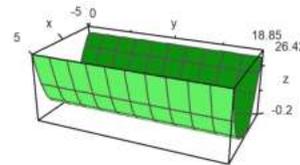


```
>plot3d("x^2+x*sin(y)",-5,5,0,6*pi):
```



Silakan lakukan modifikasi agar gambar "talang bergelombang" tersebut tidak lurus melainkan melengkung/melingkar, baik melingkar secara mendatar maupun melingkar turun/naik (seperti papan peluncur pada kolam renang). Temukan rumusnya.

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



Fungsi dari dua Variabel

Untuk grafik sebuah fungsi, gunakan

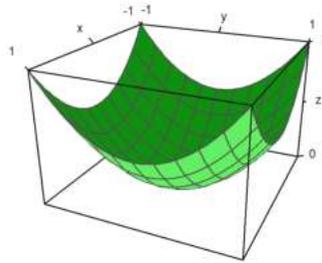
- ekspresi sederhana dalam x dan y,
- nama fungsi dari dua variabel
- atau matriks data.

Standarnya adalah kisi-kisi kawat yang terisi dengan warna yang berbeda di kedua sisi. Perhatikan bahwa jumlah default interval grid adalah 10, namun plot menggunakan jumlah default 40x40 persegi panjang untuk membangun permukaan. Hal ini dapat diubah.

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

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

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

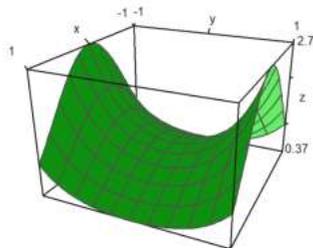


Interaksi pengguna dapat dilakukan dengan parameter `>user`. Pengguna dapat menekan tombol berikut ini.

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

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

Turn with the vector keys (press return to finish)



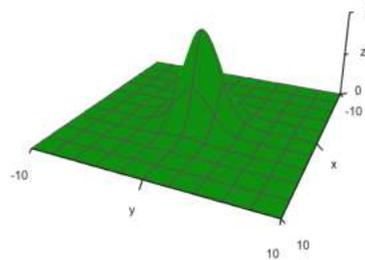
Rentang plot untuk fungsi dapat ditentukan dengan

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

Terdapat beberapa parameter untuk menskalakan fungsi atau mengubah tampilan grafik.

`fscale`: skala untuk nilai fungsi (standarnya adalah `<fscale>`).
`scale`: angka atau vektor 1x2 untuk menskalakan ke arah x dan y.
`frame`: jenis bingkai (default 1).

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



Tampilan dapat diubah dengan berbagai cara.

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

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

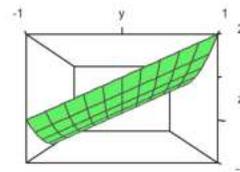
```
>view
```

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

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

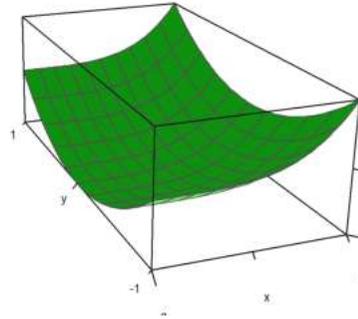
Dalam contoh berikut ini, sudut = 0 dan tinggi = 0 terlihat dari sumbu y negatif. Label sumbu untuk y disembunyikan dalam kasus ini.

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



Plot terlihat selalu ke bagian tengah kubus plot. Anda dapat memindahkan bagian tengah dengan parameter `center`.

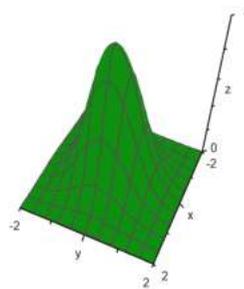
```
>plot3d("x^4+y^2", a=0, b=1, c=-1, d=1, angle=-20°, height=20°, ...
center=[0.4, 0, 0], zoom=5) :
```



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

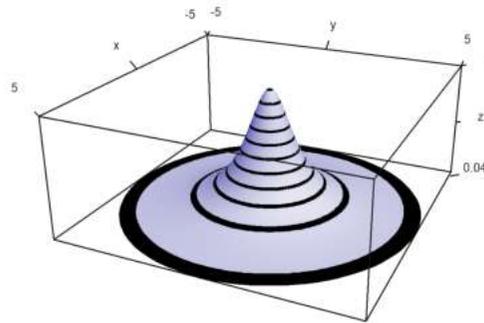
Jika Anda menonaktifkannya dengan `scale=false`, Anda harus berhati-hati agar plot tetap muat di dalam jendela plotting, dengan mengubah jarak tampilan atau zoom, dan memindahkan bagian tengahnya.

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

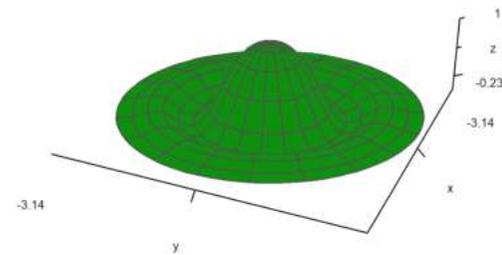


Plot polar juga tersedia. Parameter `polar=true` menggambar plot polar. Fungsi harus tetap merupakan fungsi dari x dan y . Parameter "`fscale`" menskalakan fungsi dengan skala sendiri. Jika tidak, fungsi akan diskalakan agar sesuai dengan kubus.

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



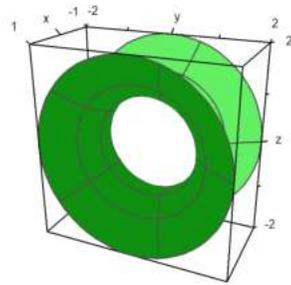
```
>function f(r) := exp(-r/2)*cos(r); ...
plot3d("f(x^2+y^2)",>polar,scale=[1,1,0.4],r=pi,frame=3,zoom=4):
```



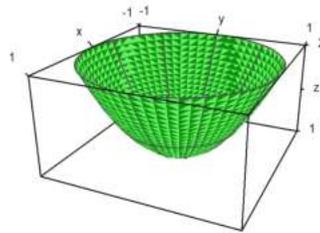
Parameter rotate memutar fungsi dalam x di sekitar sumbu x.

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

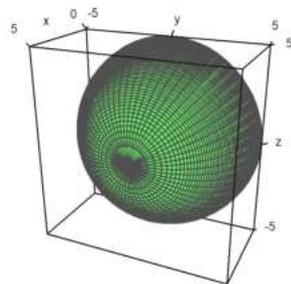
```
>plot3d("x^2+1",a=-1,b=1,rotate=true,grid=5):
```



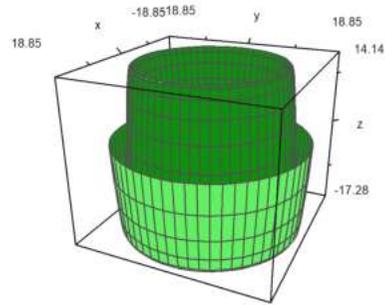
```
>plot3d("x^2+1", a=-1, b=1, rotate=2, grid=5) :
```



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

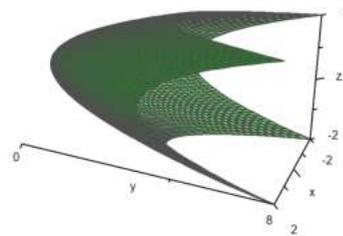


```
>plot3d("x*sin(x)", a=0, b=6pi, rotate=2) :
```



Berikut ini adalah plot dengan tiga fungsi.

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



Plot Kontur

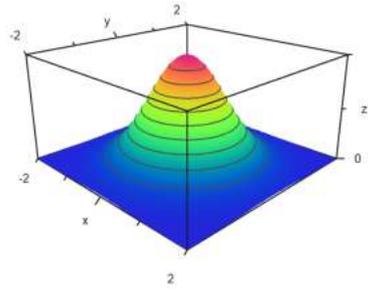
Untuk plot, Euler menambahkan garis kisi-kisi. Sebagai gantinya, dimungkinkan untuk menggunakan garis level dan rona satu warna atau rona berwarna spektral. Euler dapat menggambar ketinggian fungsi pada plot dengan bayangan. Pada semua plot 3D, Euler dapat menghasilkan anaglyph merah/cyan.

- Rona: Mengaktifkan bayangan cahaya, bukan kabel.
- >contour: Memplot garis kontur otomatis pada plot.
- level=... (atau level): Vektor nilai untuk garis kontur.

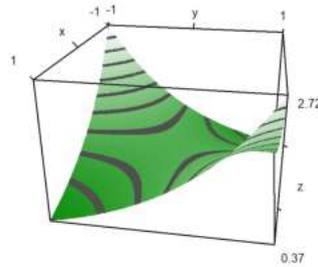
Defaultnya adalah level="auto", yang menghitung beberapa garis level secara otomatis. Seperti yang Anda lihat di plot, level sebenarnya adalah rentang level.

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

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



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

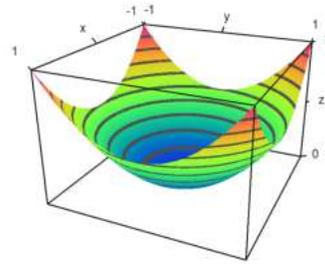


Bayangan default menggunakan warna abu-abu. Tetapi, kisaran warna spektral juga tersedia.

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

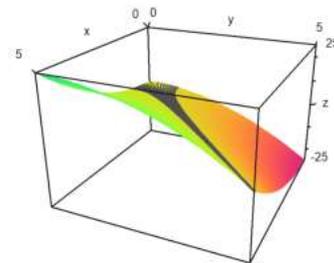
Untuk plot berikut ini, kami menggunakan skema spektral default dan menambah jumlah titik untuk mendapatkan tampilan yang sangat mulus.

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



Alih-alih garis level otomatis, kita juga dapat menetapkan nilai garis level. Hal ini akan menghasilkan garis level yang tipis, alih-alih rentang level.

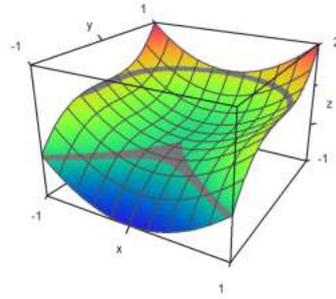
```
>plot3d("x^2-y^2",0,5,0,5,level=-1:0.1:1,color=redgreen):
```



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

Selain itu, kami menghamparkan 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):
```

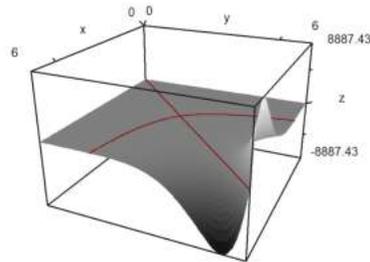


Pada contoh berikut, kami memplot himpunan, di mana

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

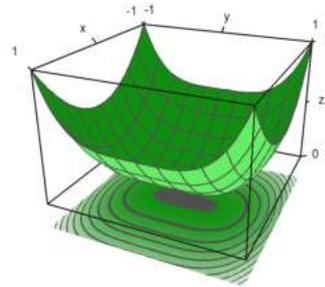
Kita menggunakan satu garis tipis untuk garis level.

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



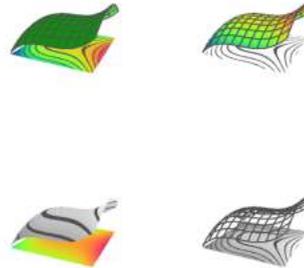
Dimungkinkan untuk menampilkan bidang kontur di bawah plot. Warna dan jarak ke plot dapat ditentukan.

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



Berikut ini beberapa gaya lainnya. Kami selalu mematikan bingkai, dan menggunakan berbagai skema warna untuk plot dan kisi-kisi.

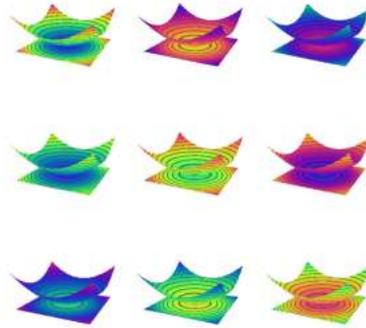
```
>figure(2,2); ...
expr="y^3-x^2"; ...
figure(1); ...
    plot3d(expr,<frame,>cp,cpcolor=spectral); ...
figure(2); ...
    plot3d(expr,<frame,>spectral,grid=10,cp=2); ...
figure(3); ...
    plot3d(expr,<frame,>contour,color=gray,nc=5,cp=3,cpcolor=greenred); ...
figure(4); ...
    plot3d(expr,<frame,>hue,grid=10,>transparent,>cp,cpcolor=gray);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



Ada beberapa skema spektral lainnya, yang diberi nomor dari 1 hingga 9. Tetapi Anda juga dapat menggunakan `color=value`, di mana `value`

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

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



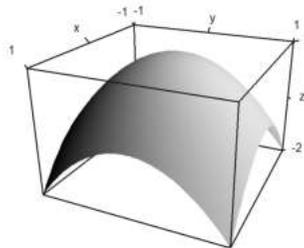
Sumber cahaya dapat diubah dengan I dan tombol kursor selama interaksi pengguna. Ini juga dapat ditetapkan dengan parameter.

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

Perhatikan, bahwa program ini tidak membuat perbedaan di antara sisi-sisi plot. Tidak ada bayangan. Untuk ini Anda akan membutuhkan Povray.

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

Press I and cursor keys (return to exit)



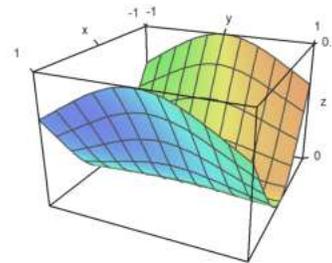
Parameter warna mengubah warna permukaan. Warna garis level juga dapat diubah.

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



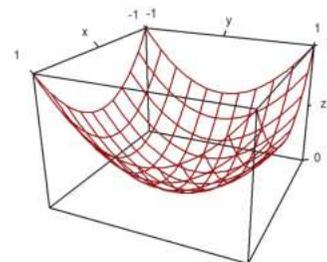
Warna 0 memberikan efek pelangi yang istimewa.

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



Permukaannya juga bisa transparan.

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



Plot Implisit

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

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

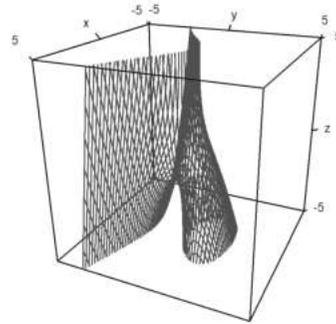
dapat divisualisasikan dalam potongan yang sejajar dengan bidang x-y, bidang x-z, dan bidang y-z.

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

Tambahkan nilai-nilai ini, jika Anda mau. Pada contoh, kami memplot

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

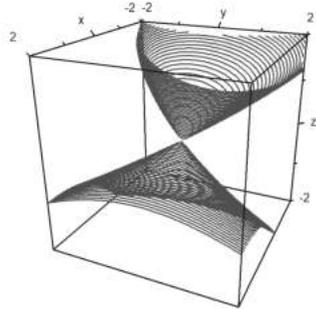
```
>plot3d("x^2+y^3+z*y-1",r=5,implicit=3):
```



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



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



Memplot Data 3D

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

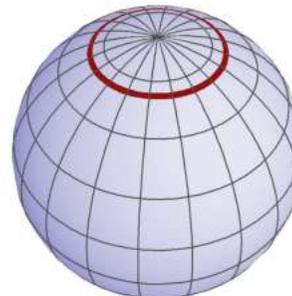
$$\gamma(t, s) = (x(t, s), y(t, s), z(t, s))$$

Karena x,y,z adalah matriks, kita mengasumsikan bahwa (t,s) berjalan melalui kotak persegi. Hasilnya, Anda dapat memplot gambar persegi panjang dalam ruang.

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

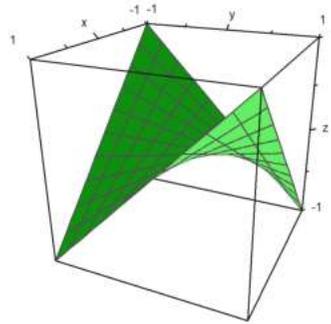
Pada contoh berikut, kita menggunakan vektor nilai t dan vektor kolom nilai s untuk memparameterkan permukaan bola. Pada gambar kita dapat menandai daerah, dalam kasus kita daerah kutub.

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



Berikut ini adalah contoh, yang merupakan grafik suatu fungsi.

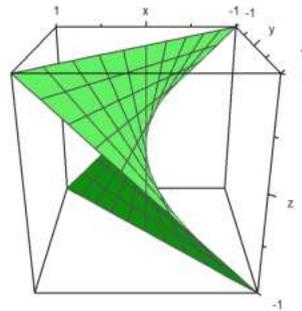
```
>t=-1:0.1:1; s=(-1:0.1:1)'; plot3d(t,s,t*s,grid=10):
```



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

$$x = yz$$

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



Dengan lebih banyak upaya, kita bisa menghasilkan banyak permukaan.

Dalam contoh berikut ini, kami membuat tampilan berbayang dari bola yang terdistorsi. Koordinat yang biasa digunakan untuk bola adalah

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

dengan

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

Kami mengubahnya dengan sebuah faktor

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

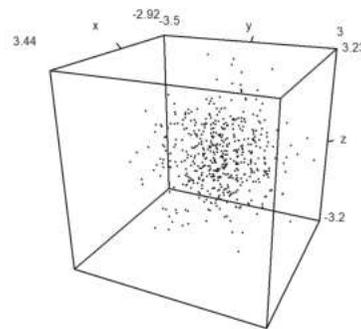
```
>t=linspace(0,2pi,320); s=linspace(-pi/2,pi/2,160)'; ...
d=1+0.2*(cos(4*t)+cos(8*s)); ...
plot3d(cos(t)*cos(s)*d,sin(t)*cos(s)*d,sin(s)*d,hue=1, ...
light=[1,0,1],frame=0,zoom=5):
```



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

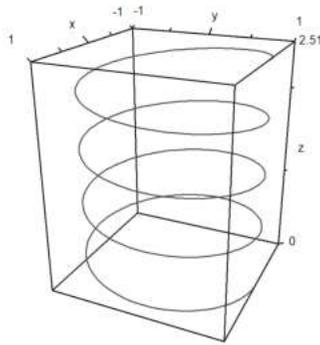
Gaya-gayanya sama seperti pada plot2d dengan `points=true`;

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

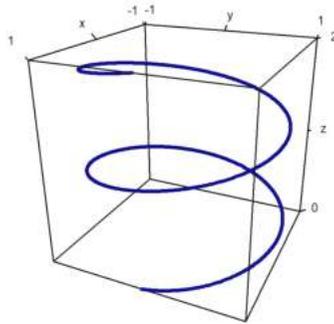


Anda juga dapat memplot kurva dalam bentuk 3D. Dalam hal ini, akan lebih mudah untuk menghitung titik-titik kurva. Untuk kurva pada bidang, kami menggunakan urutan koordinat dan parameter `wire = true`.

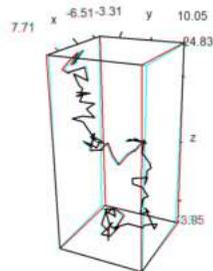
```
>t=linspace(0,8pi,500); ...  
plot3d(sin(t),cos(t),t/10,>wire, zoom=3):
```



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

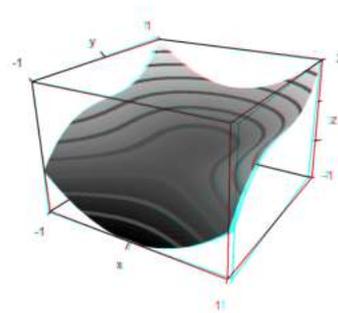


```
>X=cumsum(normal(3,100)); ...
plot3d(X[1],X[2],X[3],>anaglyph,>wire):
```



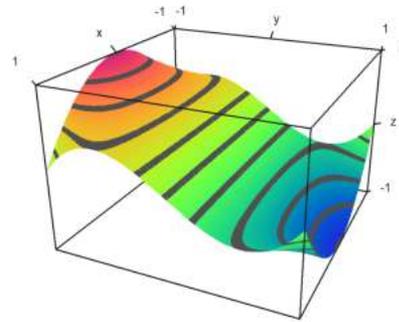
EMT juga dapat membuat plot dalam mode anaglyph. Untuk melihat plot semacam itu, Anda memerlukan kacamata merah/cyan.

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



Sering kali, skema warna spektral digunakan untuk plot. Hal ini menekankan ketinggian fungsi.

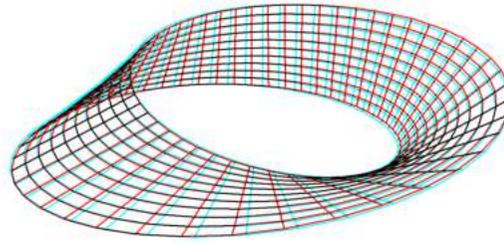
```
>plot3d("x^2*y^3-y",>spectral,>contour, zoom=3.2):
```



Euler juga dapat memplot permukaan yang diparameterkan, ketika parameternya adalah nilai x -, y -, dan z - dari gambar kisi-kisi persegi panjang di dalam ruang.

