

Nama : Rofi'ah 'Aisy Zukhruf
NIM : 23030630019
Kelas : Matematika B

Menggambar Plot 3D dengan EMT

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

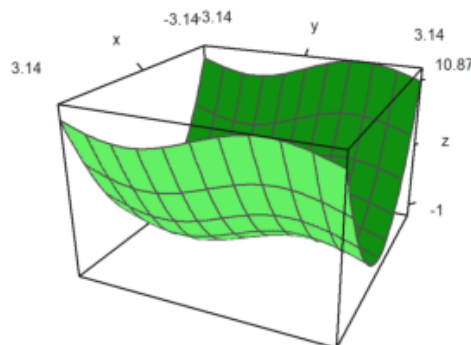
Euler menggambar fungsi seperti itu menggunakan algoritma pengurutan untuk menyembunyikan bagian di latar belakang. Secara umum, Euler menggunakan proyeksi pusat. Defaultnya adalah dari kuadran x-y positif ke arah asal $x = y = z = 0$, tetapi sudut = 0 ? melihat dari arah sumbu y. Sudut pandang dan ketinggian dapat diubah.

Euler bisa merencanakan

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

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

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



Latihan Soal:

1. plot fungsi

$$z = x^2 + \cos(y)$$

dalam rentang $-3 \leq x \leq 3$ dan $-2\pi \leq y \leq 2\pi$. Gantilah sudut pandang untuk memperlihatkan plot dengan lebih baik.

Jawab :

```
>aspect(1.5);  
>plot3d("x^2+cos(y)",r=[-3, 3, -2*pi, 2*pi])
```

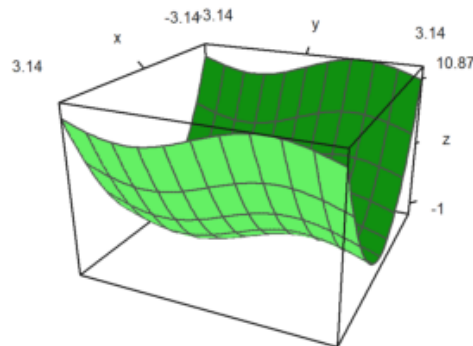
Index 5 out of bounds!

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

plot3d:

```
zmin=cz-r{5}; zmax=cz+r{6};
```

```
>view(30, 30):
```



2. Buatlah plot 3D dari fungsi

$$z = \sin(x) + \sin(y)$$

pada rentang $-p \leq x \leq p$ dan $-\pi \leq y \leq \pi$. Atur warna permukaan agar lebih menarik.

JAWAB :

```
>aspect(1.5);  
>plot3d("sin(x) + sin(y)", r = [-pi, pi, -pi, pi], color="blue");
```

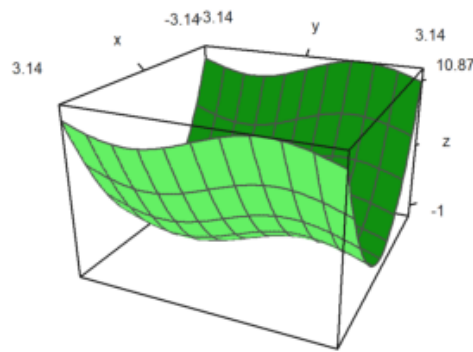
Index 5 out of bounds!

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

plot3d:

```
zmin=cz-r{5}; zmax=cz+r{6};
```

```
>view(45, 20):
```



Fungsi Dua Variabel

Untuk grafik suatu fungsi, gunakan

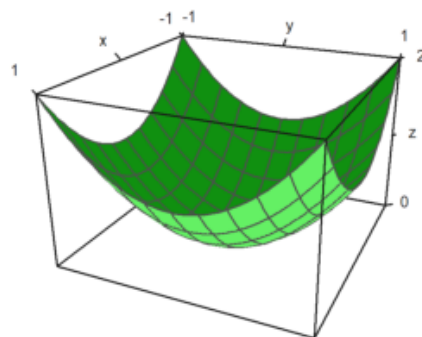
- ekspresi sederhana dalam x dan y ,
- nama fungsi dari dua variabel l
- atau matriks data.

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

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

Kami menggunakan default $n = 40$ dan $\text{grid} = 10$.

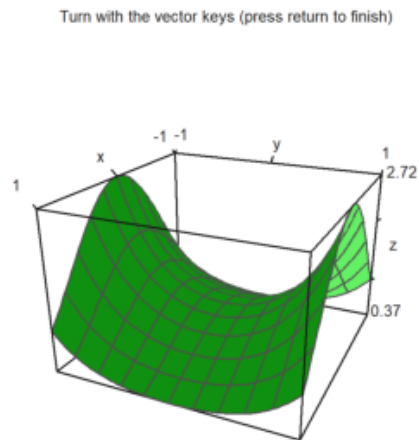
```
>plot3d("x^2+y^2") :
```



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

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

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



Rentang plot untuk fungsi dapat ditentukan dengan

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

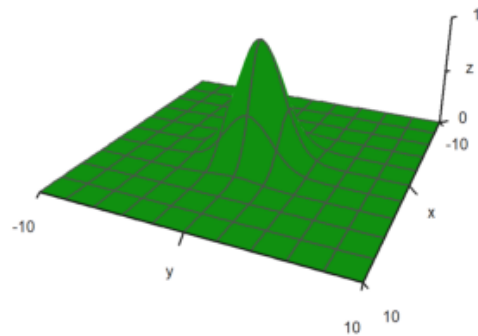
Ada beberapa parameter untuk menskalakan fungsi atau mengubah tampilan grafik.