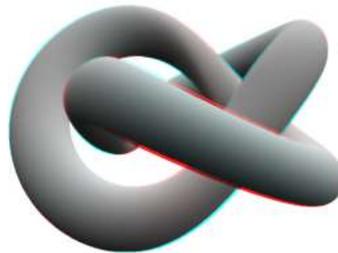
Untuk demo berikut ini, kami menyiapkan parameter u - dan v -, dan menghasilkan koordinat ruang dari parameter ini.

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



Berikut ini contoh yang lebih rumit, yang tampak megah dengan kacamata merah/cyan.

```
>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):
```



Plot Statistik

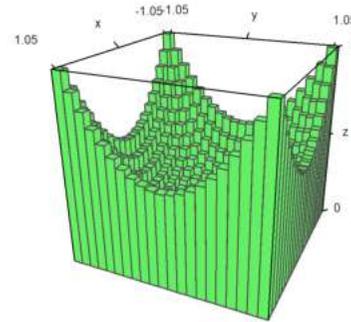
Plot batang juga dapat digunakan. Untuk ini, kita harus menyediakan

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

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

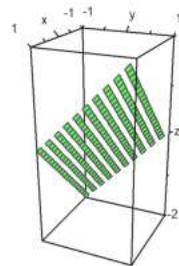
Pada contoh, pertama-tama kita menghitung nilainya. Kemudian kita sesuaikan x dan y, sehingga vektor berada di tengah-tengah nilai yang digunakan.

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



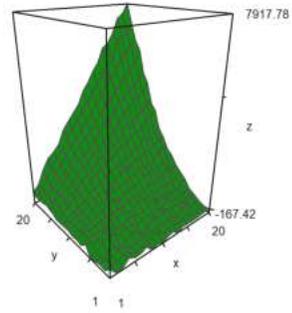
Dimungkinkan untuk membagi plot permukaan menjadi dua bagian atau lebih.

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

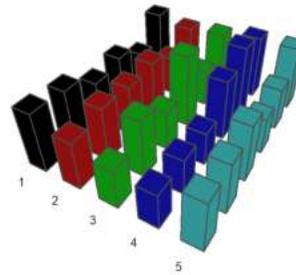


Jika memuat atau menghasilkan matriks data M dari file dan perlu memplotnya dalam 3D, Anda dapat menskalakan matriks ke $[-1,1]$ dengan `scale(M)`, atau menskalakan matriks dengan `>zscale`. Hal ini dapat dikombinasikan dengan faktor penskalaan individual yang diterapkan sebagai tambahan.

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

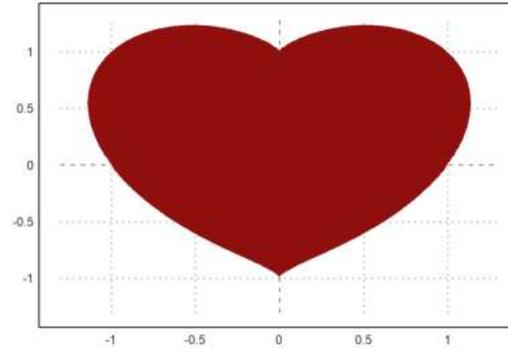


```
>Z=intrandom(5,100,6); v=zeros(5,6); ...
loop 1 to 5; v[#]=getmultiplicities(1:6,Z[#]); end; ...
columnsp3d(v',scols=1:5,ccols=[1:5]):
```



Permukaan Benda Putar

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



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

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

Kami ingin memutar kurva jantung di sekitar sumbu y. Berikut ini adalah ekspresi yang mendefinisikan jantung:

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

Selanjutnya kita atur

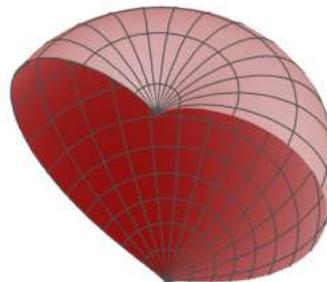
$$x = r \cdot \cos(a), \quad y = r \cdot \sin(a).$$

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

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

Hal ini memungkinkan untuk mendefinisikan fungsi numerik, yang menyelesaikan untuk r, jika a diberikan. Dengan fungsi tersebut kita dapat memplotkan jantung yang diputar sebagai permukaan parametrik.

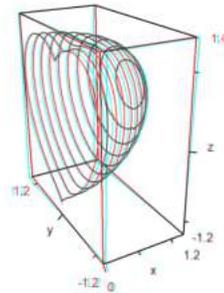
```
>function map f(a) := bisect("fr",0,2;a); ...
t=linspace(-pi/2,pi/2,100); r=f(t); ...
s=linspace(pi,2pi,100)'; ...
plot3d(r*cos(t)*sin(s),r*cos(t)*cos(s),r*sin(t), ...
>hue,<frame,color=red, zoom=4, amb=0, max=0.7, grid=12, height=50°):
```



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

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

>plot3d("f(x,y,z)", ...
xmin=0,xmax=1.2,ymin=-1.2,ymax=1.2,zmin=-1.2,zmax=1.4, ...
implicit=1,angle=-30°,zoom=2.5,n=[10,100,60],>anaglyph):
```



Plot 3D Khusus

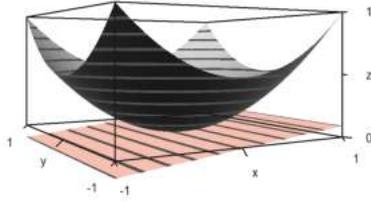
Fungsi plot3d memang bagus untuk dimiliki, tetapi tidak memenuhi semua kebutuhan. Selain rutinitas yang lebih mendasar, Anda juga bisa mendapatkan plot berbingkai dari objek apa pun yang Anda sukai.

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

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

Sekarang framedplot() menyediakan frame, dan mengatur tampilan.

```
>framedplot("myplot",[-1,1,-1,1,0,1],height=0,angle=-30°, ...
center=[0,0,-0.7],zoom=3):
```

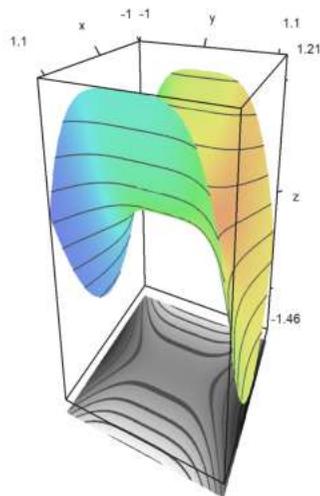


Dengan cara yang sama, Anda dapat memplot bidang kontur secara manual. Perhatikan bahwa `plot3d()` mengatur jendela ke `fullwindow()` secara default, namun `plotcontourplane()` mengasumsikannya.

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

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

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

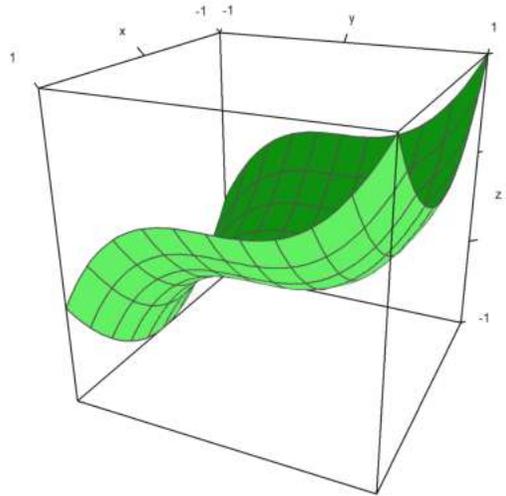


Euler dapat menggunakan frame untuk melakukan pra-komputasi animasi.

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

Silakan pelajari sumber dari rotate untuk melihat lebih detail.

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



Menggambar Povray

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

Anda perlu menginstal Povray (32bit atau 64bit) dari <http://www.povray.org/>, dan meletakkan sub-direktori "bin" dari Povray ke dalam jalur lingkungan, atau mengatur variabel "defaultpovray" dengan jalur lengkap yang mengarah 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 defaultnya adalah eulerhome(), biasanya c:\Users\Username\Euler. Povray menghasilkan sebuah file PNG, yang dapat dimuat oleh Euler ke dalam notebook. Untuk membersihkan berkas-berkas ini, gunakan povclear().

Fungsi pov3d memiliki semangat yang sama dengan plot3d. Fungsi ini dapat menghasilkan grafik dari sebuah fungsi $f(x,y)$, atau sebuah permukaan dengan koordinat X,Y,Z dalam bentuk matriks, termasuk garis-garis level yang bersifat 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-fungsi ini, mulai file Povray dengan povstart(). Kemudian gunakan writeln(...) untuk menulis objek ke file scene. Terakhir, akhiri file dengan povend(). Secara default, raytracer akan dimulai, dan PNG akan dimasukkan ke dalam buku catatan 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. Fungsi ini memiliki parameter untuk warna, transparansi, Phong Shading, dll.

Perhatikan bahwa Povray universe memiliki sistem koordinat lain. Antarmuka ini menerjemahkan semua koordinat ke sistem Povray. Jadi Anda dapat tetap berpikir dalam sistem koordinat Euler dengan z yang mengarah vertikal ke atas, dan sumbu x, y, z di tangan kanan. Anda perlu memuat file povray.

```
>load povray
```

```
Functions to generate Povray files for the open Raytracer.
```

Pastikan direktori bin povray berada di dalam path. Jika tidak, edit variabel berikut sehingga berisi jalur ke povray yang dapat dieksekusi.

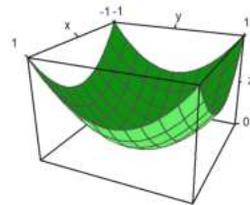
```
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

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

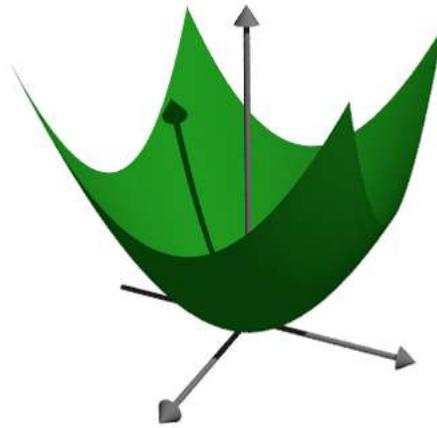
Untuk kesan pertama, kita plot sebuah fungsi sederhana. Perintah berikut ini menghasilkan file povray di direktori pengguna Anda, dan menjalankan Povray untuk melacak sinar pada file ini.

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 dijalankan. Anda dapat menekan cancel untuk menghentikan pertanyaan lebih lanjut. Anda mungkin harus menekan OK pada jendela Povray untuk mengetahui dialog awal Povray.

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

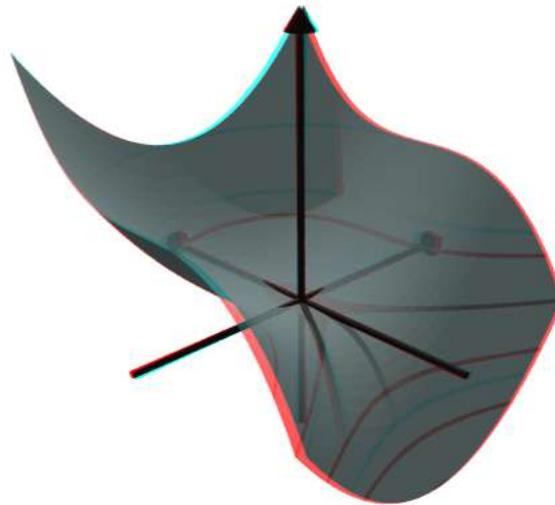


```
>pov3d("x^2+y^2", zoom=3) ;
```



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

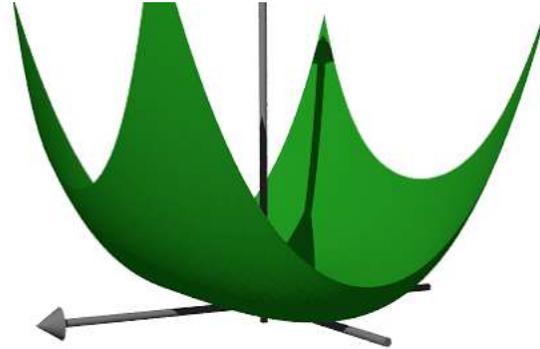
```
>pov3d("x^2+y^3",axiscolor=red,angle=-45°,>anaglyph, ...  
look=povlook(cyan,0.2),level=-1:0.5:1,zoom=3.8);
```



Kadang-kadang perlu untuk mencegah penskalaan fungsi, dan menskalakan fungsi dengan tangan.

Kami memplot kumpulan titik pada bidang kompleks, di mana hasil kali 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=3.8);
```



Merencanakan dengan Koordinat

Sebagai pengganti fungsi, kita dapat membuat plot dengan koordinat. Seperti pada plot3d, kita membutuhkan tiga matriks untuk mendefinisikan objek.

Pada contoh, kita memutar sebuah fungsi pada sumbu z.

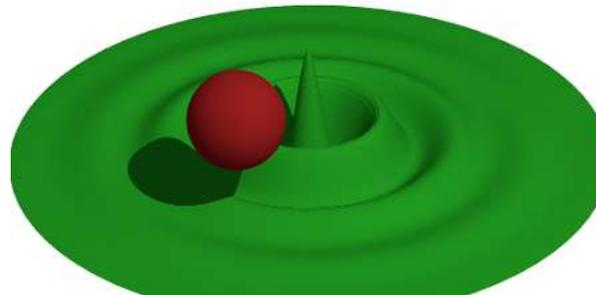
```
>function f(x) := x^3-x+1; ...  
x=-1:0.01:1; t=linspace(0,2pi,50)'; ...  
Z=x; X=cos(t)*f(x); Y=sin(t)*f(x); ...  
pov3d(X,Y,Z,angle=40°,look=povlook(red,0.1),height=50°,axis=0,zoom=4,light=[10,5,15]);
```



Pada contoh berikut, kita memplot gelombang teredam. Kami menghasilkan gelombang dengan bahasa matriks Euler.

Kami juga menunjukkan, bagaimana objek tambahan dapat ditambahkan ke adegan pov3d. Untuk pembuatan objek, lihat contoh berikut. Perhatikan bahwa plot3d menskalakan plot, sehingga sesuai 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=6, axis=0, height=30°, add=povsphere([0.5,0,0.25],0.15,povlook(red)), ...
w=500,h=300);
```



Dengan metode bayangan canggih Povray, hanya sedikit titik yang bisa menghasilkan permukaan yang sangat halus. Hanya pada batas-batas dan bayangan, trik ini bisa terlihat jelas.

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

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

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

Persamaan permukaannya adalah $[x,y,z]$. Kami menghitung dua turunan terhadap x dan y dari persamaan ini dan mengambil hasil perkalian silang sebagai normal.

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

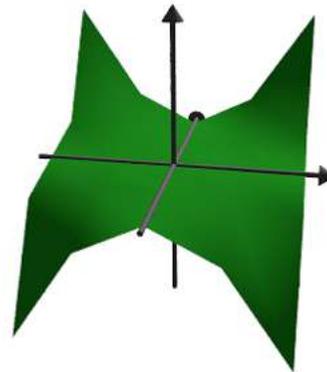
Kami mendefinisikan normal sebagai hasil kali silang dari turunan ini, dan mendefinisikan fungsi koordinat.

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

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

Kami hanya menggunakan 25 poin.

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



Berikut ini adalah simpul Trefoil yang dibuat oleh A. Busser di Povray. Ada versi yang lebih baik dari ini dalam contoh.

Trefoil Knot

Untuk tampilan yang bagus dengan tidak terlalu banyak titik, kita tambahkan vektor normal di sini. Kita menggunakan Maxima untuk menghitung normal untuk kita. Pertama, tiga fungsi untuk koordinat sebagai ekspresi simbolik.

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

```
Z &= sin(x)+2*cos(3*y);
```

Kemudian dua vektor turunan terhadap x dan y.

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

Sekarang yang normal, yang merupakan produk silang dari dua turunan.

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

Kami sekarang mengevaluasi semua ini secara numerik.

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

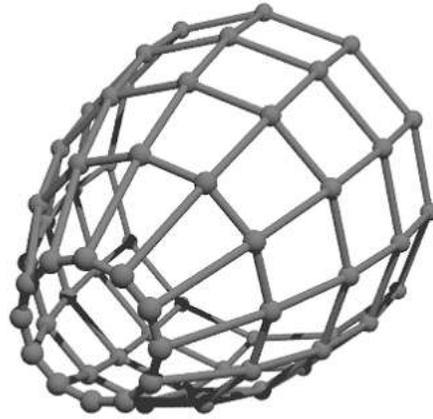
Vektor normal adalah evaluasi dari ekspresi simbolik dn[i] untuk i=1,2,3. Sintaks untuk ini adalah &"ekspresi"(parameter). Ini adalah sebuah alternatif dari metode pada contoh sebelumnya, di mana kita mendefinisikan ekspresi simbolik NX, NY, NZ terlebih dahulu.

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



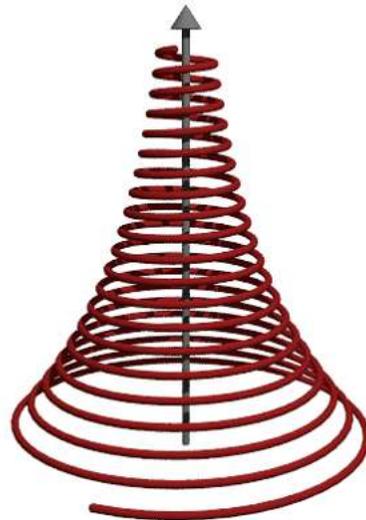
Kami juga dapat menghasilkan kisi-kisi dalam bentuk 3D.

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



Dengan `povgrid()`, kurva dapat dibuat.

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



Objek Povray

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

Kita memulai output dengan povstart().

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

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

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

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

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

```
>c2
```

```
cylinder { <0,0,-1>, <0,0,1>, 1  
  texture { pigment { color rgb <0.941176,0.941176,0.392157> } }  
  finish { ambient 0.2 }  
}
```

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

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

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

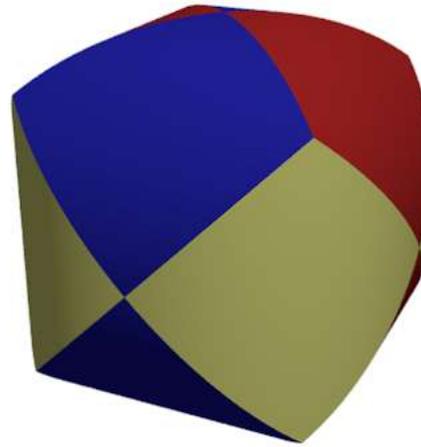
```
texture { pigment { color rgbf <0.101961,0.2,0.301961,0.1> } }  
finish { ambient 0.5 }
```

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

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

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

```
>povend;
```



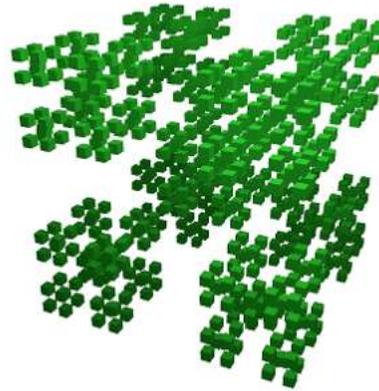
Fungsi-fungsi berikut ini menghasilkan fraktal secara rekursif.

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

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

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

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



Perbedaan memungkinkan pemotongan satu objek dari objek lainnya. Seperti persimpangan, ada bagian dari objek CSG Povray.

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

Untuk demonstrasi ini, kita akan mendefinisikan sebuah objek di Povray, alih-alih menggunakan sebuah string di Euler. Definisi akan langsung dituliskan ke file.

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

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

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

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

Kami menghasilkan kubus kedua, dan memutar serta menskalakannya sedikit.

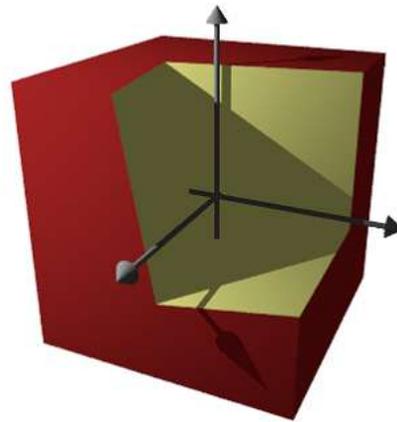
```
>c2=povobject ("mycube", povlook (yellow), translate=[1,1,1], ...  
rotate=xrotate (10°)+yrotate (10°), scale=1.2);
```

Kemudian kita ambil selisih dari kedua objek tersebut.

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

Sekarang tambahkan tiga sumbu.

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



Fungsi Implisit

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

Sintaks untuk fungsi-fungsi tersebut sedikit berbeda. Anda tidak dapat menggunakan output dari ekspresi Maxima atau Euler.

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

```
>povstart (angle=70°,height=50°,zoom=4);
>c=0.1; d=0.1; ...
writeln (povsurface (" (pow (pow (x,2)+pow (y,2)-pow (c,2),2)+pow (pow (z,2)-1,2) ) * (pow (pow (y,2)+pow (z,2)-pow (c,2),2)+pow (pow (x,2)-1,2) ) * (pow (pow (z,2)+pow (x,2)-pow (c,2),2)+pow (pow (y,2)-1,2) ) ) ) );
povend();
```

```
Error : Povray error!
```

```
Error generated by error() command
```

```
povray:
  error("Povray error!");
Try "trace errors" to inspect local variables after errors.
povend:
  povray (file,w,h,aspect,exit);
```

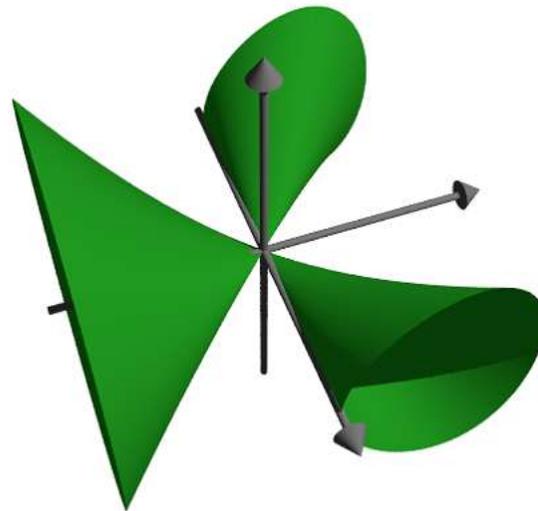
```
>povstart (angle=25°,height=10°);
>writeln (povsurface ("pow (x,2)+pow (y,2)*pow (z,2)-1",povlook (blue),povbox (-2,2,"")));
>povend();
```



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

Membuat permukaan implisit. Perhatikan sintaks yang berbeda dalam ekspresi.

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



Objek Jaring

Pada contoh ini, kami menunjukkan cara membuat objek mesh, dan menggambarnya dengan informasi tambahan.

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

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

Kita tidak dapat menyimpan objek dalam sebuah string seperti sebelumnya, karena ukurannya terlalu besar. Jadi kita mendefinisikan objek dalam file Povray menggunakan #declare. Fungsi povtriangle() melakukan hal ini secara otomatis. Fungsi ini dapat menerima vektor normal seperti halnya pov3d().

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

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

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

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

Tuliskan permukaan dikurangi kedua cakram.

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

Tuliskan kedua perpotongan tersebut.

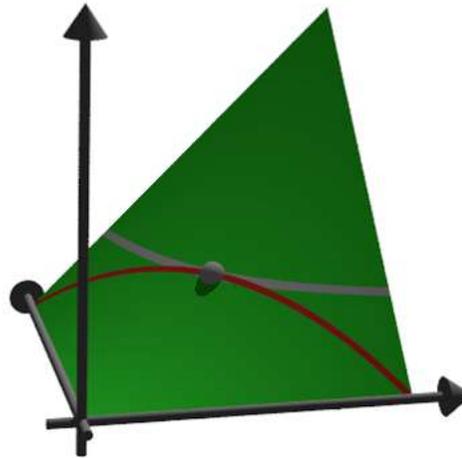
```
>writeln(povintersection([mesh,c1],povlook(red))); ...
writeln(povintersection([mesh,l1],povlook(gray)));
```

Tuliskan satu titik secara maksimal.

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

Tambahkan sumbu dan selesaikan.

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



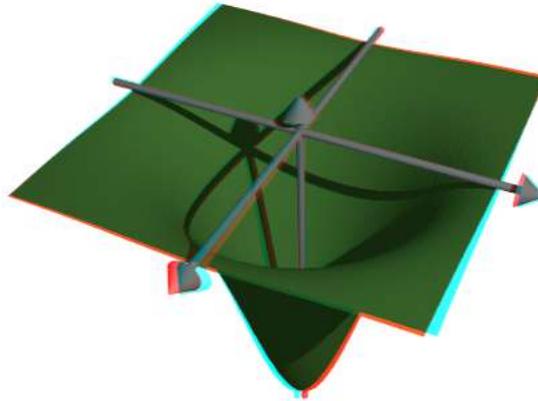
Anaglyph dalam Povray

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 membutuhkan kacamata merah/cyan untuk melihat contoh berikut dengan benar.

Fungsi `pov3d()` memiliki tombol sederhana untuk menghasilkan anaglyph.

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

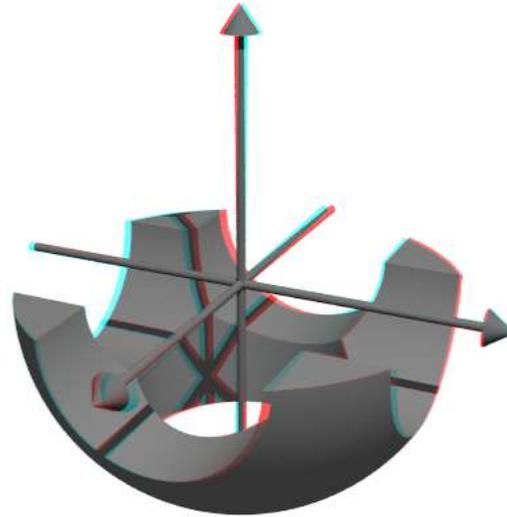


Jika Anda membuat scene dengan objek, Anda harus menempatkan pembuatan scene ke dalam suatu fungsi, dan menjalankannya dua kali dengan nilai yang berbeda untuk parameter anaglyph.

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

Fungsi povanaglyph() melakukan semua ini. Parameter-parameternya seperti pada povstart() dan povend() yang digabungkan.

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



Mendefinisikan Objek sendiri

Antarmuka povray Euler berisi banyak objek. Namun Anda tidak dibatasi pada objek-objek tersebut. Anda dapat membuat objek sendiri, yang menggabungkan objek-objek lain, atau objek yang benar-benar baru.

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

```
>function povdonat (r1,r2,look="") ...  
    return "torus {"+r1+" "+r2+look+"}";  
endfunction
```

Inilah torus pertama kami.

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

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

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

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

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

Sekarang, kita tempatkan semua benda ini ke dalam suatu pemandangan. Untuk tampilannya, kami menggunakan Phong Shading.

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

```
writeln (povobject (t2,povlook (green,phong=1))); ...
```

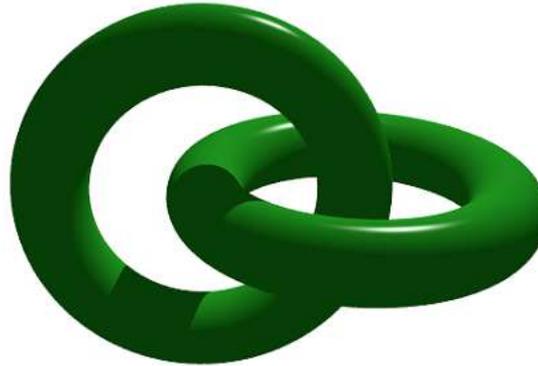
```
>povend();
```

memanggil program Povray. Namun, jika terjadi kesalahan, program ini tidak menampilkan kesalahan. Oleh karena itu, Anda harus menggunakan

```
>povend (<exit);
```

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

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



Berikut adalah contoh yang lebih rumit. Kami menyelesaikan

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

dan menunjukkan titik-titik yang layak dan optimal dalam plot 3D.

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

Pertama, mari kita periksa, apakah contoh ini memiliki solusi atau tidak.

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

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

Ya, benar.

Selanjutnya kita mendefinisikan dua objek. Yang pertama adalah pesawat

$$a \cdot x \leq b$$

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

Kemudian kita tentukan perpotongan semua setengah ruang dan kubus.

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

Sekarang, kita bisa merencanakan adegan tersebut.

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

Berikut ini adalah lingkaran di sekeliling optimal.

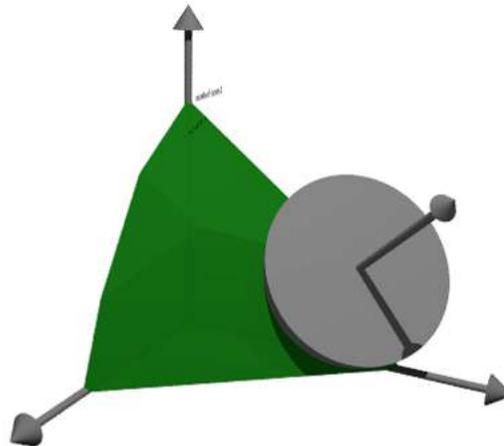
```
>writeln (povintersection ([povsphere (x,0.5),povplane (c,c.x')], ...
  povlook (red,0.9)));
```

Dan kesalahan pada arah yang optimal.

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

Kami menambahkan teks ke layar. Teks hanyalah sebuah objek 3D. Kita perlu menempatkan dan memutarnya sesuai dengan pandangan kita.

```
>writeln (povtext ("Linear Problem", [0,0.2,1.3],size=0.05,rotate=5°)); ...
povend();
```



Contoh Lainnya

Anda dapat menemukan beberapa contoh lain untuk Povray di Euler dalam file-file berikut.

[Examples/Dandelin Spheres](#)
[Examples/Donat Math](#)

DEFINISI FUNGSI

Sebuah fungsi f adalah suatu aturan padanan yang menghubungkan tiap obyek x dalam satu himpunan, yang disebut daerah asal, dengan sebuah nilai unik $f(x)$ dari himpunan kedua. Himpunan nilai yang diperoleh secara demikian disebut daerah hasil.

DEFINISI FUNGSI DUA VARIABEL

Sebuah fungsi bernilai-riil dari dua variabel riil; yakni, fungsi f yang memadankan setiap pasangan terurut (x,y) pada suatu himpunan D dari bidang dengan satu dan hanya satu bilangan real yang ditulis sebagai $z = f(x,y)$.

Himpunan D disebut daerah asal fungsi. Sedangkan daerah nilai fungsi adalah himpunan nilai-nilainya. Misalnya $z = f(x,y)$, merupakan fungsi dua variabel dengan x,y disebut sebagai variabel bebas(independent variable) dan z variabel tak bebas (dependent variable). Sebagai contoh

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

Perhatikan grafik fungsi permukaan bola dengan persamaan

$$x^2 + y^2 + z^2 = 1$$

yang berpusat di titik asal $O(0,0,0)$ dan berjari-jari 1. Dalam permukaan tersebut titik-titik $(x,y)=(0,0)$ berpadanan dengan dua nilai z , yakni -1 dan 1. Artinya oleh permukaan tersebut terdapat pemetaan dari $(0,0)$ ke dua nilai berbeda, maka pemetaan seperti itu bukan merupakan suatu fungsi.

PERMUKAAN DALAM R^3 (RUANG DIMENSI 3)

Terdapat tiga sumbu koordinat yang saling tegak lurus yaitu: sumbu x , sumbu y , sumbu z . Ruang R^3 oleh ketiga sumbu x,y,z tersekut dalam delapan oktan. Kumpulan dari titik-titik di R^3 dapat berupa kurva ataupun permukaan. Dalam R^3 terdapat permukaan linear(berupa bidang datar) dan kuadratik(berupa bidang lengkung). Permukaan linear tidak mungkin dapat dibuat keseluruhan bidangnya, cukup digambar wakil bidang yang dapat berupa segitiga, segiempat, dll. Permukaan kuadratik dapat berupa permukaan bola, elipsoida, paraboloida, tabung elips, tabung lingkaran, atau tabung parabola. Beberapa persamaan umum dari permukaan kuadratik.

- Bola: $x^2+y^2+z^2=a^2, a>0$

$$x^2 + y^2 + z^2 = a^2, a > 0$$

- Elipsoida: $\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1, a,b,c>0$

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1, a, b, c > 0$$

- Hiperboloida Berdaun Satu: $\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = 1, a,b,c>0$

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = 1, a, b, c > 0$$

- Hiperboloida Berdaun Dua: $\frac{x^2}{a^2} - \frac{y^2}{b^2} - \frac{z^2}{c^2} = 1, a,b,c>0$

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} - \frac{z^2}{c^2} = 1, a, b, c > 0$$

- Paraboloida Eliptik: $z = \frac{x^2}{a^2} + \frac{y^2}{b^2}, a,b>0$

$$z = \frac{x^2}{a^2} + \frac{y^2}{b^2}, a, b > 0$$

- Paraboloida Hiperbolik: $z = \frac{y^2}{b^2} - \frac{x^2}{a^2}, a,b>0$

$$z = \frac{y^2}{b^2} - \frac{x^2}{a^2}, a, b > 0$$

- Kerucut Eliptik: $\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = 0$

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} - \frac{z^2}{c^2} = 0$$

Dalam menggambar sketsa permukaan, dapat dibuat langkah bantuan dengan menggambar perpotongan permukaan tersebut dengan tiga bidang utama, yaitu XOY, XOZ, dan YOZ.
Jika permukaan sangat rumit, dapat digunakan komputer untuk menggambar grafiknya,

GRAFIK FUNGSI DUA VARIABEL

Grafik fungsi dua variabel adalah himpunan

$$(x, y, z) | z = f(x, y), (x, y) \in D$$

yang merupakan himpunan titik dalam ruang atau R³. Himpunan ini pada umumnya membentuk permukaan di ruang. Ketika kita menyebut grafik dari fungsi f dengan dua variabel, yang dimaksud adalah grafik dari persamaan $z = f(x, y)$.

Beberapa fungsi matematika yang 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.

Contoh:

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

2. Fungsi Kuadratik

Bentuk umum

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

dimana a, b, c, d, e, dan f adalah konstanta. Grafik fungsi kuadrat ini adalah sebuah permukaan yang dapat memiliki berbagai bentuk tergantung pada nilai-nilai konstantanya.

Contoh:

$$f(x, y) = 2x^2 - 4y^2 + 3xy$$

3. Fungsi Trigonometri

Fungsi trigonometri dua variabel adalah fungsi matematika yang melibatkan operasi trigonometri (seperti sin, cos, tan) pada kedua variabel x dan y. Contoh:

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

4. Fungsi Aljabar

Fungsi aljabar adalah fungsi yang bisa didefinisikan sebagai akar dari sebuah persamaan aljabar. Fungsi aljabar merupakan ekspresi aljabar menggunakan sejumlah suku terbatas, yang melibatkan operasi aljabar seperti penambahan, pengurangan, perkalian, pembagian, dan peningkatan menjadi pangkat pecahan. Contoh dari fungsi tersebut adalah:

$$f(x, y) = 1/xy$$

$$f(x, y) = \sqrt{xy}$$

$$f(x, y) = \frac{\sqrt{1+x^3}}{3^{3/7} - \sqrt[7]{y^{1/3}}}$$

5. Fungsi Eksponensial

Fungsi eksponensial dua variabel bisa dinyatakan

$$f(x, y) = a \cdot b^{xy}$$

dimana a dan b adalah konstanta, x dan y adalah variabel. Fungsi ini menggambarkan pertumbuhan eksponensial yang bergantung pada nilai x dan y.

Contoh:

$$f(x, y) = 2 \cdot 3^{xy}$$

1. Menggambar Grafik Fungsi Dua Variabel dalam Bentuk Ekspresi

Langsung

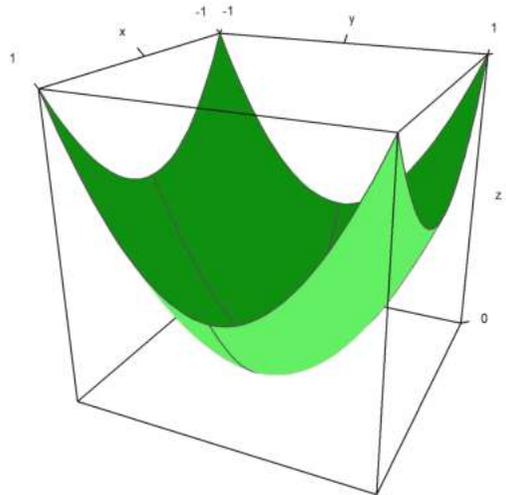
Grafik fungsi dua variabel dalam bentuk ekspresi langsung adalah representasi visual dari hubungan matematis antara dua variabel independen yang dinyatakan dalam bentuk persamaan atau ekspresi matematis.

Contoh Soal 1(Fungsi Kuadratik)

Gambarlah grafik dari fungsi berikut.

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

```
>plot3d("2*x^2+3*y^2",n=40,grid=2):
```



Gambar di atas menampilkan grafik fungsi dengan $n=40$ dan $grid=2$.

- n = jumlah interval kisi-kisi, jumlah n default=40

- $grid$ = jumlah garis kisi di setiap arah, jumlah $grid$ default=10

Penjelasan:

misalkan

$$z = 2x^2 + 3y^2$$

$$z = \frac{x^2}{\frac{1}{2}} + \frac{y^2}{\frac{1}{3}}$$

(yang dikenal sebagai persamaan sebuah paraboloida eliptik) dan perhatikan bahwa

$$z \geq 0$$

cari jejak pada bidang koordinat
-bidang XOY(z=0):

$$\frac{x^2}{\frac{1}{2}} + \frac{y^2}{\frac{1}{3}} = 0$$

jika z=0 maka x^2 dan y^2 juga harus 0, maka diperoleh titik (0,0,0)

-bidang YOZ(x=0)

$$\frac{y^2}{\frac{1}{3}} = z$$

$$y^2 = \frac{1}{3}z$$

(berupa parabola searah sumbu z dan titik puncaknya (0,0))

-bidang XOZ(y=0)

$$\frac{x^2}{\frac{1}{2}} = z$$

$$x^2 = \frac{1}{2}z$$

(berupa parabola searah sumbu z dan titik puncaknya (0,0))

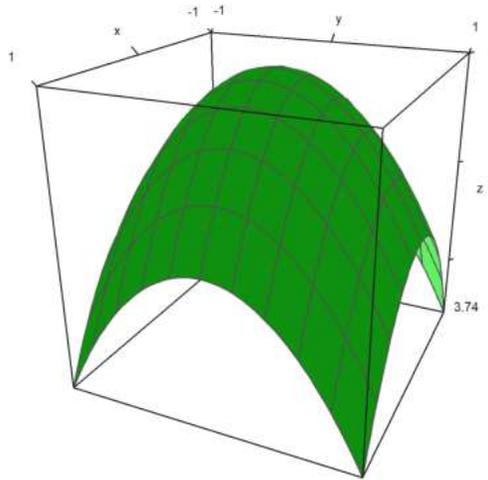
Contoh Soal 2(Fungsi Aljabar)

Gambarlah grafik dari fungsi berikut.

$$f(x, y) = \sqrt{16 - (x^2 + y^2)}$$

```
>plot3d("(16-x^2-y^2)^(1/2)",>user, ...  
title= "Turn with the vector keys (press return to finish)":
```

Turn with the vector keys (press return to finish)



Gambar di atas menampilkan grafik fungsi dengan menggunakan >user.

Untuk menggunakan >user, kita dapat menekan tombol:

- kiri,kanan,atas,bawah:putar sudut pandang
- + -:memperbesar atau memperkecil
- a:menghasilkan anaglyph
- l:sakelar untuk memutar sumbu cahaya
- spasi:setel ulang ke default
- enter: mengakhiri interaksi

Penjelasan :
misalkan

$$z = \sqrt{16 - (x^2 + y^2)}$$

dan perhatikan bahwa

$$z \geq 0$$

Jika kedua ruas dikuadratkan dan sederhanakan, akan kita peroleh persamaan

$$x^2 + y^2 + z^2 = 16$$

yang kita kenal sebagai persamaan sebuah bola.

cari jejak pada bidang koordinat
-bidang XOY(z=0):

$$x^2 + y^2 = 16$$

(berupa lingkaran dengan pusat(0,0) dan jari-jari 4)
-bidang YOZ(x=0)

$$y^2 + z^2 = 16$$

(berupa lingkaran dengan pusat(0,0) dan jari-jari 4)
-bidang XOZ(y=0)

$$x^2 + z^2 = 16$$

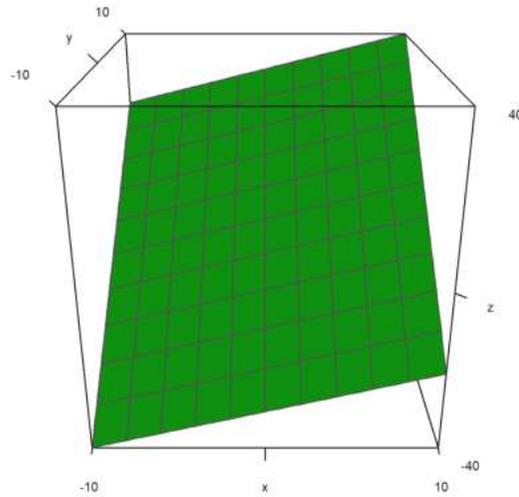
(berupa lingkaran dengan pusat(0,0) dan jari-jari 4)

Contoh Soal 3 (Fungsi Linear)

Gambarlah grafik dari fungsi berikut.