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

scale: angka atau vektor 1x2 untuk skala 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):
```



Tampilan dapat diubah dengan berbagai cara.

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

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

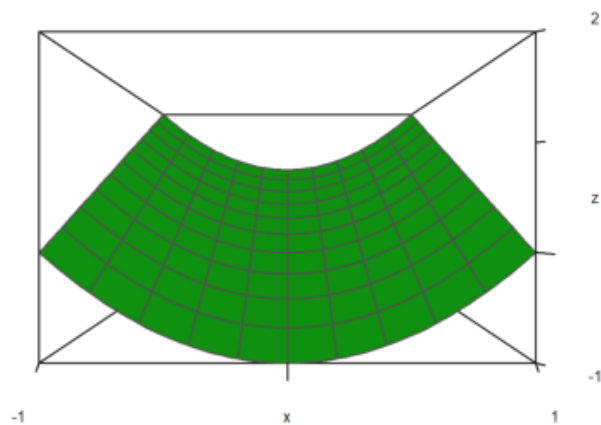
```
>view
```

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

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

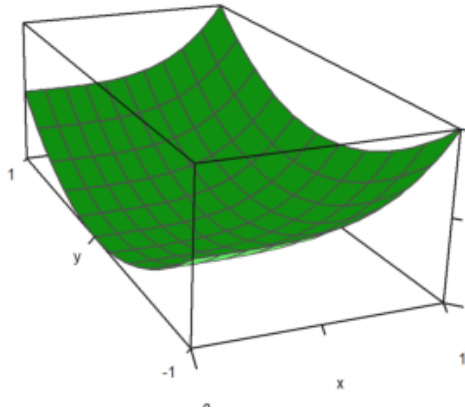
Dalam contoh berikut, sudut = 0 dan tinggi = 0 dilihat dari sumbu y negatif. Label sumbu untuk y disembunyikan dalam kasus ini.

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



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

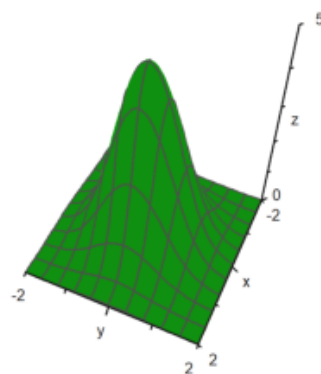
```
>plot3d("x^4+y^2",a=0,b=1,c=-1,d=1,angle=-20°,height=20°, ...  
> center=[0.4,0,0],zoom=5):
```



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

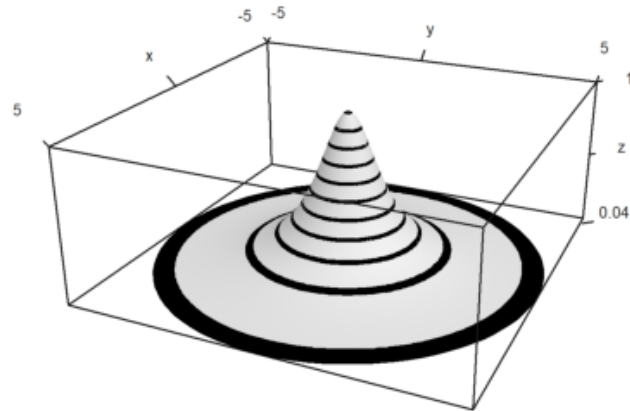
Jika Anda mematikannya dengan `scale = false`, Anda harus berhati-hati, bahwa plot masih pas dengan jendela plotting, dengan mengubah jarak pandang atau zoom, dan memindahkan pusat.

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

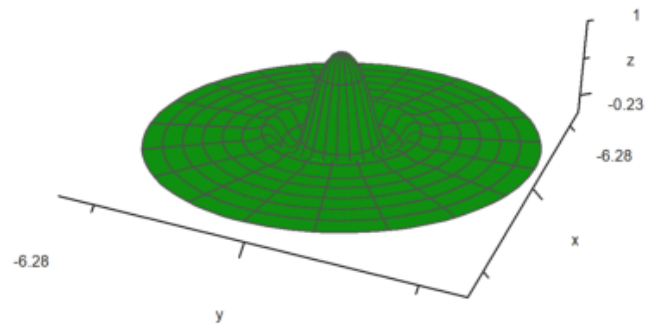


Plot kutub juga tersedia. The parameter `polar= true` menggambarkan plot kutub. Fungsi tersebut harus tetap merupakan fungsi dari x dan y . Parameter "`fscale`" menskalakan fungsi dengan skala sendiri. Jika tidak, fungsi diskalakan agar sesuai dengan kubus.

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



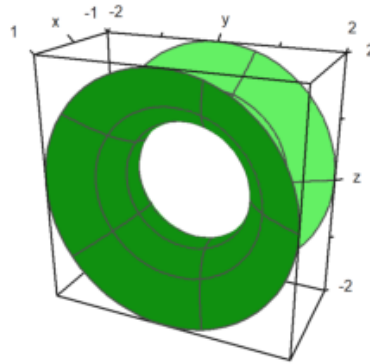
```
>function f(r) := exp(-r/2)*cos(r); ...
>plot3d("f(x^2+y^2)",>polar,scale=[1,1,0.4],r=2pi,frame=3,zoom=4):
```



Parameter memutar memutar fungsi dalam x di sekitar sumbu x.

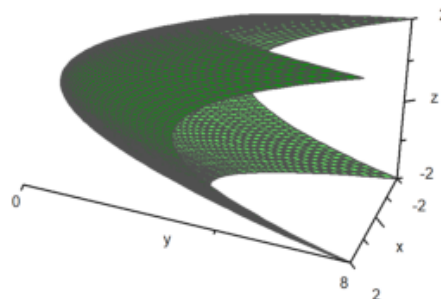
- rotate = 1: Menggunakan sumbu x
- rotate = 2: Menggunakan sumbu z

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



Berikut adalah plot dengan tiga fungsi.

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



Latihan Soal

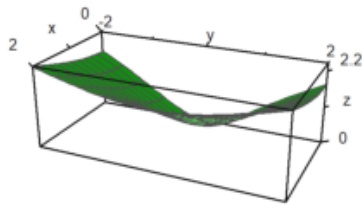
1. Plot fungsi

$$z = \ln(x^2 + y^2 + 1)$$

untuk rentang $0 < x < 2$ dan $-2 < y < 2$. Atur jumlah subinterval untuk plot menjadi 50.

JAWAB :