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

```
>plot3d("x+3*y", angle=0°, a=-10, b=10, c=-10, d=10, fscale=10) :
```



Gambar di atas menampilkan grafik fungsi dengan angle=0 derajat, a=-10, b=10, c=-10, d=10, fscale=10.

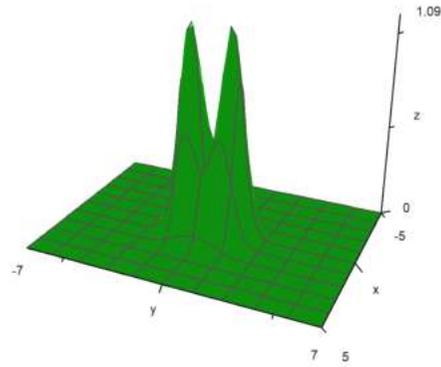
- angle: sudut pandang
- a, b: rentang x
- c, d: rentang y
- fscale: skala ke nilai fungsi (defaultnya adalah <fscale)

Contoh Soal 4(Fungsi Eksponensial)

Gambarlah grafik dari fungsi berikut.

$$f(x, y) = (x^2 + 3y^2)e^{-x^2 - y^2}$$

```
>plot3d("(x^2+3*y^2)*E^(-x^2-y^2)", scale={1, 2}, xmin=-5, xmax=5, ymin=-7, ymax=7, frame=3) :
```



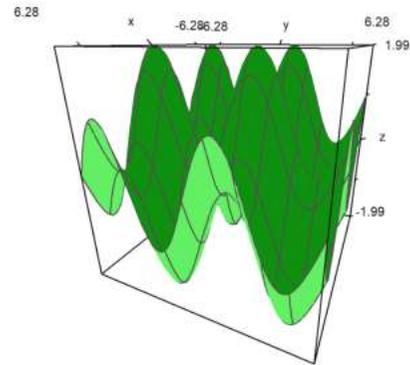
Gambar di atas menampilkan grafik fungsi dengan `scale=[1,2]`, `xmin=-5`, `xmax=5`, `ymin=-7`, `ymax=7`, `frame=3`.
- `scale`: angka atau vektor 1x2 untuk menskalakan ke arah x dan y
- `xmin,xmax`: rentang x
- `ymin,ymax`: rentang y
- `frame`: jenis bingkai (default 1)

Contoh Soal 5(Fungsi Trigonometri)

Gambarlah grafik dari fungsi berikut.

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

```
>plot3d("sin(x)+sin(y)", r=2*pi, distance=3, zoom=1, center=[0.1,0,0], height=20°):
```



Gambar di atas menampilkan grafik fungsi dengan $r=2\pi$, $distance=3$, $zoom=1$, $center=[0.1,0,0]$, $height=20$ derajat.

- r: dapat digunakan sebagai ganti xmin, xmax, ymin, ymax; r dapat berupa vektor [rx, ry] atau [rx, ry, rz]
- distance: jarak pandang plot
- zoom: nilai zoom
- center: memindahkan bagian tengah plot
- height: ketinggian tampilan dalam radian

Nilai default dari distance, zoom, angle, height dapat diperiksa atau diubah dengan fungsi view. Fungsi ini mengembalikan parameter sesuai urutan di atas.

```
>view
```

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

2. Menggambar Grafik Fungsi Dua Variabel yang Rumusnya Disimpan

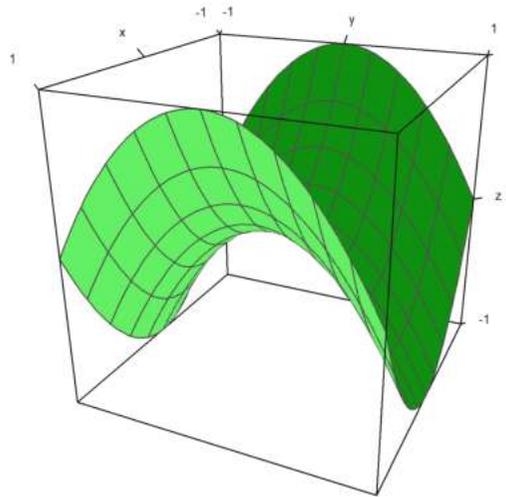
dalam Variabel Ekspresi

Untuk menyimpan sebuah fungsi, dapat dilakukan menggunakan perintah function. Kemudian untuk memanggil atau membuat grafiknya tinggal memanggil nama fungsi tersebut. Contohnya :

```
>function b(x,y):=x^2-y^2;
```

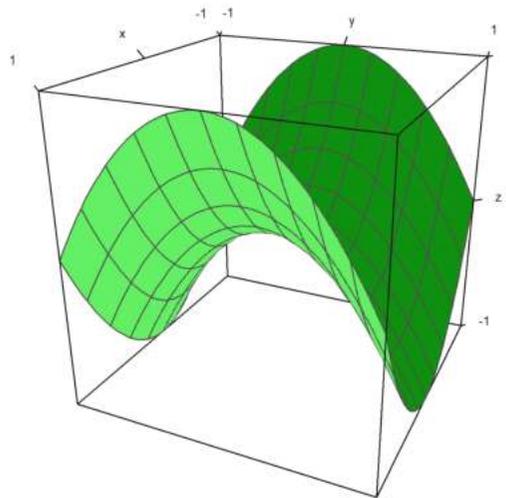
selanjutnya kita akan membuat grafik dari fungsi tersebut

```
>plot3d("b(x,y) "):
```



selain cara di atas, kita juga bisa membuat grafik dari fungsi tersebut dengan

```
>plot3d("b") :
```

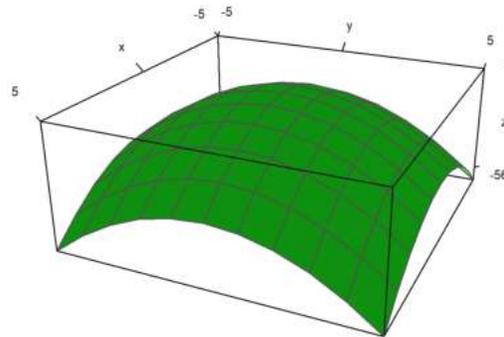


Contoh Soal 1(Fungsi Kuadratik)

Gambarlah grafik dari fungsi tersebut.

$$f(x, y) = -6 - x^2 - y^2$$

```
>function p(x,y) :=-6-x^2-y^2;  
>plot3d("p(x,y)",r=5, ...  
fscale=2,n=10, zoom=2.7) :
```



Penjelasan :
misalkan

$$z = -6 - x^2 - y^2$$

Cari domainnya

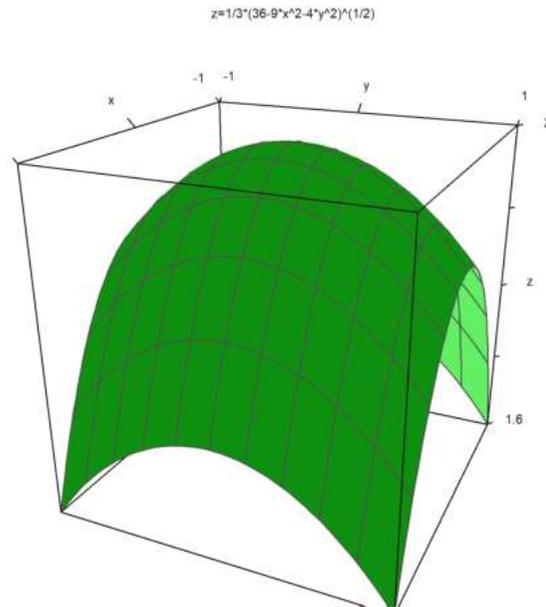
Cari daerah hasilnya

Contoh Soal 2(Fungsi Aljabar)

Gambarlah grafik dari fungsi berikut.

$$f(x, y) = \frac{1}{3} \sqrt{36 - 9x^2 - 4y^2}$$

```
>function z(x,y) :=1/3*(36-9*x^2-4*y^2)^(1/2);  
>plot3d("z(x,y)",title="z=1/3*(36-9*x^2-4*y^2)^(1/2)", zoom=3) :
```



Penjelasan :
misalkan

$$z = \frac{1}{3} \sqrt{36 - 9x^2 - 4y^2}$$

dan perhatikan bahwa

$$z \geq 0$$

Jika kedua ruas di kuadratkan dan sederhanakan, akan diperoleh:

$$9x^2 + 4y^2 + 9z^2 = 36$$

yang dikenal sebagai persamaan sebuah elipsoida.

cari jejak pada bidang koordinat
-bidang XOY($z=0$):

$$9x^2 + 4y^2 = 36$$

$$\frac{x^2}{4} + \frac{y^2}{9} = 1$$

(berupa elips dengan pusat(0,0), titik puncak : (0,-2),(0,2),(0,3),(0,-3))

-bidang YOZ($x=0$)

$$4y^2 + 9z^2 = 36$$

$$\frac{y^2}{9} + \frac{z^2}{4} = 1$$

(berupa elips dengan pusat (0,0), titik puncak : (0,-3),(0,3),(0,-2),(0,2))

-bidang XOZ($y=0$)

$$9x^2 + 9z^2 = 36$$

$$x^2 + z^2 = 4$$

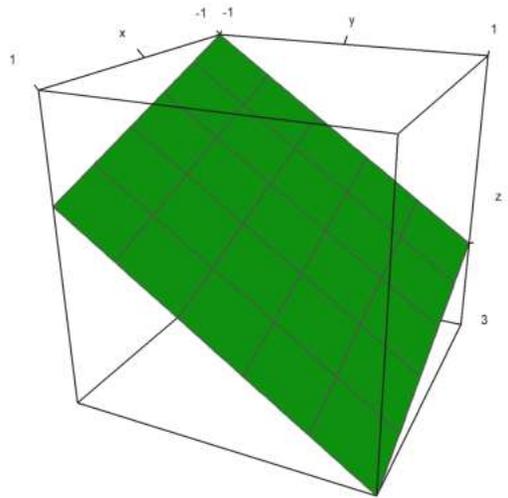
(berupa lingkaran dengan pusat(0,0), jari-jari=2)

Contoh Soal 3(Fungsi Linear)

a) Gambarlah grafik dari fungsi tersebut.

$$f(x,y) = 6 - x - 2y$$

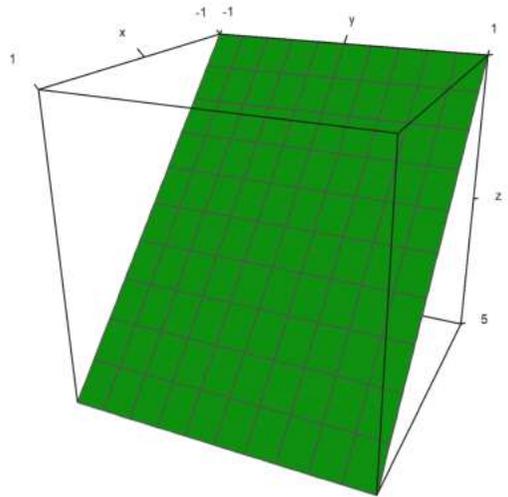
```
>function q(x,y):=6-x-2*y;  
>plot3d("q(x,y)",grid=5):
```



b) Gambarlah grafik dari fungsi tersebut.

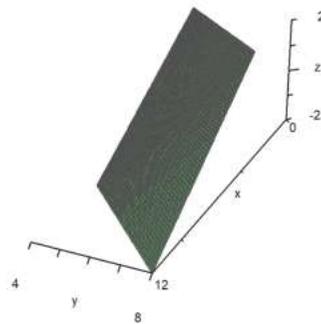
$$f(x,y) = 6 - x$$

```
>function j(x,y):=6-x;  
>plot3d("j"):
```



Berikut adalah plot dari tiga fungsi.

```
>plot3d("q", "j", "y", r=2, zoom=3, frame=3) :
```

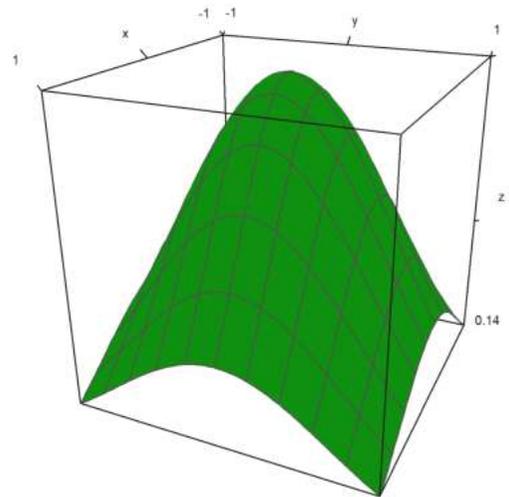


Contoh Soal 4(Fungsi Eksponensial)

a) Gambarkan grafik dari fungsi tersebut.

$$f(x, y) = e^{-(x^2+y^2)}$$

```
>function n(x, y) := E^(-(x^2+y^2));  
>plot3d("n", >fscale, >scale):
```

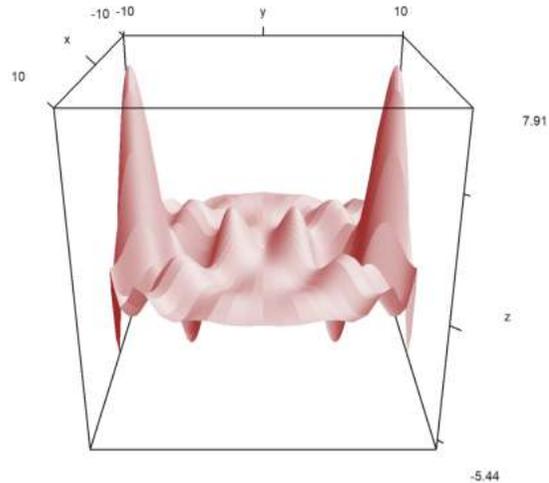


Contoh Soal 5(Fungsi Trigonometri)

Gambarkan grafik dari fungsi tersebut.

$$f(x, y) = \frac{\sin x \sin y}{xy}$$

```
>function m(x, y) := (sin(x) * sin(y)) / x * y;  
>plot3d("m", r=10, angle=90°, fscale=-1, >user, >polar, color=red, >hue):
```



Dalam membuat grafik di atas terdapat >polar.

- >polar: menghasilkan plot polar
- hue: mengaktifkan bayangan cahaya
- color: mengatur warna pada grafik

Subtopik 3 ** Menggambar Grafik Fungsi Numerik Dua Variabel

Untuk membuat plot sebuah fungsi di EMT, cara termudahnya adalah dengan menggunakan ekspresi numerik atau simbolik, dalam variabel x. Pada subtopik kali ini, akan dibahas mengenai 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 sebagainya.

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

Penulisan Sintaks:

1) definisikan fungsi numerik

function f(x,y):= ax+by dengan a dan b adalah suatu konstanta dan fungsi tidak selalu direpresentasikan dengan f tetapi bisa dengan huruf apapun. Contoh:g(x,y)

2) sintaks plot3d

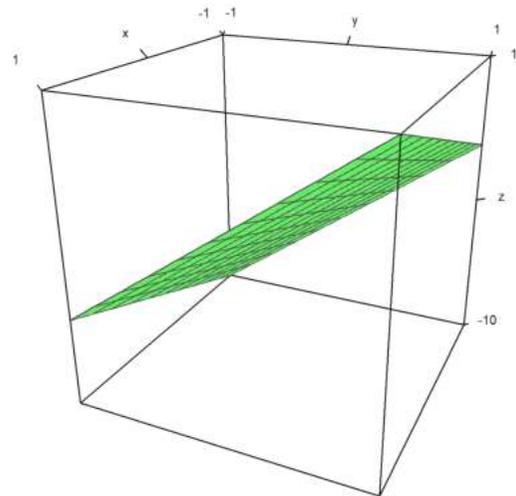
plot3d("f"):

Contoh Soal

1. Grafik fungsi linear dua variabel

$$f(x, y) = 3x + 7y$$

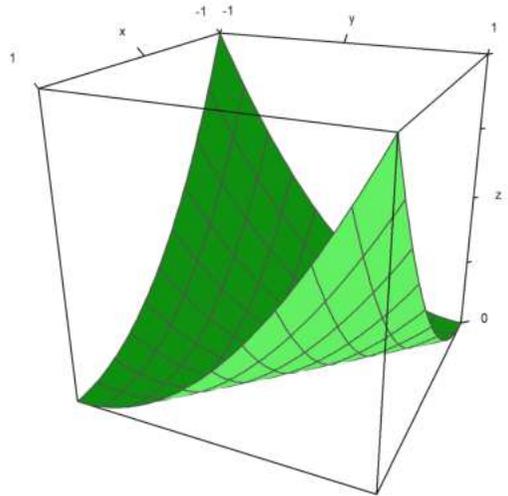
```
>function f(x,y) := 3*x+7*y  
>plot3d("f") :
```



2. Grafik fungsi kuadrat dua variabel

$$f(x, y) = x^2 + 2xy + y^2$$

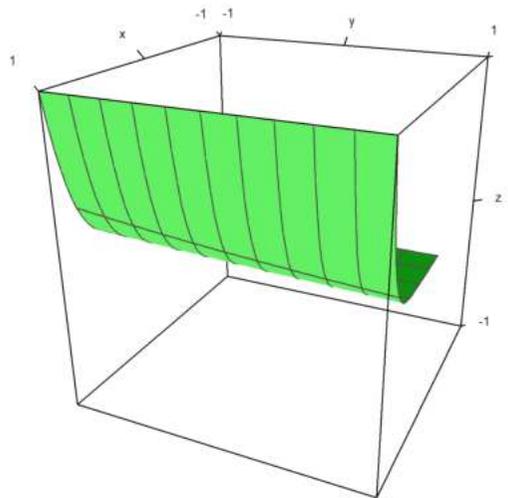
```
>function f(x,y) := x^2+2x*y+y^2  
>plot3d("f") :
```



3. Grafik fungsi eksponen dua variabel

$$f(x, y) = x^{2y+8}$$

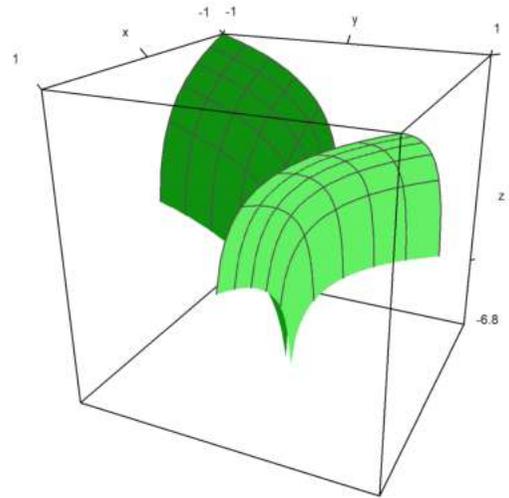
```
>function f(x,y):=x^(2*y+8)
>plot3d("f"):
```



4. Grafik fungsi logaritma dua variabel

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

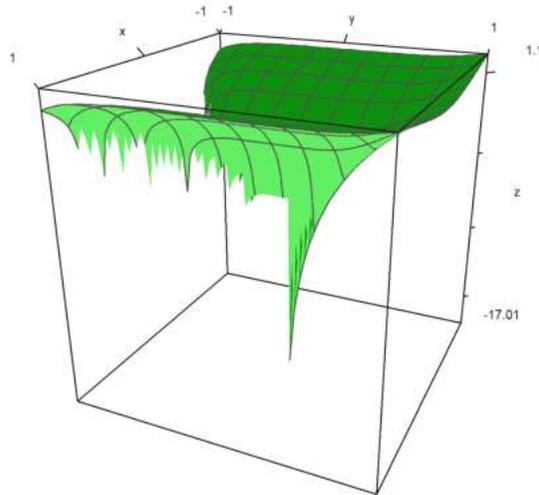
```
>function f(x,y) := log(x*y)  
>plot3d("f") :
```



5. Grafik fungsi trigonometri dua variabel

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

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



Subtopik 4 ** Menggambar Grafik Fungsi Simbolik Dua Variabel

Pada subtopik kali ini, akan dibahas mengenai grafik fungsi dua variabel yang fungsinya didefinisikan sebagai fungsi simbolik.

Fungsi simbolik memberikan ekspresi simbolik ke sebuah variabel dalam Maxima dan Euler, setelah ekspresi tersebut dievaluasi oleh Maxima. Ekspresi $\&=&$ memberikan sebuah ekspresi hanya dalam bentuk simbolik. Ekspresi dievaluasi sebelum ekspresi tersebut didefinisikan. Hal ini berguna jika ekspresi berisi fungsi-fungsi yang tidak dapat dievaluasi secara numerik. Misalnya, jika kita memiliki fungsi

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

Di dalam Euler, kita dapat menggambar plot dengan menggunakan perintah `plot3d()`. Untuk fungsi simbolik kita definisikan terlebih dahulu menggunakan tanda $\&=&$. Lalu, untuk menampilkan gambar hasil plot di layar notebook, perintah `plot3d()` dapat diakhiri dengan titik dua (`:`).

Perbedaan utama antara fungsi numerik dan fungsi simbolik adalah bahwa fungsi numerik memberikan hasil numerik secara langsung (menghasilkan angka), sementara fungsi simbolik memungkinkan kita untuk bekerja dengan simbol matematika sebelum menghitung nilai numeriknya. Pilihan antara keduanya tergantung pada kebutuhan analisis matematika yang kita lakukan.

Tujuan menggambar grafik fungsi dua variabel adalah untuk memahami pola, sifat, dan hubungan antara dua variabel tersebut secara visual, yang dapat membantu dalam analisis dan pemahaman masalah matematika atau ilmu pengetahuan yang melibatkan fungsi ini.

Penulisan sintaks:

1) definisikan fungsi simbolik

`function f(x,y):= ax+by` dengan a dan b adalah suatu konstanta dan fungsi tidak selalu direpresentasikan dengan f tetapi bisa dengan huruf apapun. Contoh: `g(x,y)`

2) sintaks `plot3d`

`plot3d("f"):`

3) menentukan rentang variabelnya

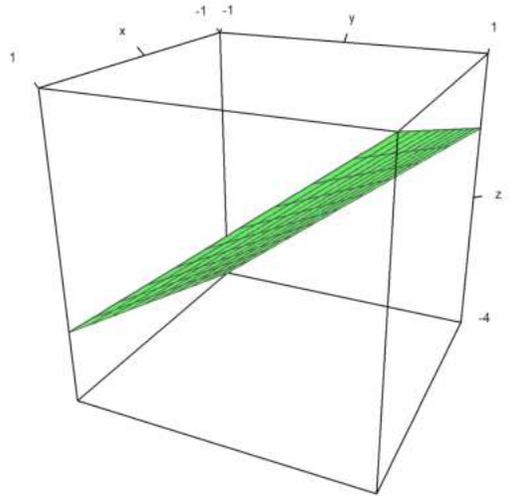
`plot3d("f", m,n,o,p*pi):` dengan m,n,o, dan p adalah suatu konstanta

Contoh Soal

1. Grafik fungsi linear dua variabel

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

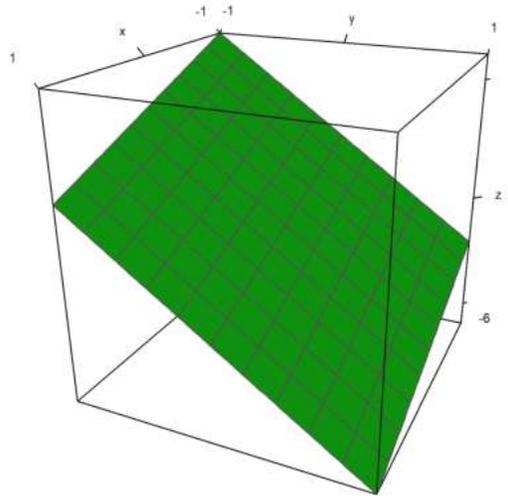
```
>function g(x,y) &= x+3*y;  
>plot3d("g"):
```



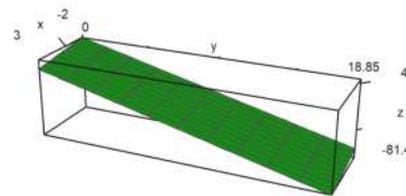
Berikut adalah contoh lain dari grafik fungsi linear dua variabel dengan rentang variabel yang telah ditentukan.

$$g(x,y) = -2x - 4y$$

```
>function g(x,y) &= -2*x-4*y;  
>plot3d("g"):
```



```
>plot3d("g",-2,3,0,6*pi):
```

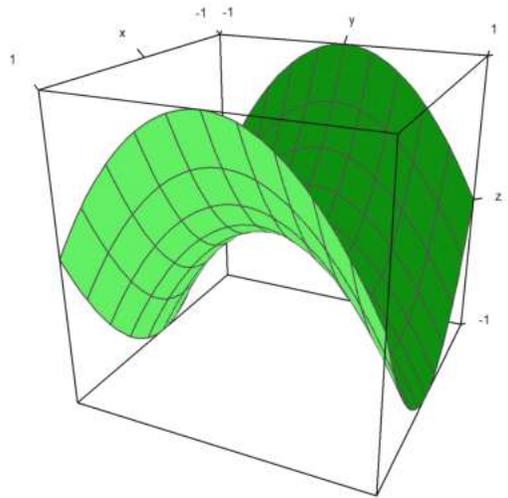


Dari contoh grafik di atas dapat dilihat bahwa secara default gambar plot menggunakan sumbu x dengan rentang nilai dari -2 sampai dengan 2.

2. Grafik fungsi kuadrat dua variabel

$$g(x,y) = x^2 - y^2$$

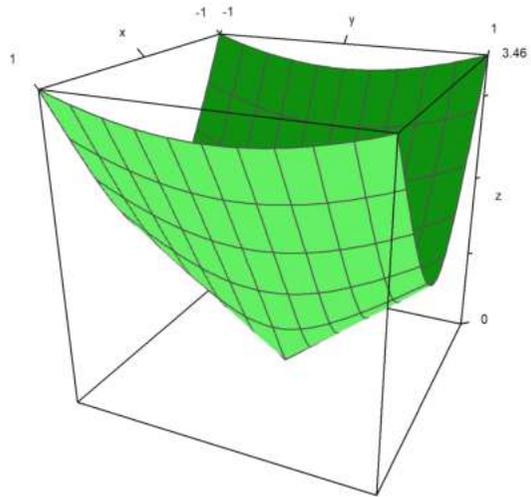
```
>function g(x,y) &= x^2-y^2;  
>plot3d("g"):
```



3. Grafik fungsi akar kuadrat dua variabel

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

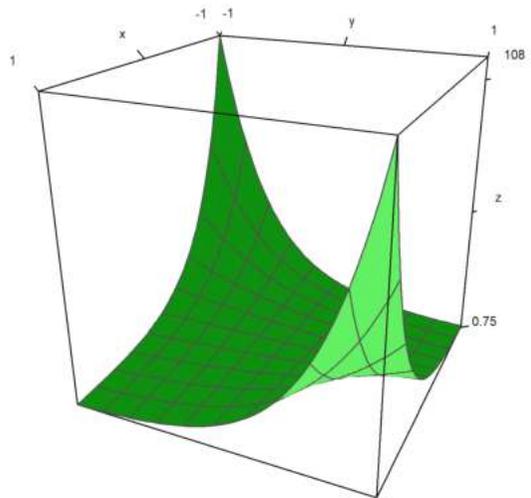
```
>function g(x,y) &= sqrt(10*x^2+2*y^2);  
>plot3d("g"):
```



4. Grafik fungsi eksponen dua variabel

$$g(x, y) = 9.12^{xy}$$

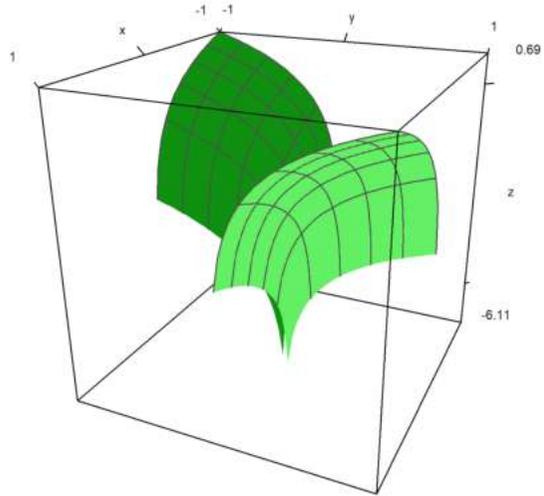
```
>function g(x,y) &=9*12^(x*y);
>plot3d("g"):
```



5. Grafik fungsi logaritma dua variabel

$$g(x, y) = \log(x \cdot 2y), \text{Basis10}$$

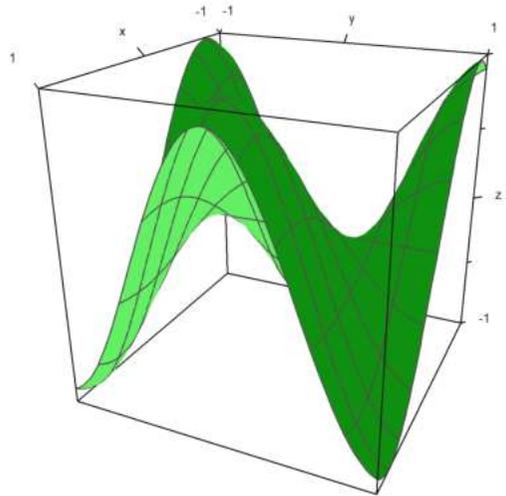
```
>function g(x,y)&=log(x*2*y);  
>plot3d("g"):
```



6. Grafik fungsi trigonometri dua variabel

$$g(x, y) = \sin(2x) \cdot \cos(3y)$$

```
>function g(x,y)&=sin(2*x)*cos(3*y);  
>plot3d("g"):
```



Subtopik 5 ** Menggambar Data x , y , z pada ruang Tiga Dimensi

(3D)

Definisi

Menggambar data pada ruang tiga dimensi (3D) adalah proses visualisasi data yang mengubah informasi dalam tiga dimensi, yaitu panjang, lebar, dan tinggi, menjadi representasi visual yang dapat dipahami dan dianalisis.

Tujuan:

Tujuan dari menggambar data 3D adalah untuk membantu pemahaman dan interpretasi data yang lebih baik, terutama ketika data tersebut memiliki komponen yang tidak dapat direpresentasikan dengan baik dalam dua dimensi.

Sama seperti plot2d, plot3d menerima data. Untuk objek 3D, Kita 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) melalui sebuah kotak persegi. Hasilnya, Anda dapat memplot gambar persegi panjang di ruang angkasa.

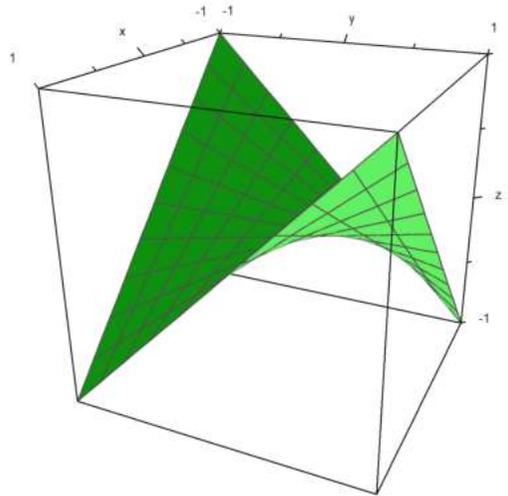
Kita dapat menggunakan bahasa matriks Euler untuk menghasilkan koordinat secara efektif.

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

Contoh 1

grafik fungsi

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



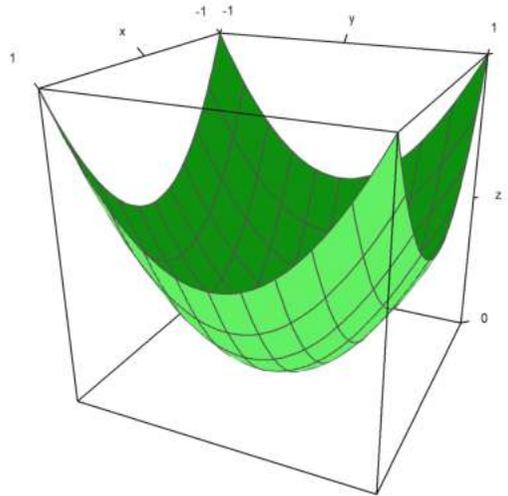
Penjelasan sintaks dari plot
plot 3d = membawa euler untuk mengetahui perintah apa yang harus dilakukan
(" ... ") = tempat untuk memasukkan perintah yang kita inginkan

Contoh 2

Membentuk plot dengan fungsi :

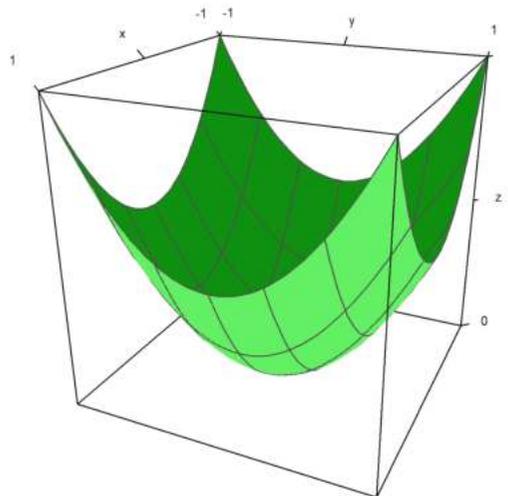
$$x^2 + y^2$$

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



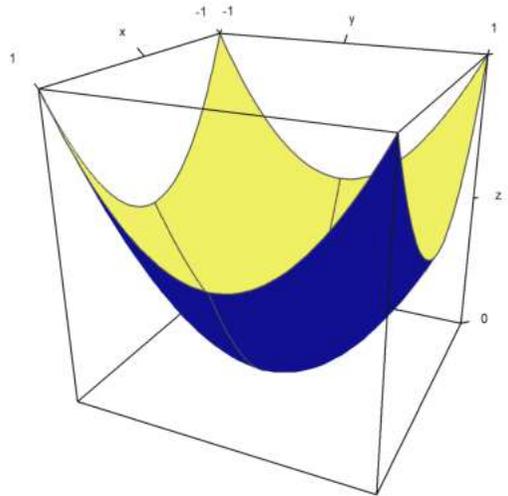
Selanjutnya kita akan menggambar garis pada plot menggunakan grid

```
>plot3d("x^2+y^2",grid=7):
```



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= 2,fillcolor=[blue,yellow]):
```



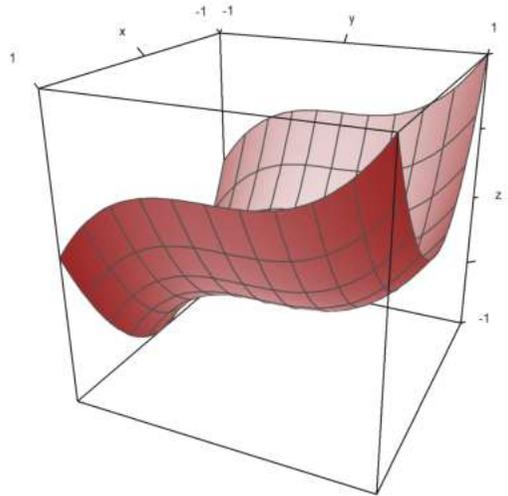
Contoh 3

Membuat plot 3d pada fungsi

$$2x^2 + y^3$$

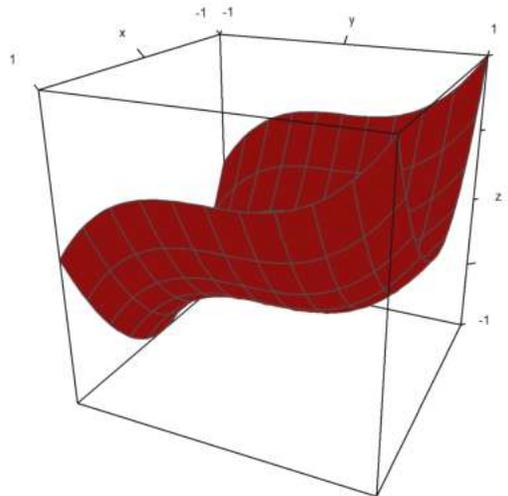
kita bisa menggunakan perintah seperti di bawah ini

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



Jika kita mau menebalkan warna pada gambar diatas maka dapat menggunakan perintah

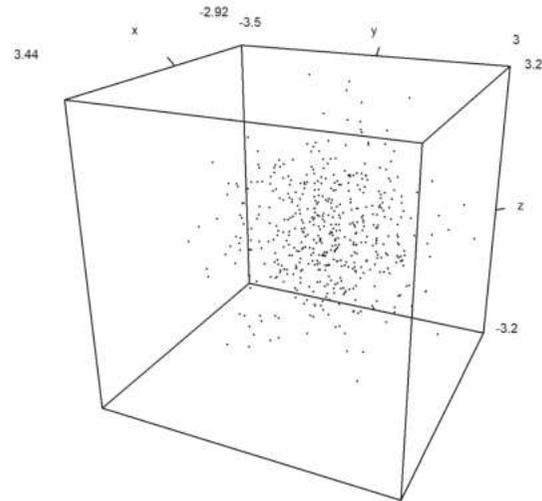
```
>plot3d("2x^2+y^3",grid=10,fillcolor=[red,red], >user);  
>insimg()
```



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

Gayanya sama seperti di plot2d dengan points=true;

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



Contoh 5

Dengan lebih banyak usaha, kami dapat menghasilkan banyak permukaan.

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

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

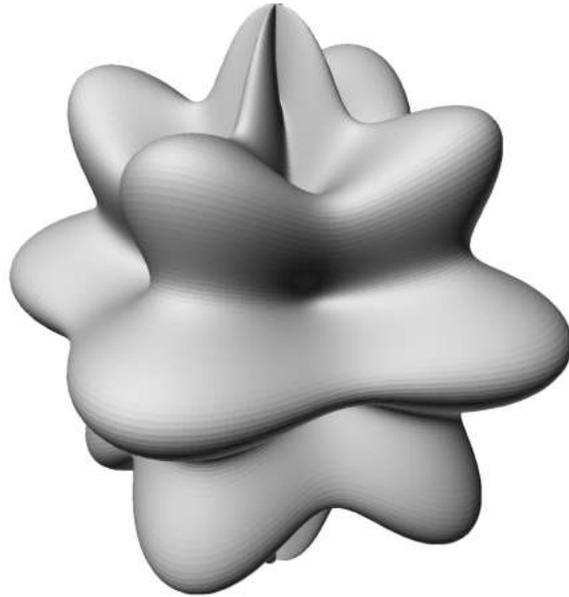
dengan

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

Kami mendistorsi ini dengan sebuah faktor

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

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



Membuat Gambar Grafik Tiga Dimensi (3D)

yang Bersifat Interaktif dan animasi grafik 3D

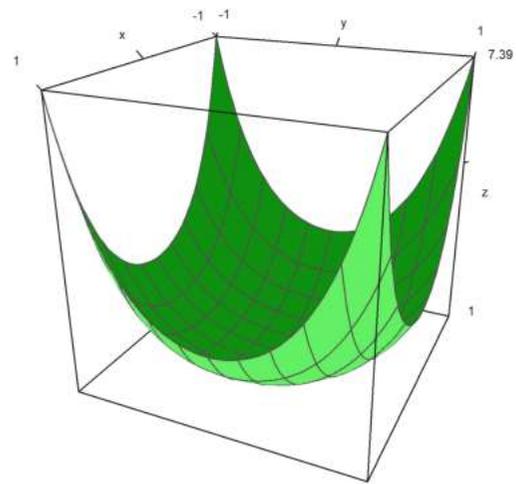
Membuat gambar grafik tiga dimensi (3D) yang bersifat interaktif adalah proses menciptakan visualisasi tiga dimensi yang memungkinkan pengguna berinteraksi dengan objek-objek 3D. Interaktivitas dalam gambar 3D memungkinkan pengguna untuk melakukan tindakan seperti mengubah sudut pandang, memindahkan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

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

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

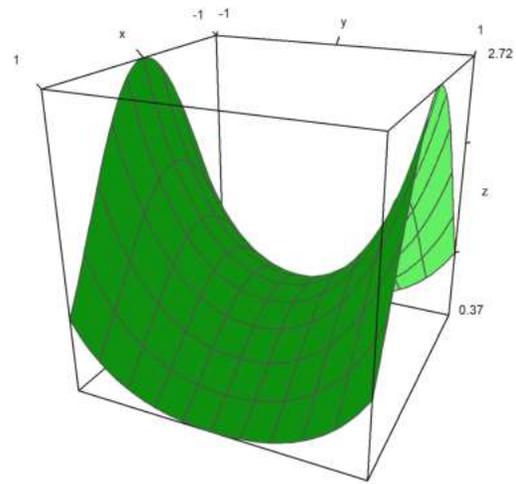
```
>plot3d("exp(x^2+y^2)",>user,...  
title="Coba Gerakkan"):
```

Coba Gerakkan



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

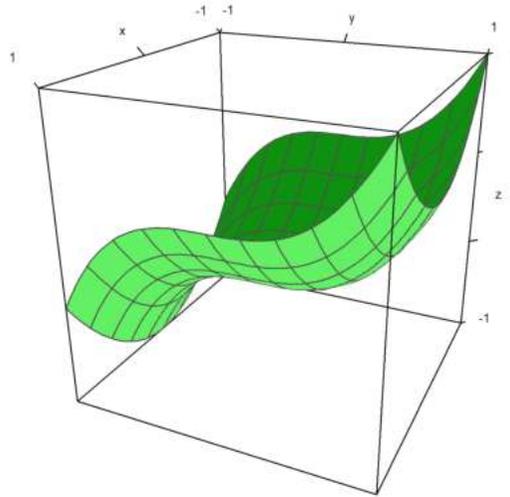
Coba Gerakkan



Animasi 3D

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

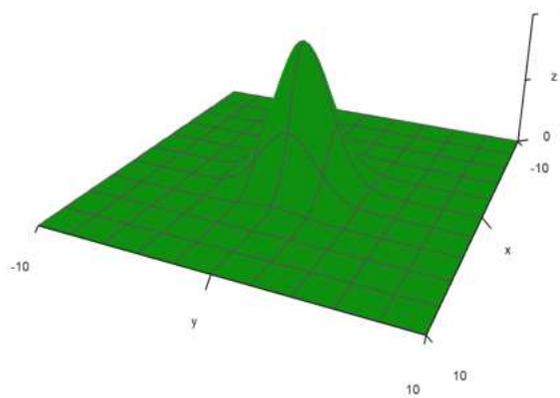
```
rotate("testplot"); testplot():
```



Fungsi rotate untuk memutar plot.
Fungsi ini akan membuat animasi plot 3D dari fungsi

yang berputar di sekitar sumbu z dari sudut 0 hingga 360 derajat

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



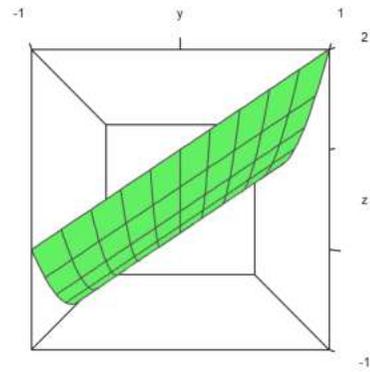
Ada beberapa parameter untuk menskalakan fungsi atau mengubah tampilan grafik.

`fscale`: menskalakan ke nilai fungsi (defaultnya adalah `<fscale>`) (0 = tidak ada, -1 = otomatis)

`scale`: angka atau vektor 1x2 untuk diskalakan ke arah x dan y. (0 = tidak ada penskalaan, 1 = default). Jika skala adalah vektor 1x3, itu akan penskalaan dalam setiap arah. Plot fungsi akan secara otomatis disesuaikan.

`frame`: jenis bingkai (default 1). Jika 0 (`<frame>`), tidak ada bingkai yang digambar

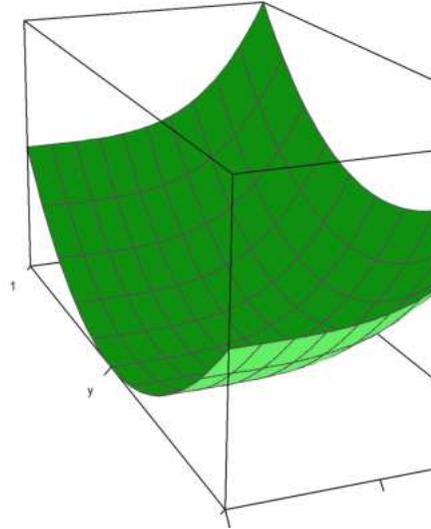
```
>plot3d("x^2+y", distance=3, zoom=1, angle=pi/2, height=0) :
```



Tampilan dapat diubah dengan berbagai cara.

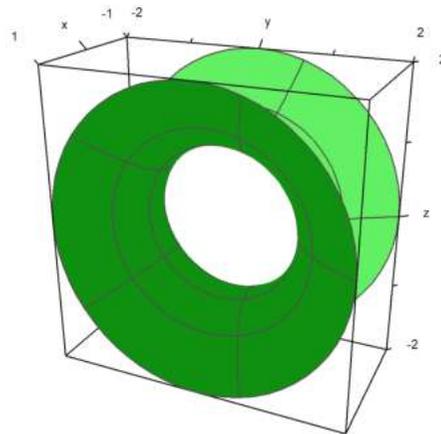
- distance: jarak pandang ke plot.
- zoom: nilai zoom.
- angle: sudut terhadap sumbu y negatif dalam radian.
- height: ketinggian tampilan dalam radian.