```
>aspect(1.5);
>plot3d("ln(x^2 + y^2 + 1)", a=0, b=2, c=-2, d=2, n=50, frame=1) :
```

2. Gambarlah permukaan dari fungsi

$$z = x^3 - 3xy^2$$

dalam rentang $-2 \leq x \leq 2$ dan $-2 \leq y \leq 2$. Gantilah sudut pandang untuk memberikan tampilan yang lebih baik.
JAWAB:

```
>aspect(1.5);
>plot3d("x^3 - 3*x*y^2", r=2, n=40);
```

Space between commands expected!

Found:

(character 10)

You can disable this in the Options menu.

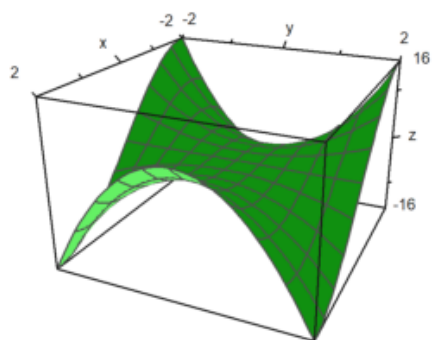
Error in:

```
plot3d("x^3 - 3*x*y^2", r=2, n=40);
```

...

^

```
>view(45, 30):
```



Plot Kontur

Untuk plot, Euler menambahkan garis kisi. Alih-alih, dimungkinkan untuk menggunakan garis level dan corak satu warna atau corak warna spektral. Euler dapat menggambar ketinggian fungsi pada plot dengan shading. Di semua plot 3D, Euler dapat menghasilkan anaglyph merah / cyan.

-> hue: Mengaktifkan bayangan terang, bukan kabel.

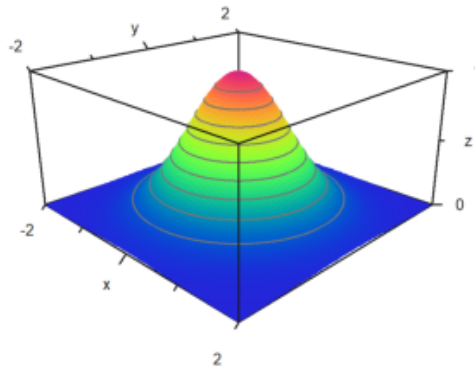
-> contour: Membuat plot garis kontur otomatis pada plot.

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

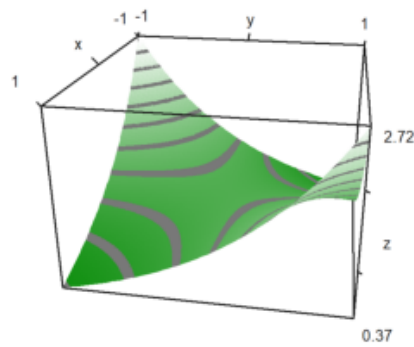
Standarnya adalah level = "auto", yang menghitung beberapa baris level secara otomatis. Seperti yang Anda lihat di plot, level sebenarnya adalah rentang level.

Gaya default dapat diubah. Untuk plot kontur berikut, kami menggunakan kisi yang lebih halus untuk 100x100 titik, menskalakan 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):
```



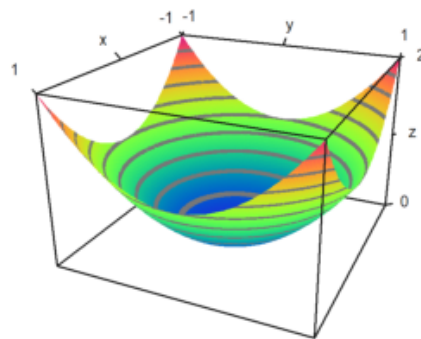
Bayangan default menggunakan warna abu-abu. Tetapi berbagai spektrum warna juga tersedia.

-> spectral: Menggunakan skema spektral default

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

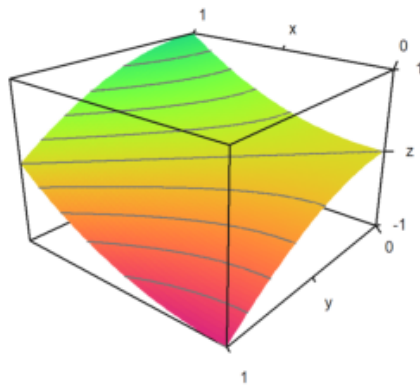
Untuk plot berikut, kami menggunakan skema spektral default dan menambah jumlah poin untuk mendapatkan tampilan yang sangat mulus.