```
>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

Menggambar fungsi parametrik 3 dimensi (3D)

Persamaan parametrik adalah jenis persamaan matematika yang menggambarkan suatu objek dalam ruang, baik dua dimensi(bidang) atau tiga dimensi(ruang), dengan menggunakan parameter-parameter. Persamaan ini bergantung pada satu atau lebih parameter. Dengan mengubah nilai parameter tersebut, Anda dapat menghasilkan berbagai bentuk atau pola dari objek yang dijelaskan oleh persamaan tersebut.

Contoh 1 :

Sebagai contoh :

$$u(x, y) = \cos(x)\cos(y)$$

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

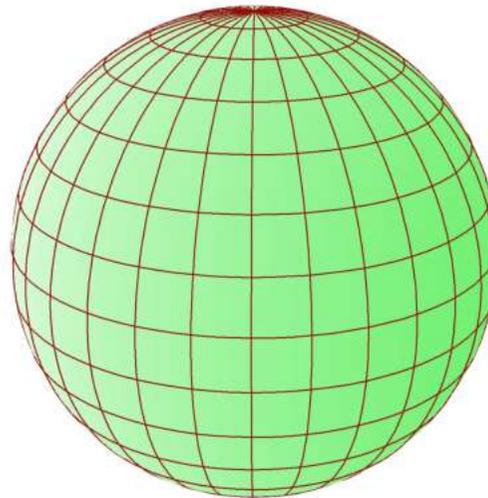
$$w(x, y) = \sin(y)$$

dengan

$$0 \leq x \leq 2\pi$$

$$-\frac{\pi}{2} \leq y \leq \frac{\pi}{2}$$

```
>plot3d("cos(x)*cos(y)", "sin(x)*cos(y)", "sin(y)", a=0, b=2*pi, c=pi/2, d=-pi/2, ...  
>hue,color=lightgreen, light=[0,1,0], <frame, ...  
n=90, grid=[15,30], wirecolor=red, zoom=4):
```



a dan b : parameter yang mengatur batasan rentang variabel x
c dan d : parameter yang mengatur batasan rentang variabel y
plot akan menggunakan model berbayang dengan menggunakan >hue dan juga dengan sumber cahaya tertentu dengan menggunakan light. Di sini saya ingin menghilangkan frame dengan menggunakan <frame. selanjutnya akan dibuat 15 garis lintang dan 30 garis bujur menggunakan grid[15,30].
wirecolor : mengatur warna garis pada plot

Contoh 2 :

$$u(x, y) = \sin(x)y\sin(x/2)$$

$$v(x, y) = \cos(x)y/2\cos(x/2)$$

$$w(x, y) = y/2\sin(x/2)$$

```
>aspect(16/9); allwindow; ...
x:=linspace(0,2*pi,100); y:=(-1:0.1:1)'; ...
plot3d(sin(x)*(y*sin(x/2)),cos(x)*(y/2*cos(x/2)),y/2*sin(x/2), ...
<frame,hue=2,max=0.5,scale=1.8):
```



allwindow : menampilkan layar penuh untuk plot
x:=linspace(0,2*pi,100) : perintah untuk membuat vektor x yang berisi 100 nilai yang sama jaraknya antara 0 hingga 2*pi
linspace : fungsi yang digunakan untuk menghasilkan vektor dengan nilai yang merata terdistribusi diantara dua titik.
y:=(-1:0.1:1)' : perintah untuk membuat vektor y yang berisi nilai-nilai dari -1 hingga 1 dengan selang 0.1
hue : mengaktifkan bayangan cahaya
max : menentukan kegelapan maksimal.

```
>aspect(16/9); allwindow; ...
x:=linspace(0,2*pi,100); y:=(-1:0.1:1)'; ...
plot3d(sin(x)*(y*sin(x/2)),cos(x)*(y/2*cos(x/2)),y/2*sin(x/2), ...
>lines,<frame,xmin=0,xmax=130,n=100):
```



lines : menampilkan garis dalam satu arah

```
>aspect(16/9); allwindow; ...
x:=linspace(0,2*pi,100); y:=(-1:0.1:1)'; ...
plot3d(sin(x)*(y*sin(x/2)),cos(x)*(y/2*cos(x/2)),y/2*sin(x/2), ...
>lines,<frame,xmin=0,xmax=130,n=100,>user):
```



>User :

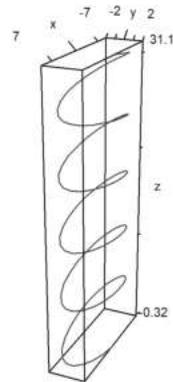
Contoh 3 :

$$7\sin(x)$$

$$2\cos(x)$$

$$x - 2/2\pi$$

```
>reset;...  
plot3d("7*sin(x)", "2*cos(x)", "x-2/2Pi", >lines, xmin=0, xmax=10pi, n=100, >user) :
```



Contoh 4:

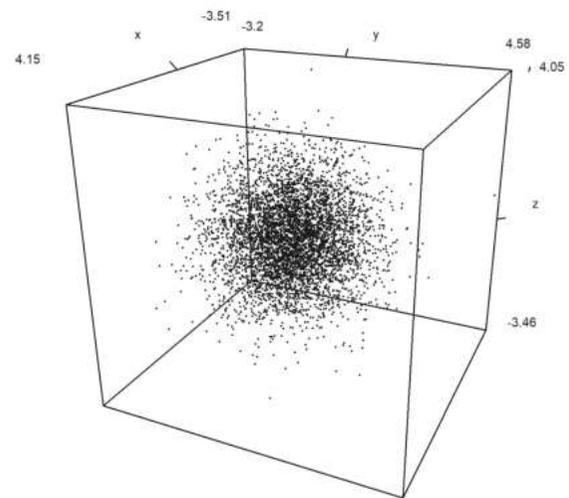
variabel D yang berisi data acak yang diambil dari distribusi normal dengan rata-rata 15 dan deviasi standar 5000.

D[2] : mengambil komponen kedua dari data D

D[4] : mengambil komponen keempat

D[6] : mengambil komponen keenam

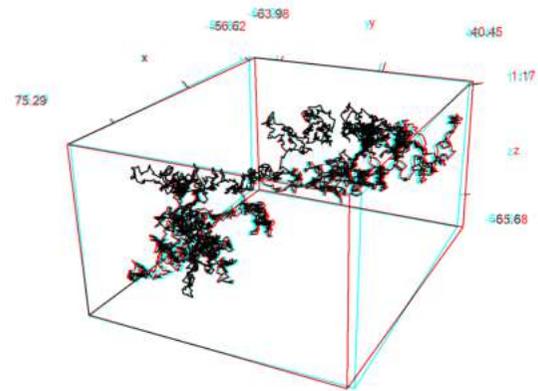
```
>D:=normal(15,5000); plot3d(D[2],D[4],D[6],>points,style="."):
```



>points : memplot titik-titik dalam 3D

E merupakan hasil dari jumlah kumulatif D. Ini berarti setiap elemen dari E adalah jumlah dari elemen-elemen sebelumnya pada D.

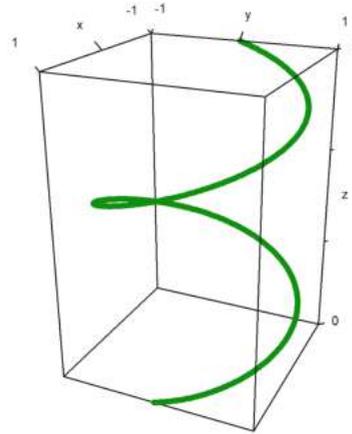
```
>D:=normal(15,5000); E:=cumsum(normal(15,5000)); ...  
plot3d(E[2],E[4],E[6],>wire, ...  
linewidth=1,>anaglyph, zoom=2.8):
```



- >wire : menghubungkan titik-titik dengan garis-garis
- >linewidth : lebar garis
- >anaglyph : plot akan ditampilkan dalam format anaglyph

Contoh 5:

```
>t=linspace(0,3pi,600); plot3d(cos(t),sin(t),t/pi,>wire,...  
linewidth=5,wirecolor=green):
```



```
>reset;
```

Menggambar Fungsi Implisit 3 Dimensi (3D)

Fungsi implisit 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.

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

solusi dari

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

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

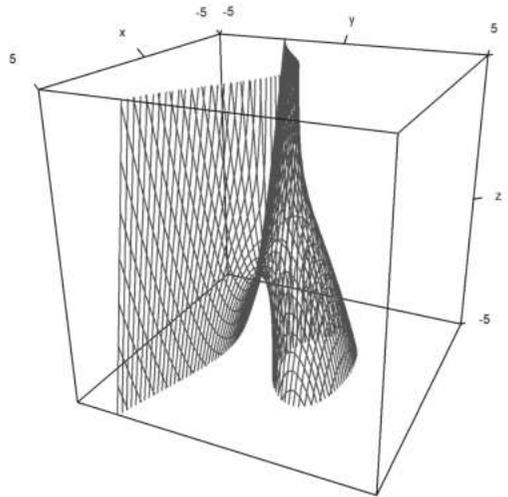
- implicit=1: potong sejajar dengan bidang y-z
- implicit=2: memotong sejajar dengan bidang x-z
- implicit=4: memotong sejajar dengan bidang x-y

Sebagai contoh :

Contoh 1:

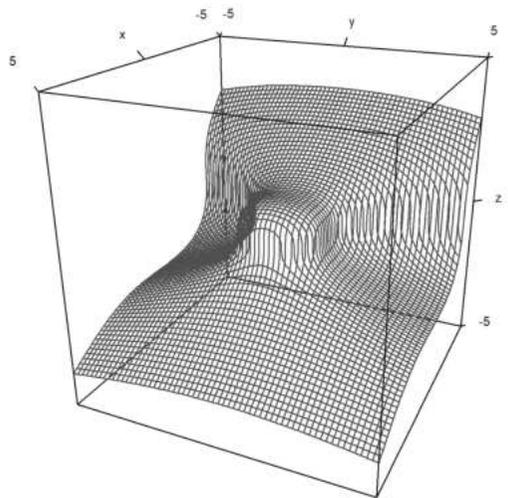
$$M = \{(x, y, z) : x^2 + y^3 + zy = 1\}$$

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



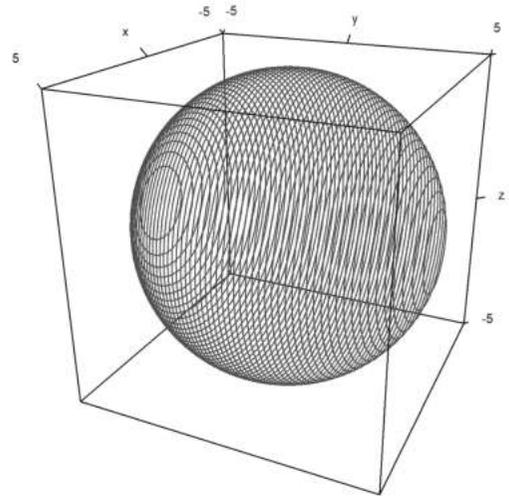
Contoh 2:

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



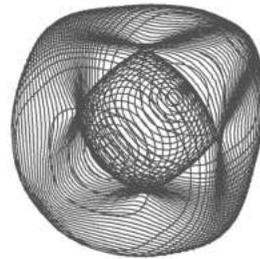
Contoh 3

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



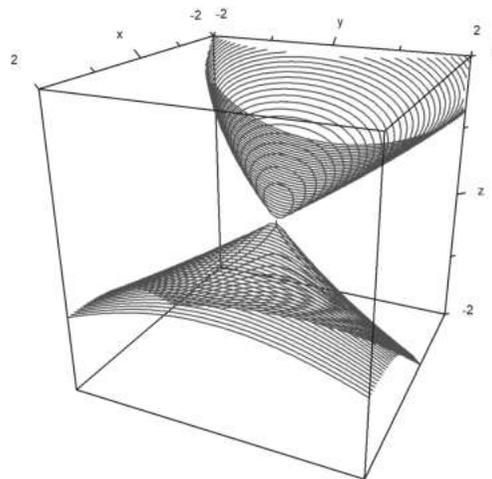
Contoh 4:

```
>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,<frame,>implicit,>user):
```



Contoh 5:

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



Menggambar Fungsi Implisit dengan Menggunakan Povray

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

Contoh :

$$x^2 + y^2 z^2 - 1 = 0$$

```
>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 : memuat file povray
>defaultpovray : Perintah berikut ini menghasilkan file povray di direktori pengguna Anda, dan menjalankan Povray untuk melacak sinar pada file ini.
>writeln(povsurface(...)) : perintah untuk menuliskan kode
povlook : mengatur tampilan warna
povbox(-2,2, ""): Ini mendefinisikan sebuah kotak (box) dengan panjang sisi 4 (dari -2 hingga 2) dan tidak memiliki tekstur atau warna.

contoh :

```
>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();
```

Command was not allowed!

exec:

```
return _exec(program,param,dir,print,hidden,wait);
```

povray:

```
exec(program,params,defaulthome);
```

Try "trace errors" to inspect local variables after errors.

povend:

```
povray(file,w,h,aspect,exit);
```

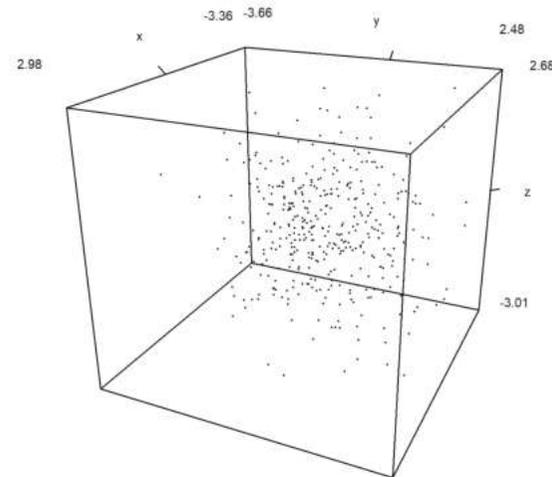
Sub-Topik 9 Menggambar Titik pada ruang 3D

Menggambar titik pada ruang tiga dimensi (3D) merupakan proses visualisasi titik dalam sistem koordinat tiga dimensi (3D). Ruang tiga dimensi (3D) memiliki tiga sumbu utama : sumbu x,y, dan z yang bersilangan tegak lurus satu sam lain. Titik dalam ruang tiga dimensi (3D) dapat didefinisikan dengan tiga koordinat tersebut.

Untuk menggambar/memplot data titik dalam ruang,kita membutuhkan tiga vektor untuk koordinat titik-titik tersebut.

Gayanya sama seperti gaya di plot2D, yaitu dengan `points=true`;

```
>n=400;...  
plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style=".") :
```



Penjelasan :

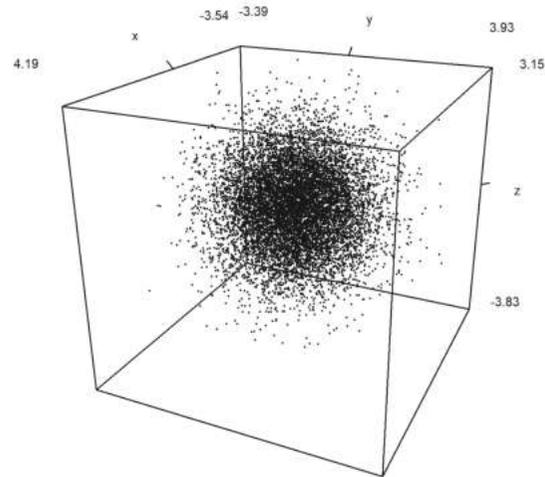
1. `n=...`; digunakan untuk menginisialisasi variabel `n` dengan nilai ... variabel ini akan digunakan sebagai penentuan jumlah titik yang akan digunakan dalam plot 3D.
2. `normal(1,n)`, Fungsi normal digunakan untuk menghasilkan nilai-nilai acak yang terdistribusi secara normal (gaussian) dengan rata-rata 1 dan deviasi standar 1. hal ini untuk mendapatkan koordinat x,y, dan z untuk plot 3D.

3. `plot3d(...)` merupakan fungsi yang digunakan untuk plot 3D dengan parameter-parameter, antara lain yaitu :

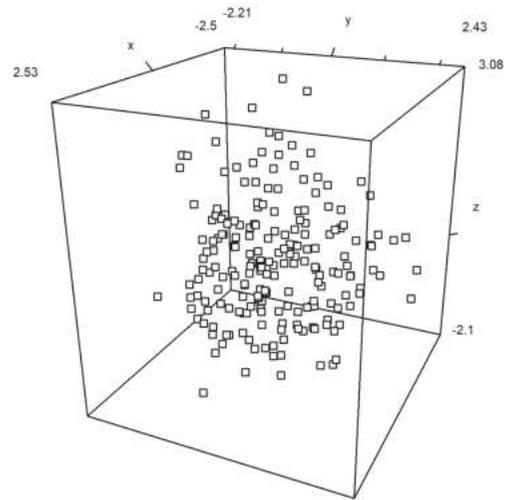
- `points=true` (mengatur agar titik-titik data ditampilkan dalam plot tersebut).
- `style` (untuk mengatur gaya titik yang akan ditampilkan dalam plot).

Berikut akan ditunjukkan contoh lainnya:

```
>n=10000;...  
plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style=".") :
```

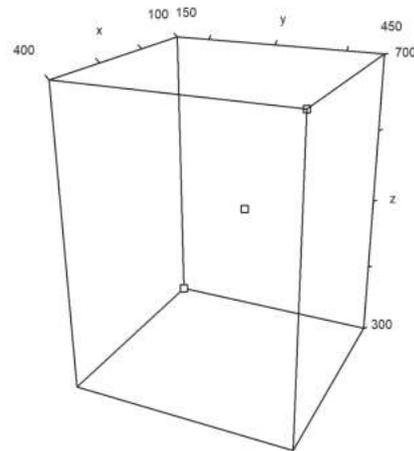


```
>n=200;...  
plot3d(normal(1,n),normal(1,n),normal(1,n),points=true,style=",") :
```



Pada contoh berikut akan ditunjukkan titik yang ditentukan dengan vektor baris x,y,z

```
>x=[100,200,400]; y=[150,300,450]; z=[300,500,700]; plot3d(x,y,z,points=true,style=""):
```

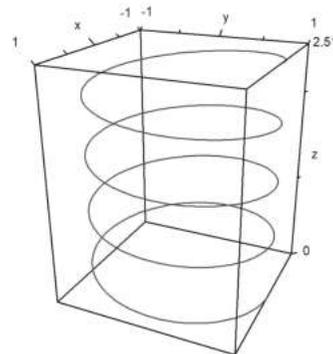


Dimungkinkan juga untuk memplot kurva dalam 3D. Dalam hal ini, lebih mudah untuk menghitung terlebih dahulu titik-titik kurva. Untuk kurva pada bidang kita menggunakan barisan koordinat dan parameter `wire=true`.