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



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

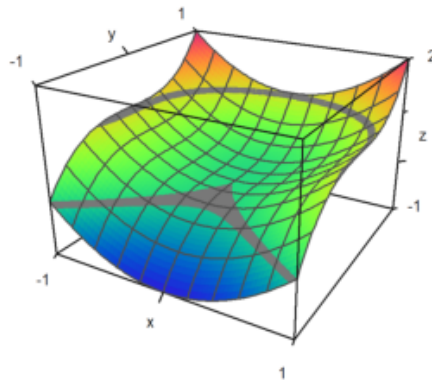
```
>plot3d("x^2-y^2",0,1,0,1,angle=220°,level=-1:0.2:1,color=redgreen):
```



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

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

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

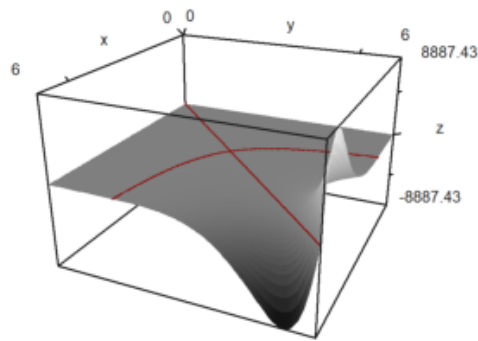


Dalam contoh berikut, kami memplot set, di mana

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

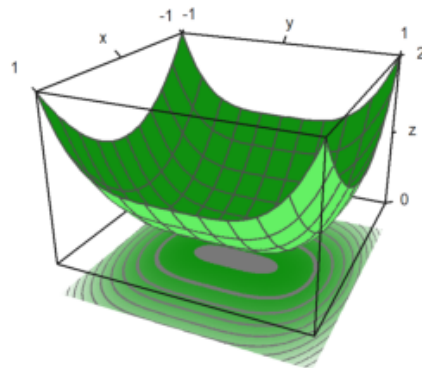
Kami menggunakan satu garis tipis untuk garis level.

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



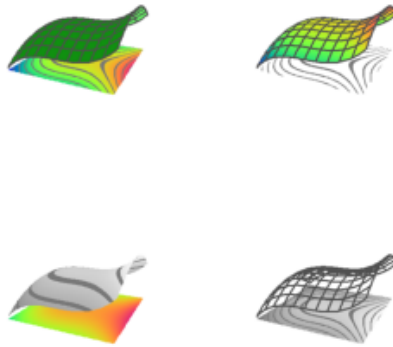
Dimungkinkan untuk menunjukkan bidang kontur di bawah plot. Warna dan jarak ke plot dapat ditentukan.

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



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

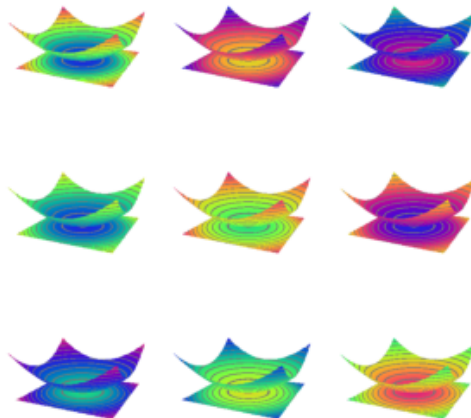
```
>figure(2,2); ...
>expr="y^3-x^2"; ...
>figure(1); ...
> plot3d(expr,<frame,>cp,cpcolor=spectral); ...
>figure(2); ...
> plot3d(expr,<frame,>spectral,grid=10,cp=2); ...
>figure(3); ...
> plot3d(expr,<frame,>contour,color=gray,nc=5,cp=3,cpcolor=greenred); ...
>figure(4); ...
> plot3d(expr,<frame,>hue,grid=10,>transparent,>cp,cpcolor=gray);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



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

- `spectral`: untuk rentang dari biru hingga merah
- `white`: untuk rentang yang lebih redup
- `yellowblue`, `purplegreen`, `blueyellow`, `greenred`
- `blueyellow`, `greenpurple`, `yellowblue`, `redgreen`

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

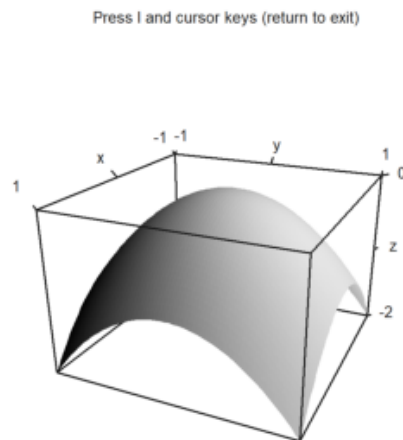


Sumber cahaya dapat diubah dengan `l` dan tombol kursor selama interaksi pengguna. Itu juga dapat diatur dengan parameter.

- `light` : arah cahaya
- `amb`: cahaya ambient antara 0 dan 1

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

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



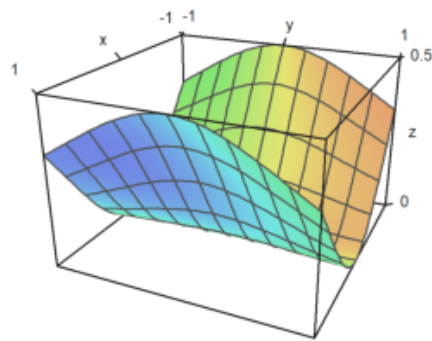
Parameter warna mengubah warna permukaan. Warna garis level juga dapat diubah.

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



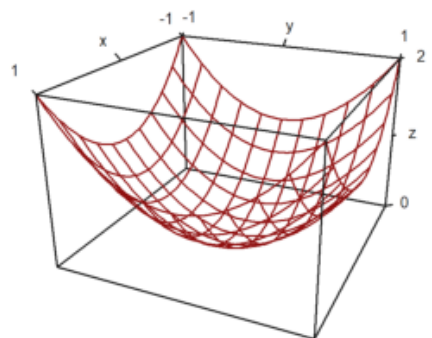
Warna 0 memberikan efek pelangi khusus.