```
>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=2):
```



Contoh lainnya :
pertama kita tentukan terlebih dahulu titik-titik kurvanya

```
>t=linspace(0,6pi,700)
```

```
[0, 0.0269279, 0.0538559, 0.0807838, 0.107712, 0.13464, 0.161568,  
0.188496, 0.215423, 0.242351, 0.269279, 0.296207, 0.323135,  
0.350063, 0.376991, 0.403919, 0.430847, 0.457775, 0.484703,
```

```

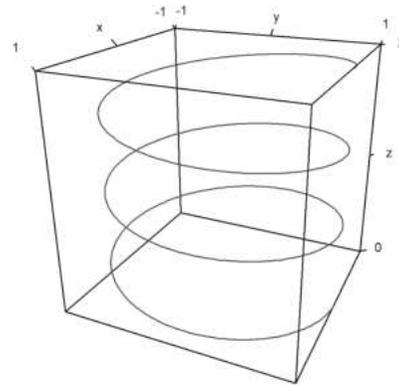
0.511631, 0.538559, 0.565487, 0.592415, 0.619343, 0.64627,
0.673198, 0.700126, 0.727054, 0.753982, 0.78091, 0.807838,
0.834766, 0.861694, 0.888622, 0.91555, 0.942478, 0.969406,
0.996334, 1.02326, 1.05019, 1.07712, 1.10405, 1.13097, 1.1579,
1.18483, 1.21176, 1.23869, 1.26561, 1.29254, 1.31947, 1.3464,
1.37332, 1.40025, 1.42718, 1.45411, 1.48104, 1.50796, 1.53489,
1.56182, 1.58875, 1.61568, 1.6426, 1.66953, 1.69646, 1.72339,
1.75032, 1.77724, 1.80417, 1.8311, 1.85803, 1.88496, 1.91188,
1.93881, 1.96574, 1.99267, 2.0196, 2.04652, 2.07345, 2.10038,
2.12731, 2.15423, 2.18116, 2.20809, 2.23502, 2.26195, 2.28887,
2.3158, 2.34273, 2.36966, 2.39659, 2.42351, 2.45044, 2.47737,
2.5043, 2.53123, 2.55815, 2.58508, 2.61201, 2.63894, 2.66587,
2.69279, 2.71972, 2.74665, 2.77358, 2.80051, 2.82743, 2.85436,
2.88129, 2.90822, 2.93515, 2.96207, 2.989, 3.01593, 3.04286,
3.06978, 3.09671, 3.12364, 3.15057, 3.1775, 3.20442, 3.23135,
3.25828, 3.28521, 3.31214, 3.33906, 3.36599, 3.39292, 3.41985,
3.44678, 3.4737, 3.50063, 3.52756, 3.55449, 3.58142, 3.60834,
... ]

```

```

>t=linspace(0,6pi,700);...
plot3d(sin(t),cos(t),t/3pi,>wire, zoom=2):

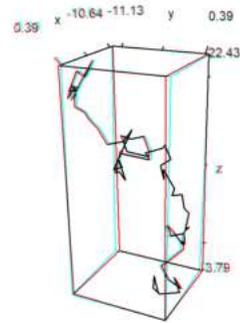
```



```

>X=cumsum(normal(7,70)); plot3d(X[1],X[2],X[3],>anaglyph,>wire, zoom=2):

```



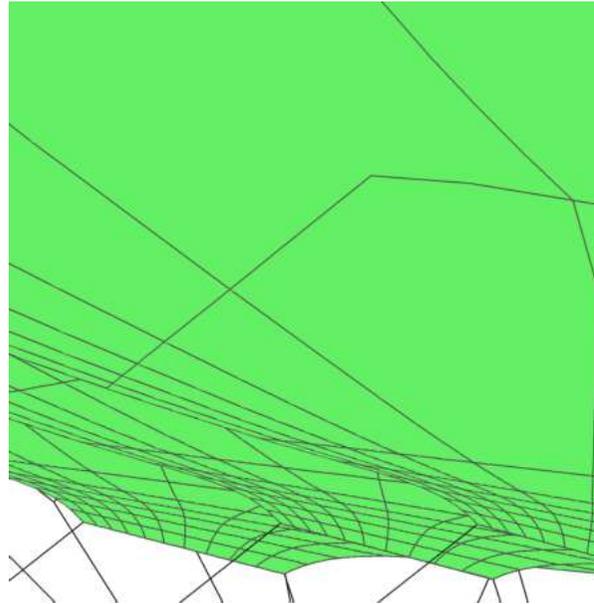
writeaxes : perintah untuk menambah koordinat

Mengatur tampilan, warna, dan sudut pandang gambar permukaan 3D

Berikut adalah parameter untuk mengatur tampilan grafik.

- fscale : menskalakan ke nilai fungsi (default is <fscale)
- scale : angka atau vektor 1x2 untuk menskalakan ke arah x dan y
- frame : jenis bingkai (default 1)

```
>figure(3,3); ...
for n=1 to 9; figure(n); plot3d("x^3+2x+1",fscale=1,scale=1:n,frame=1); end; ...
figure(0):
```



untuk mengatur tampilan pada bangun ruang 3D di eluer yaitu dengan berbagai cara, yaitu:

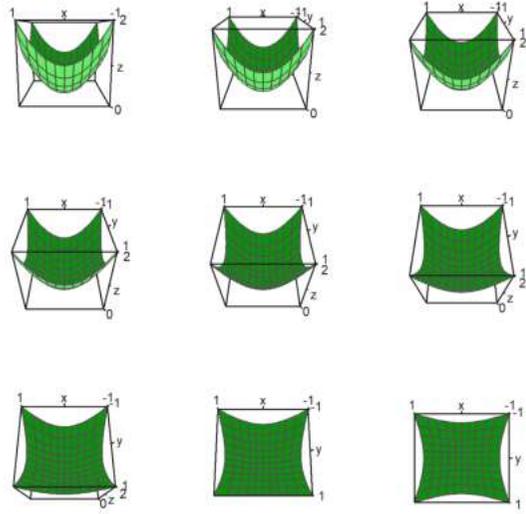
- distance : untuk mengatur jarak pandang ke plot
- zoom : untuk mengatur zoom hasilnya
- angle : untuk mengatur sudut pandang
- height untuk mengatur ketinggian pandangan dalam radian

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

```
>view
```

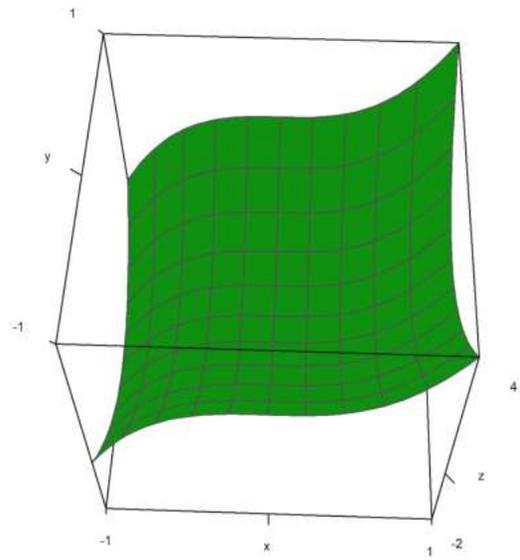
```
[5, 2.6, 2, 0.4]
```

```
>figure(3,3); ...  
for n=1 to 9; figure(n); plot3d("x^2+y^2",distance=5,angle=pi,height=10°*n); end; ...  
figure(0):
```



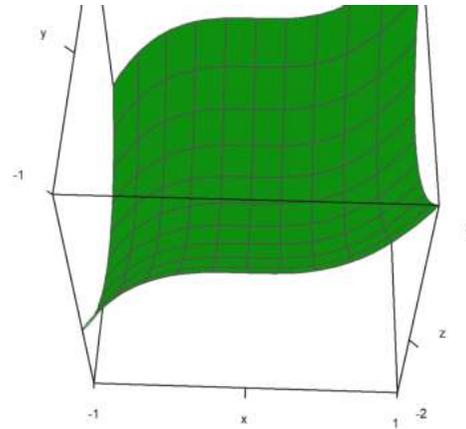
contoh lain

```
>reset; ...
plot3d("2*x^3+2*y^2", distance=7, zoom=4, angle=pi/90, height=1):
```



Agar plot terlihat berbeda kita dapat memindahkan bagian tengahnya dengan parameter "center"

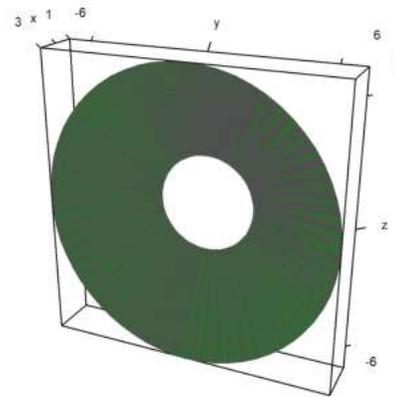
```
>plot3d("2*x^3+2*y^2",distance=7,zoom=4,angle=pi/90,height=1, ...  
center=[0,1,0]):
```



Selain itu cara untuk mengatur tampilan dari sisi lain kita dapat menggunakan parameter "rotate"

- rotate=1 : untuk memutar pada sumbu x
- rotate=2 : untuk memutar pada sumbu z

```
>plot3d("2*x",a=1,b=3,rotate=1,grid=1000):
```



Selanjutnya untuk mengubah rona suatu warna atau rona warna spektral. Euler dapat menggambar ketinggian fungsi pada plot dengan arsiran. Di semua plot 3D, Euler dapat menghasilkan anaglyph.

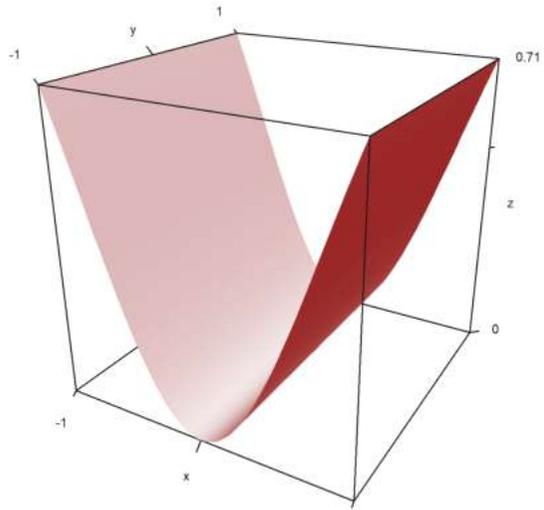
- >hue :Mengaktifkan bayangan cahaya dan memberi warna sesuai ketinggian.
- >contour : Membuat plot garis kontur otomatis pada plot.
- >spectral : Membuat warna spektral pada plot
- level=...(atau levels): A Vektor nilai garis kontur.
- color : mengubah warna selain warna spektral

Sebagai contohnya :

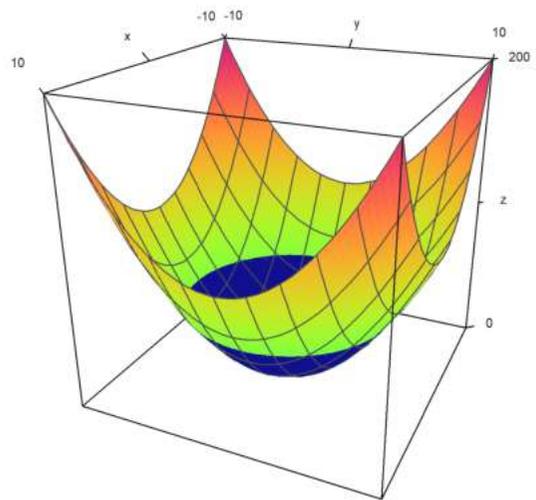
Kita akan menggambarkan

Dengan menggunakan >hue,>contour,>spectral dan color=... untuk mengubah warna serta memberi bayangan pada gambar tersebut.

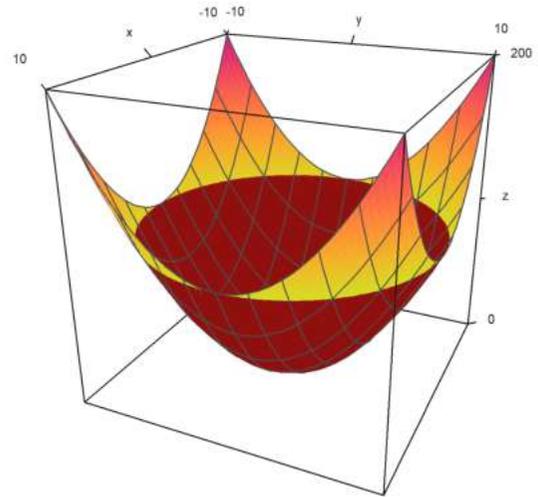
```
>plot3d("sin(x)^2", angle=pi/6, >hue, color=red) :
```



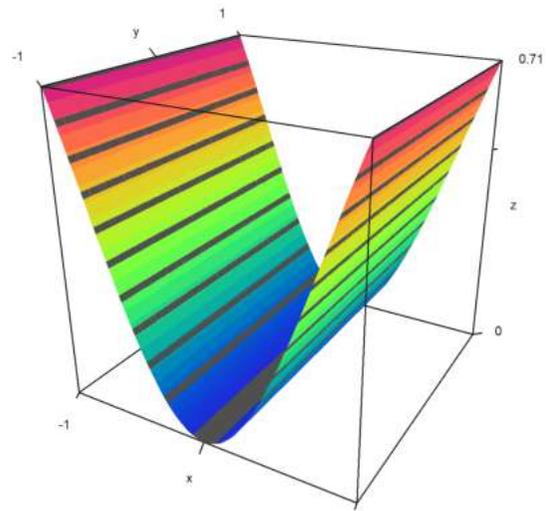
```
>plot3d("x^2+y^2",-10,10,-10,10,level=[0.1,0.2;50],grid=10,spectral=5,contourcolor=blue):
```



```
>plot3d("x^2+y^2",-10,10,-10,10,level=[0.1,0.5;100],grid=10,spectral=5,contourcolor=red):
```



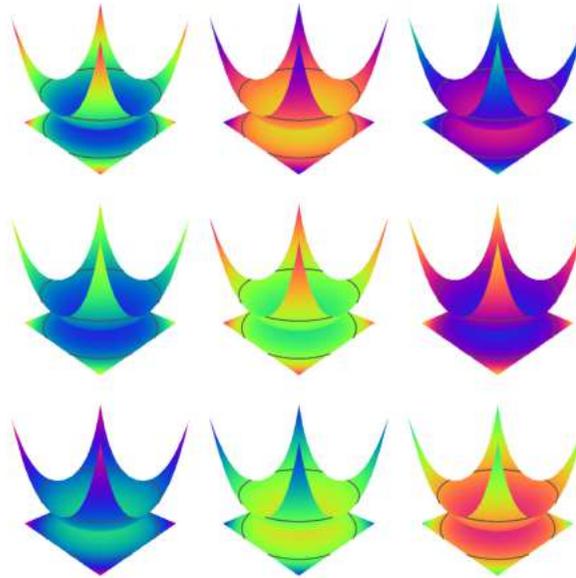
```
>plot3d("sin(x)^2",angle=pi/6,>spectral,>contour):
```



cp: menggambar bidang kontur di bawah plot (>cp)

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

```
angle=pi/4,level=3); end; ...  
figure(0):
```



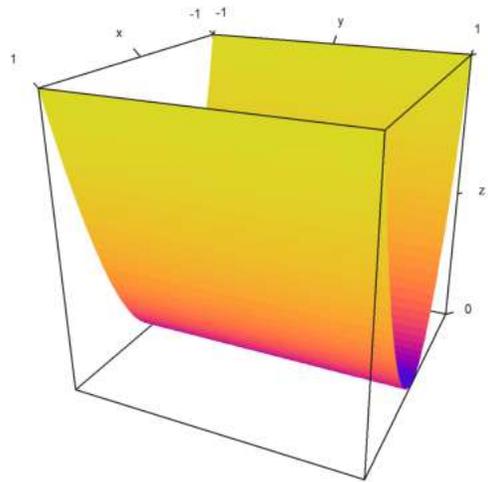
Sumber cahaya dapat diubah dengan I dan tombol kursor selama interaksi pengguna. Itu juga dapat diatur dengan parameter.

- light: arah
- amb: cahaya sekitar antara 0 and 1

Catatan : program tidak membuat perbedaan antara sisi plot. Tidak ada bayangan. Untuk ini, Anda memerlukan Povray.

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

Press 1 and cursor keys (return to exit)



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 topologi 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.

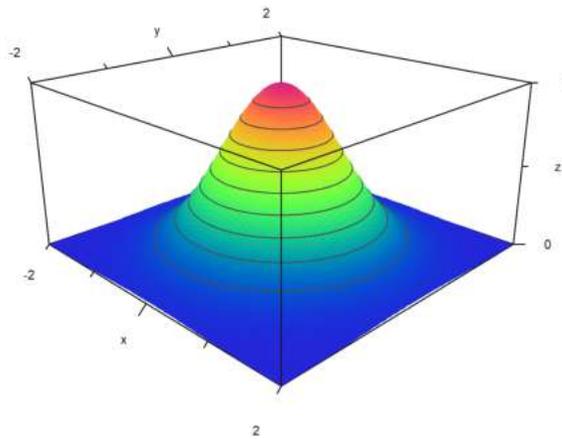
Untuk membuat plotnya, Euler menambahkan garis grid. Sebaliknya dimungkinkan untuk menggunakan garis datar dan rona satu warna atau rona warna spektral. Euler dapat menggambar ketinggian fungsi pada plot dengan arsiran. Di semua plot 3D, Euler dapat menghasilkan anaglyph merah/cyan.

->hue: Mengaktifkan bayangan cahaya, bukan kabel.
->contour: Membuat plot garis kontur otomatis pada plot.
- level=... (atau levels): Vektor nilai garis kontur.

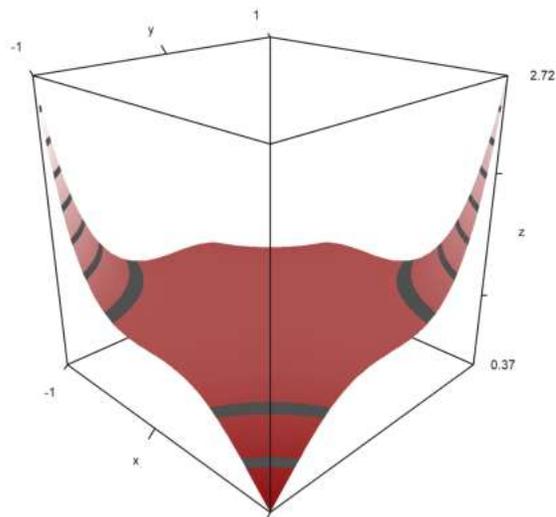
Standarnya adalah level="auto", yang menghitung beberapa garis level secara otomatis. Seperti yang Anda lihat di plot, level sebenarnya adalah rentang level.

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.