```
>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):
```



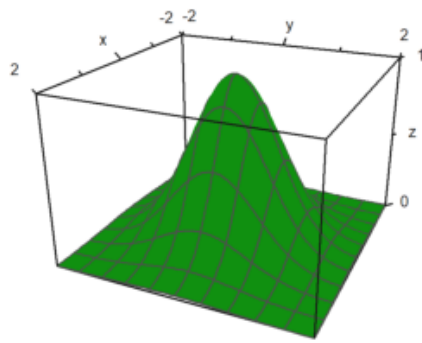
Latihan Soal

1. Plot fungsi

$$z = e^{(-x^2 - y^2)}$$

dengan garis kontur otomatis. Gunakan rentang $-2 \leq x \leq 2$ dan $-2 \leq y \leq 2$. Atur jumlah titik untuk mendapatkan tampilan yang halus dengan 100x100 titik.

```
>aspect(1.5);
>plot3d("exp(-x^2 - y^2)", r=2, n=100, <contour, <spectral):
```

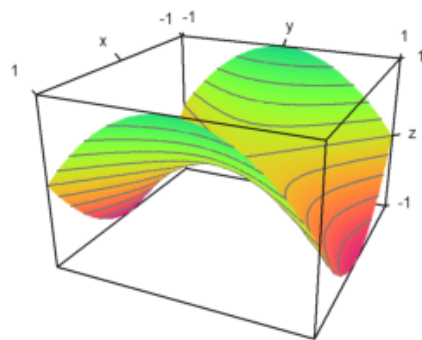
2. Gambarlah garis kontur dari fungsi

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

dengan level garis kontur yang ditentukan dari -1 hingga 1 dengan interval 0.2. Gunakan rentang $1 \leq x \leq 1$ dan $1 \leq y \leq 1$.

JAWAB :

```
>aspect(1.5);
>plot3d("x^2 - y^2", a=-1, b=1, c=-1, d=1, level=-1:0.2:1, <contour, color=redgreen):
```



Plot Implisit

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

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

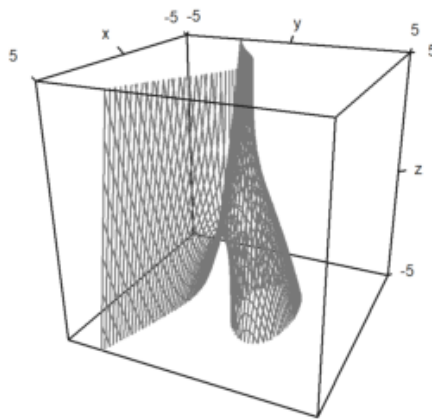
dapat divisualisasikan dalam potongan sejajar dengan bidang x-y, x-z, dan y-z.

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

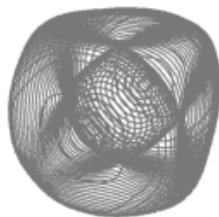
Tambahkan nilai-nilai ini, jika Anda suka. Dalam contoh yang kami plot

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

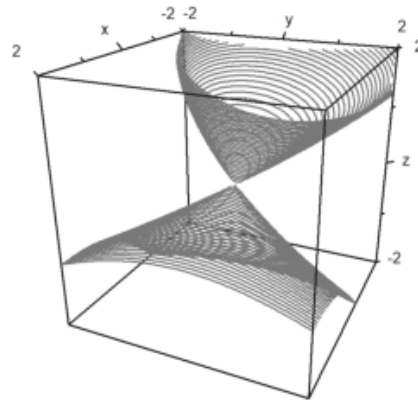
```
>plot3d("x^2+y^3+z*y-1",r=5,implicit=3):
```



```
>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)",r=5,implicit=3):
```



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



Soal Latihan

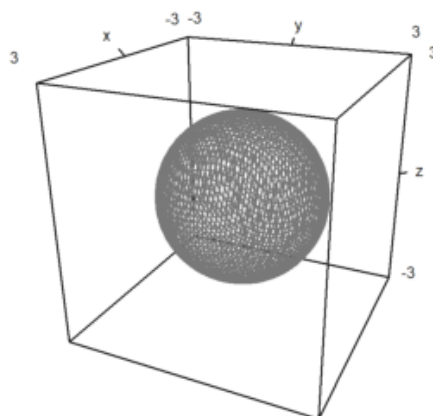
1. Plot himpunan nol dari fungsi

$$f(x, y, z) = x^2 + y^2 + z^2 - 4.$$

Gunakan potongan sejajar dengan bidang x-y dan x-z.

JAWAB :

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



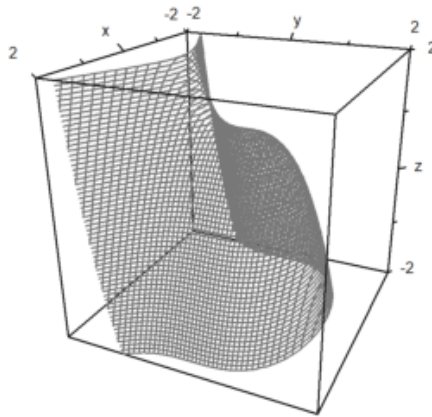
2. Gambarkanlah potongan dari himpunan nol fungsi

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

dengan potongan sejajar bidang y-z dan x-z.

JAWAB :

```
>plot3d("x^2 + y^3 + z - 1", r=2, implicit=6):
```



Plotting 3D Data

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

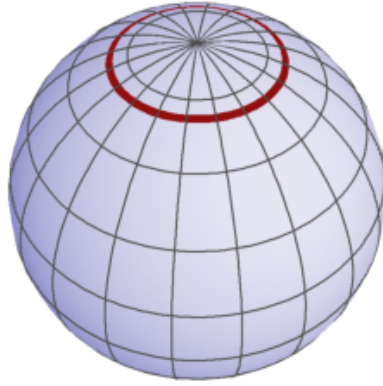
$$\gamma(t, s) = (x(t, s), y(t, s), z(t, s))$$

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

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

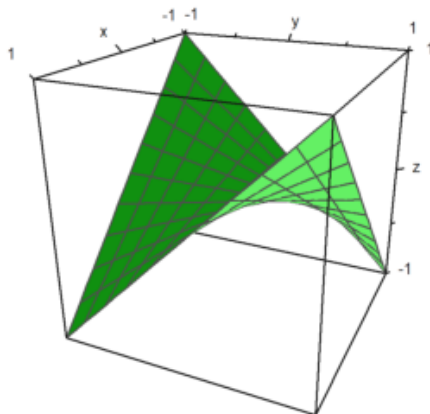
Dalam contoh berikut, kami menggunakan vektor nilai t dan vektor kolom nilai s untuk membuat parameter permukaan bola. Dalam gambar kita bisa menandai daerah, dalam kasus kita daerah kutub.

```
>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 adalah contoh grafik dari suatu fungsi.

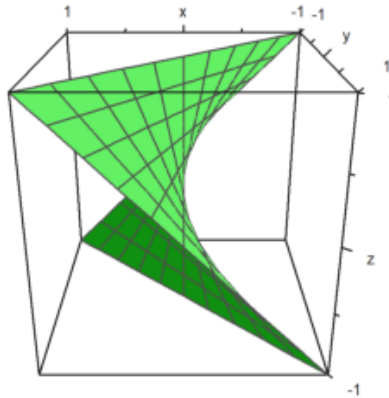
```
>t=-1:0.1:1; s=(-1:0.1:1)'; plot3d(t,s,t*s,grid=10):
```



Namun, kita bisa membuat semua jenis permukaan. Ini adalah permukaan yang sama sebagai suatu fungsi

$$x = yz$$

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



Dengan lebih banyak usaha, kami dapat menghasilkan banyak permukaan.

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

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

dengan

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

Kami menyimpangkan ini dengan sebuah faktor

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

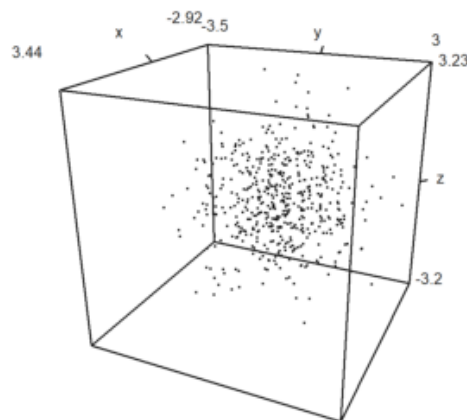
```
>t=linspace(0,2pi,320); s=linspace(-pi/2,pi/2,160)'; ...
>d=1+0.2*(cos(4*t)+cos(8*s)); ...
>plot3d(cos(t)*cos(s)*d,sin(t)*cos(s)*d,sin(s)*d,hue=1, ...
> light=[1,0,1],frame=0,zoom=5):
```



Tentu saja, point cloud juga dimungkinkan. Untuk memplot data titik dalam ruang, kita membutuhkan tiga vektor sebagai koordinat titik.

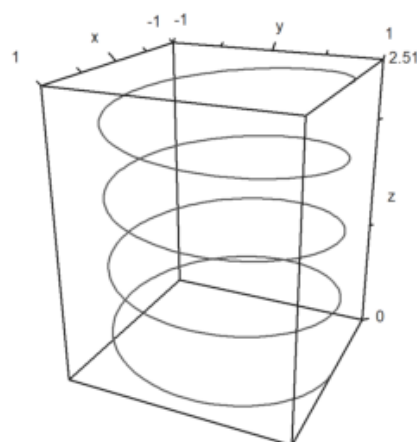
Gayanya sama seperti di plot2d dengan `points = true`;

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

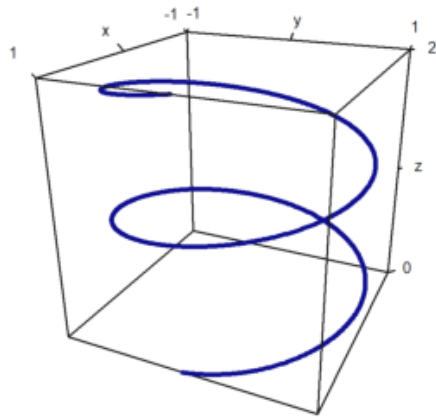


Juga dimungkinkan untuk memplot kurva dalam 3D. Dalam kasus ini, lebih mudah untuk menghitung sebelumnya titik-titik kurva. Untuk kurva di bidang kita menggunakan urutan koordinat dan parameter `wire = true`.