```
>plot3d("exp(-x^2-y^2)", r=2, n=100, level="thin", >contour, >spectral, angle=pi/4, height=20°, fscale=1, scale=1.1):
```



```
>plot3d("exp(x^3*y^3)", angle=pi/4, >contour, color=red):
```

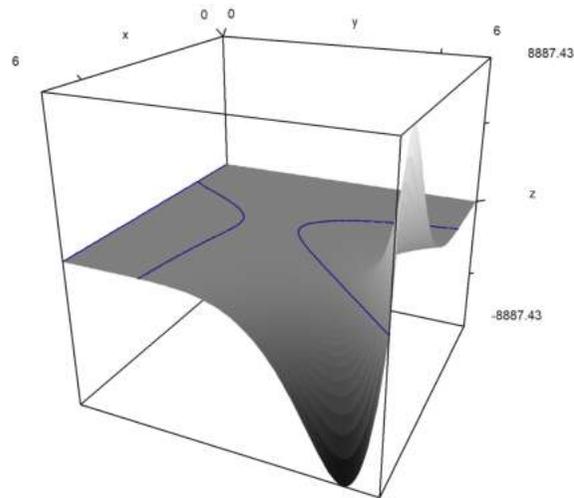


Pada contoh berikut, kita memplot himpunan, di mana

Kami menggunakan satu garis tipis untuk garis level.

Level disini adalah nilai dari $f(x,y)$, dalam soal ini nilai level=0

```
>plot3d("x^y-y^x", level=1, a=0, b=6, c=0, d=6, contourcolor=blue, n=100) :
```



Menggambar Grafik Tiga Dimensi dalam Modus Anaglif

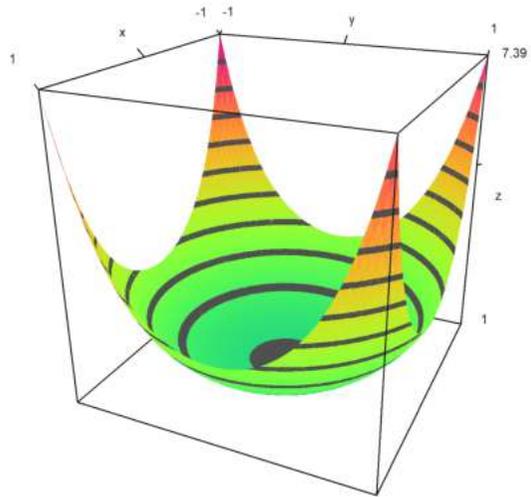
Modus anaglif adalah salah satu teknik yang digunakan untuk menciptakan efek tiga dimensi pada gambar atau video dengan menggunakan dua warna yang berbeda. Teknik ini memanfaatkan persepsi mata manusia yang dapat membedakan warna merah dan biru (hijau/cyan) untuk menciptakan efek kedalaman.

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.

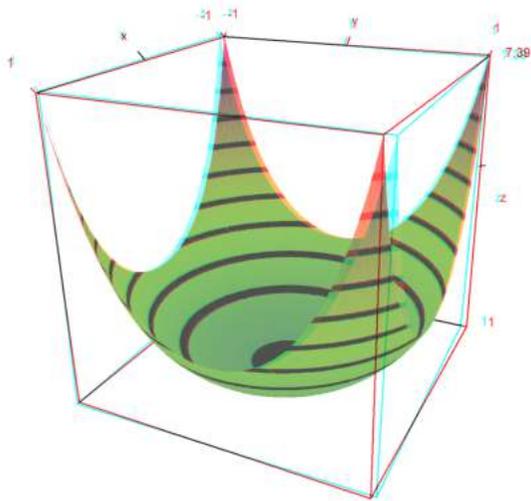
Namun, terdapat beberapa variasi dalam penggunaan warna pada kacamata anaglif, terutama dalam penggunaan lensa merah dan biru. Misalnya, ada juga kacamata anaglif yang menggunakan lensa merah dan cyan (biru muda) sebagai gantinya. Warna lensa yang digunakan dapat bervariasi tergantung pada preferensi 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("exp(x^2+y^2)", >contour, spectral=5) :
```



```
>plot3d("exp(x^2+y^2)",>contour,spectral=5,>anaglyph):
```



>

Menggambar Diagram Batang Pada 3D

Diagram batang, juga dikenal sebagai "bar chart" dalam bahasa Inggris, adalah jenis grafik yang digunakan untuk memvisualisasikan data kategori atau data diskrit. Diagram ini menggambarkan data dalam bentuk batang vertikal, di mana tinggi batang tersebut mewakili nilai atau frekuensi dari data tersebut.

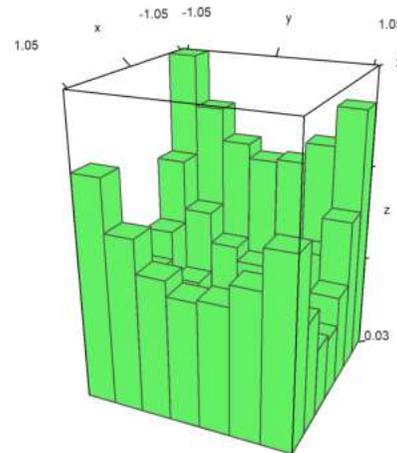
Plot bar juga dimungkinkan. Untuk ini, kita harus menyediakan

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

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

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

```
>x=-1:0.3:1; y=x'; z=2*x^2+y^2; ...  
xa=(x|1.1)-0.05; ya=(y_1.1)-0.05; ...  
plot3d(xa,ya,z,bar=true):
```



>

> x = -1:0.3:1; Kode ini membuat larik x yang berisi angka dari -1 hingga 1 dengan selang 0.3. Hasilnya adalah x akan berisi [-1, -0.7, -0.4, ..., 1].

> y = x'; Ini menghasilkan larik y yang merupakan transposisi dari larik x, sehingga y akan memiliki nilai yang sama dengan x, yaitu [-1, -0.7, -0.4, ..., 1].

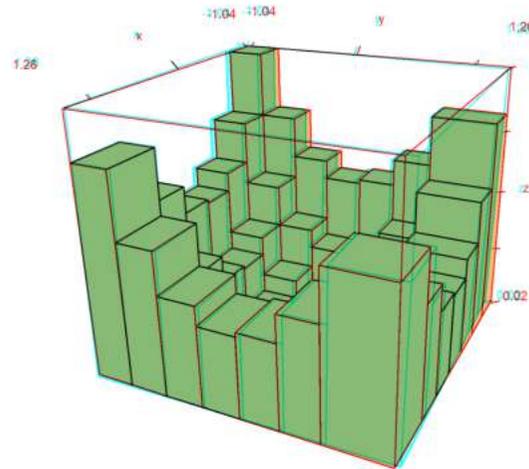
> z = 2*x^2 + y^2; Di sini, Anda menghitung larik z berdasarkan ekspresi matematika $2x^2 + y^2$. Ini berarti setiap elemen z adalah hasil dari perhitungan yang melibatkan kuadrat elemen-elemen yang sesuai dari x dan y.

> xa = (x|1.1) - 0.05; Ini mencoba mengubah larik x dengan menggantikan setiap elemen yang kurang dari 1.1 dengan hasil perhitungan $(x|1.1) - 0.05$. Operasi $(x|1.1)$ mungkin berarti pengambilan nilai maksimum antara setiap elemen x dan 1.1, dan kemudian dikurangi 0.05.

> ya = (y_1.1) - 0.05; Ini mencoba mengubah larik y dengan mengurangi 0.05 dari setiap elemen yang ada pada larik y setelah dinaikkan pangkat sebanyak 1.1.

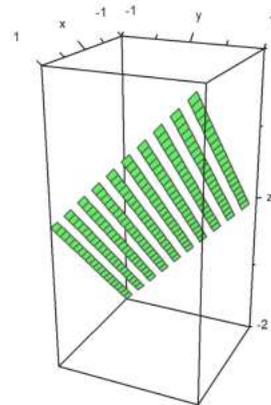
> plot3d(xa, ya, z, bar=true); Ini adalah perintah untuk membuat plot 3D. Plot ini akan menggunakan data dari larik xa, ya, dan z sebagai data inputnya. Opsi bar=true mungkin digunakan untuk menghasilkan plot batang 3D.

```
>x=-1:0.3:1; y=x'; z=x^2+y^2; ...  
xa=(x|1.3)-0.04; ya=(y_1.3)-0.04; ...  
plot3d(xa, ya, z, bar=true, >anaglyph):
```



dengan menggunakan >anaglyph, kita dapat mengaktifkan efek anaglyph pada plot, yang biasanya digunakan dalam pemrosesan citra 3D untuk memberikan efek tiga dimensi saat dilihat dengan kacamata anaglyph.

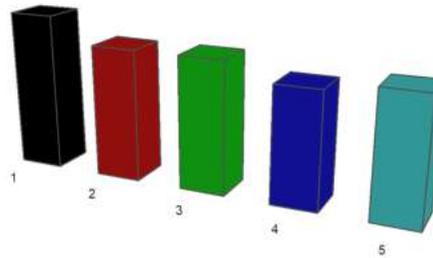
```
>x=-1:0.1:1; y=x'; z=x+y; d=zeros(size(x)); ...  
plot3d(x, y, z, disconnect=2:2:20):
```



dengan perintah `disconnect`, kita dapat memutus atau membagi garis-garis antara titik-titik data tertentu. `> d = zeros(size(x));` Ini membuat larik `d` yang berisi angka nol dengan ukuran yang sama seperti larik `x`. Larik `d` mungkin akan digunakan untuk menggambarkan efek khusus pada plot. `> plot3d(x, y, z, disconnect=2:2:20);` Ini adalah perintah untuk membuat plot 3D. Plot ini akan menggunakan data dari larik `x`, `y`, dan `z` sebagai data inputnya. Parameter `disconnect=2:2:20` menginstruksikan plot untuk memutuskan garis-garis antara titik-titik data dengan jarak 2 (indeks ke-2, ke-4, ke-6, dst.) hingga indeks ke-20, menciptakan beberapa segmen diskrit dalam plot.

Dengan menggunakan perintah `>columnplot3d`. Kita dapat juga membuat diagram batang. Fungsi ini akan menunjukkan diagram batang dengan bentuk 3 dimensi dengan ketinggian `z[i,j]`. `z` dapat berupa matriks bilangan riil `nxm`.

```
>Z=intrandom(5,100,6); v=zeros(5,1); ...
loop 1 to 5; v[#]=getmultiplicities(1:1,Z[#]); end; ...
columnplot3d(v',scols=[1:5],angle=25°,height=20°,zoom=3;ccols=[1:5]):
```



> Z = intrandom(5, 100, 6); Kode ini menghasilkan matriks Z dengan ukuran 5x6 yang berisi bilangan bulat acak antara 5 dan 100.

> v = zeros(5, 1); Ini menginisialisasi matriks v dengan ukuran 5x1 yang diisi dengan angka nol.

> loop 1 to 5; v[#] = getmultiplicities(1:1, Z[#]); end; Ini adalah perulangan yang berjalan dari 1 hingga 5. Di setiap iterasi, perulangan ini menghitung "multiplicitas" atau jumlah kemunculan angka 1 dalam matriks Z pada baris yang sesuai, dan hasilnya disimpan dalam matriks v pada baris yang sesuai juga.

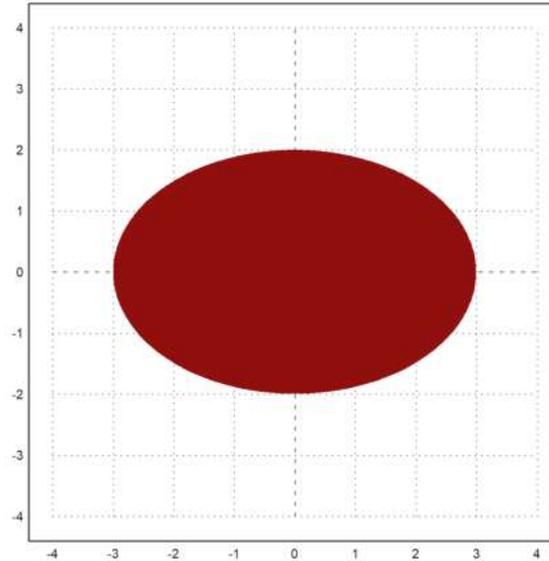
> columnsplot3d(v', scols=[1:5], angle=25°, height=20°, zoom=3, ccols=[1:5]); Ini adalah perintah untuk membuat plot 3D. Plot ini akan menggunakan data dari matriks v' (transposisi dari matriks v), dengan kolom-kolom yang dipilih dari 1 hingga 5 (scols=[1:5]). Selain itu, parameter angle=25° mengatur sudut pandang plot dalam ruang 3D, height=20° mengatur tinggi pandangan plot, zoom=3 mengatur tingkat zoom, dan ccols=[1:5] mungkin mengatur warna kolom.

Jadi, kode ini menghasilkan plot 3D yang memvisualisasikan data dalam matriks v setelah menghitung jumlah kemunculan angka 1 dalam matriks Z. Plot ini mungkin menampilkan kolom-kolom 3D dengan berbagai warna yang sesuai.

Menggambar Permukaan Benda Putar

Menggambar Permukaan Benda Putar merupakan cara kita mengetahui bentuk 3d dari suatu plot2d dengan memutarnya terhadap suatu sumbu

```
> plot2d("((x^2)/3^2)+((y^2)/2^2)-1",r=4, ...
style="#",color=red,<outline, ...
level=[-2;0],n=120):
```



> plot2d("((x^2)/3^2) + ((y^2)/2^2) - 1", r=4, ...: Ini adalah perintah untuk membuat plot 2D dari persamaan elips di mana radius maksimumnya adalah 4.

> style="": Dalam parameter style, Anda mungkin telah mengatur tampilan plot ke tampilan default (kosong) atau tanpa gaya tertentu.

> color=cyan: Ini mengatur warna plot, dalam hal ini, plot akan berwarna biru muda (cyan).

<outline: Ini mungkin digunakan untuk menyoroti atau menggambar garis tepi dalam plot.

> level=[-2;0]: Parameter ini menunjukkan level atau rentang nilai yang akan diplot dalam grafik. Di sini, nilai yang akan diplot berada dalam rentang -2 hingga 0.

> n=120: Ini mengatur jumlah titik data yang akan digunakan untuk membuat plot, dalam hal ini, 120 titik data akan digunakan.

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

$$\frac{y^2}{4} + \frac{x^2}{9} - 1$$

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

Selanjutnya kita atur

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

$$\frac{(\cos(2a) + 1) r^2}{18} + \frac{(1 - \cos(2a)) r^2}{8} - 1$$

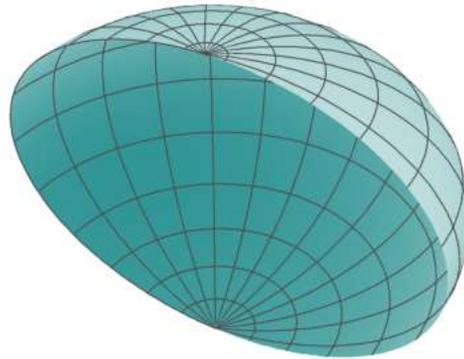
> function fr(r, a): Ini adalah deklarasi fungsi yang dinamakan fr dengan dua parameter, yaitu r dan a.

> ekspresi with [x = r * cos(a), y = r * sin(a)]: Ini adalah bagian dari perintah yang mendefinisikan ekspresi yang akan dihitung dalam fungsi fr. Dalam ekspresi ini, x dan y digantikan oleh nilai-nilai yang sesuai, yaitu x = r * cos(a) dan y = r * sin(a).

> trigreduce: Ini adalah perintah yang mungkin digunakan untuk menggantikan ekspresi trigonometri dengan bentuk yang lebih sederhana atau tereduksi.

> \$fr(r, a): Ini adalah hasil dari ekspresi yang menghitung fungsi fr dengan parameter r dan a. Hasil ini akan menggantikan nilai ekspresi yang telah didefinisikan dalam fungsi fr.

```
>function map f(a) := bisect("fr",0,4;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=cyan,zoom=4,amb=0,max=0.7,grid=12,angle=120°,height=35°):
```



> function map f(a) := bisect("fr", 0, 4; a):: Ini adalah definisi fungsi map yang disebut f. Fungsi ini menerima parameter a dan memanggil fungsi fr dengan nilai a sebagai argumen. Fungsi bisect digunakan untuk mencari akar atau solusi dari fungsi fr di dalam rentang [0, 4] dengan menggunakan nilai a.

> t = linspace(-pi/2, pi/2, 100):: Ini menghasilkan larik t yang berisi 100 titik dalam rentang dari -p/2 hingga p/2.

> r = f(t):: Ini menghitung nilai r dengan memanggil fungsi f (yang dalam hal ini mencari akar fungsi fr) dengan menggunakan nilai-nilai dalam larik t.

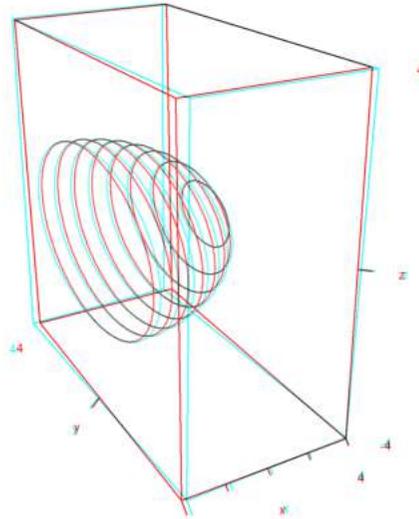
> s = linspace(pi, 2 * pi, 100):: Ini menghasilkan larik s yang berisi 100 titik dalam rentang dari p hingga 2p.

> plot3d(r * cos(t) * sin(s), r * cos(t) * cos(s), r * sin(t), ...: Ini adalah perintah untuk membuat plot 3D dengan koordinat yang dihitung dari nilai-nilai r, t, dan s. Plot ini menggunakan warna yang berubah-ubah (hue) dan memiliki beberapa parameter lainnya seperti frame, zoom, angle, height, dan lainnya.

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

```
>function f(x,y,z) ...
r=x^2+y^2;
return (r/3^2)+(z^2/2^2)-1;
endfunction
```

```
>plot3d("f(x,y,z)", ...
xmin=0,xmax=4,ymin=-4,ymax=4,zmin=-4,zmax=4, ...
implicit=1,angle=-30°,zoom=3,n=[10,60,60],frame=1,>anaglyph):
```



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

> n=[10, 60, 60]: Parameter ini mengatur jumlah titik data dalam masing-masing arah x, y, dan z dalam plot 3D. Dalam hal ini, ada 10 titik dalam arah x, 60 titik dalam arah y, dan 60 titik dalam arah z.

> frame=1: Ini mungkin adalah parameter yang mengatur tampilan frame atau frame plot.

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 sub-direktori "bin" dari Povray ke dalam jalur lingkungan, atau atur variabel "defaultpovray"

dengan jalur penuh yang mengarah ke "pvengine.exe". Antarmuka Povray dari Euler menghasilkan file Povray di direktori home pengguna, dan memanggil Povray untuk mengurai berkas-berkas ini. Nama file default adalah current.pov, dan direktori defaultnya adalah eulerhome(), biasanya c:\Pengguna\Nama Pengguna\Euler. Povray menghasilkan sebuah file PNG, yang dapat dimuat oleh Euler ke dalam buku catatan. Untuk membersihkan berkas-berkas ini, gunakan povclear(). Fungsi pov3d() memiliki semangat yang sama dengan plot3d. Fungsi ini dapat menghasilkan grafik dari sebuah fungsi $f(x,y)$, atau sebuah 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 akan dimulai, dan PNG akan dimasukkan ke dalam buku catatan Euler. Fungsi objek memiliki parameter yang disebut "look", yang membutuhkan sebuah string dengan kode povray untuk tekstur dan hasil akhir dari o

```
>load povray;
```

Pastikan direktori bin Povray berada di dalam path. Jika tidak, edit variabel berikut sehingga berisi path ke eksekusi povray.

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

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

Untuk kesan pertama, kita merencanakan sebuah fungsi sederhana. Perintah berikut ini menghasilkan sebuah file povray pada direktori user Anda, dan menjalankan Povray untuk melacak sinar pada 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 dijalankan. Anda dapat menekan cancel untuk menghentikannya lebih lanjut. pertanyaan. Anda mungkin harus menekan OK pada jendela Povray untuk mengetahui dialog start-up Povray

```
>pov3d("x^4+y^4",zoom=4);
```

```
Command was not allowed!  
exec:  
    return _exec(program,param,dir,print,hidden,wait);  
povray:  
    exec(program,params,defaulthome);  
Try "trace errors" to inspect local variables after errors.  
pov3d:  
    if povray then povray(currentfile,w,h,w/h); endif;
```

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

```
>
```

```
>pov3d("x^2+y^3",axiscolor=red,angle=45°, ...  
look=povlook(green,0.2),level=-1:0.5:1, zoom=3.8);
```

```
Command was not allowed!  
exec:  
    return _exec(program,param,dir,print,hidden,wait);  
povray:  
    exec(program,params,defaulthome);  
Try "trace errors" to inspect local variables after errors.  
pov3d:  
    if povray then povray(currentfile,w,h,w/h); endif;
```

Kadang-kadang perlu untuk mencegah penskalaan fungsi, dan menskalakan fungsi dengan tangan. Kami memplot kumpulan titik pada bidang kompleks, di mana hasil kali jarak ke 1 dan -1 sama dengan 1. s equal to 1. lainnya. Kita juga dapat menambahkan garis level ke plot fungsi.

```
>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);
```