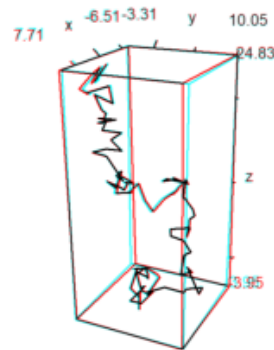
```
>t=linspace(0,8pi,500); ...
>plot3d(sin(t),cos(t),t/10,>wire,zoom=3) :
```



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

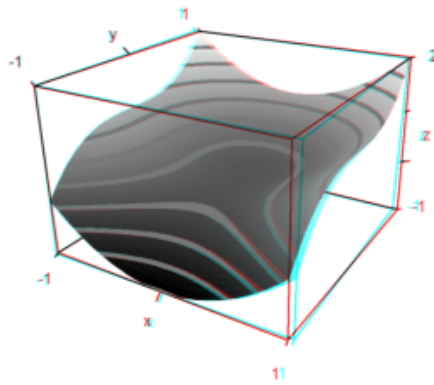


```
>X=cumsum(normal(3,100)); ...
> plot3d(X[1],X[2],X[3],>anaglyph,>wire):
```



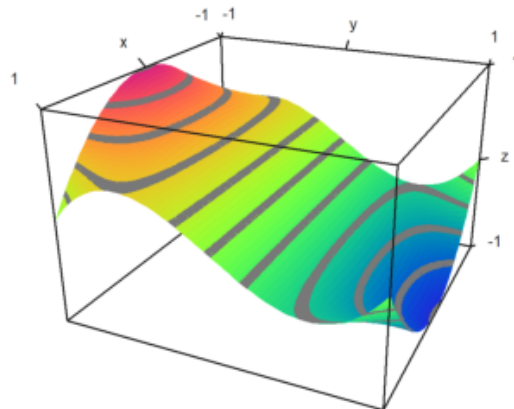
EMT juga dapat memplot dalam mode anaglyph. Untuk melihat plot seperti itu, Anda membutuhkan red/cyan glasses.

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

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

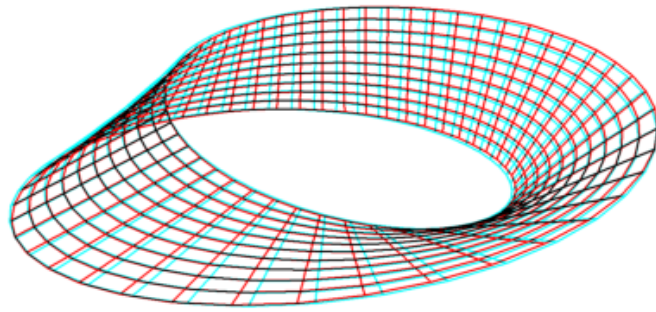
```
>plot3d("x^2*y^3-y",>spectral,>contour, zoom=3.2) :
```



Euler juga dapat memplot permukaan berparameter, jika parameternya adalah nilai x , y , dan z dari gambar kisi persegi panjang di dalam ruang.

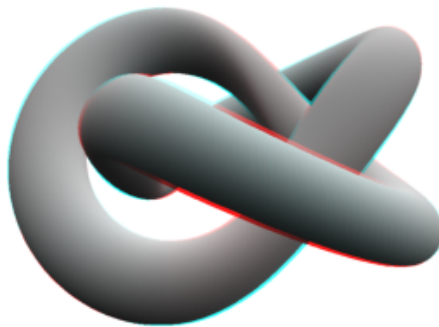
Untuk demo berikut, kami menyiapkan parameter u - dan v -, dan menghasilkan koordinat ruang dari ini.

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



Berikut adalah contoh yang lebih rumit, yang megah dengan red/cyan glasses.

```
>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):
```



Soal Latihan

1. Buatlah plot permukaan dari fungsi

$$z = \sin(x) * \cos(y)$$

untuk rentang $-\pi \leq x \leq \pi$ dan $-\pi \leq y \leq \pi$ dengan grid 50x50.

JAWAB :

```
>x := linspace(-pi, pi, 50);  
>y := linspace(-pi, pi, 50)';
```

Space between commands expected!

Found:

(character 10)

You can disable this in the Options menu.

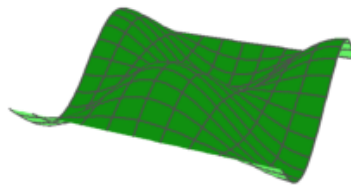
Error in:

```
y := linspace(-pi, pi, 50)';
```

...

^

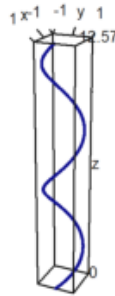
```
>z = sin(x) * cos(y);  
>plot3d(x, y, z, hue=0, frame=0, grid=[10, 20]):
```



2. Gambarkanlah plot untuk kurva 3D dengan parameter $x(t)=\cos(t)$, $y(t)=\sin(t)$, dan $z(t)=t$ dengan t dari 0 hingga 4π .

JAWAB :

```
>t = linspace(0, 4 * pi, 1000);  
>x = cos(t); y = sin(t); z = t;  
>plot3d(x, y, z, wire=true, linewidth=2, wirecolor=blue):
```



Plot Statistik

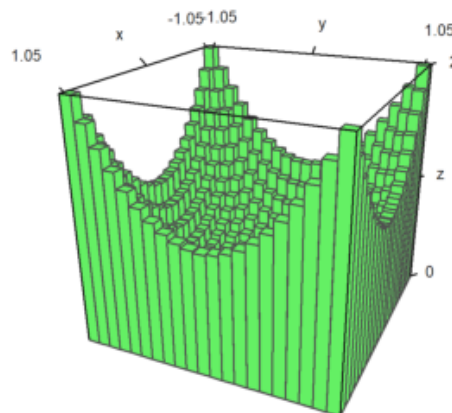
Plot batang juga dimungkinkan. Untuk ini, kami harus menyediakan

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

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

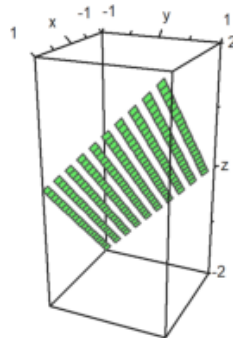
Dalam contoh, pertama-tama kita menghitung nilainya. Kemudian kita menyesuaikan x dan y, sehingga vektor berpusat pada nilai yang digunakan.

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



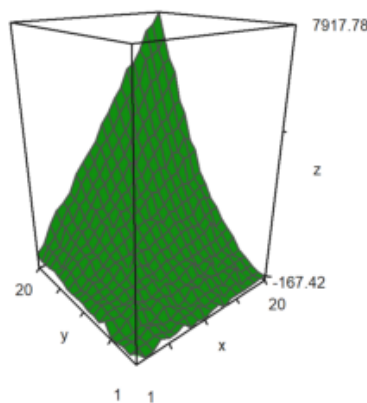
Dimungkinkan untuk membagi plot permukaan menjadi dua bagian atau lebih.

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

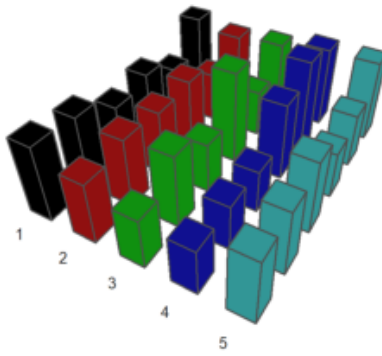


Jika memuat atau membuat matriks data M dari file dan perlu memplotnya dalam 3D, Anda dapat menskalakan matriks ke $[-1,1]$ dengan skala (M) , atau menskalakan matriks dengan `zscale`. Ini dapat dikombinasikan dengan faktor penskalaan individu yang diterapkan sebagai tambahan.

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



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



Latihan Soal

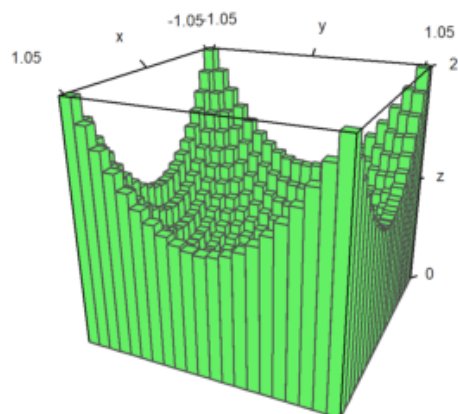
1. Gambarlah plot batang 3D dari fungsi

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

untuk x dan y dalam rentang dari -1 hingga 1 dengan interval 0.1.

JAWAB :

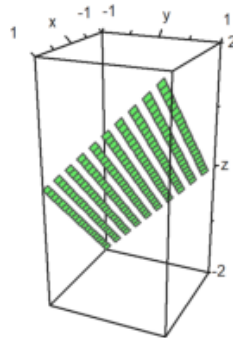
```
>x = -1:0.1:1; y = x'; z = x^2 + y^2;
>xa = (x|1.1) - 0.05; ya = (y_1.1) - 0.05;
>plot3d(xa, ya, z, bar=true):
```



2. Buatlah plot permukaan dari fungsi $Z=x+y$ dan pisahkan plot menjadi dua bagian untuk x dan y dalam rentang dari -1 hingga 1.

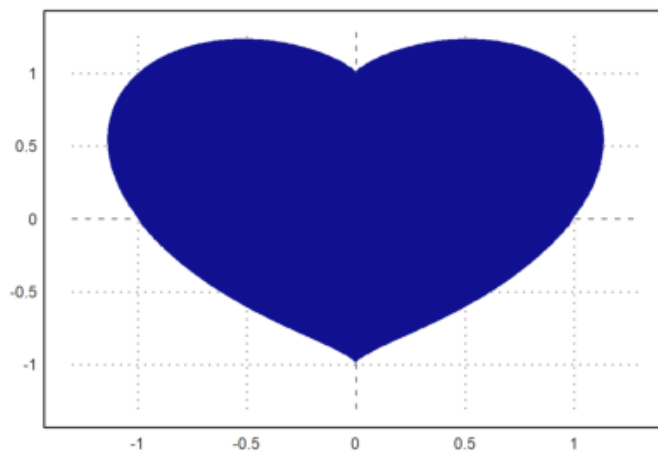
JAWAB :

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



Permukaan Benda Putar

```
>plot2d("(x^2+y^2-1)^3-x^2*y^3",r=1.3, ...  
>style="#",color=blue,<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. Inilah ungkapan yang mendefinisikan hati:

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

Selanjutnya kita atur

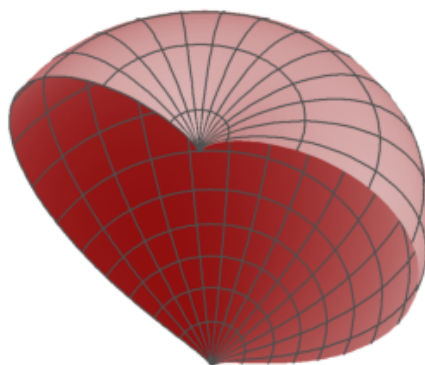
$$x = r \cdot \cos(a), \quad y = r \cdot \sin(a).$$

```
>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 menentukan fungsi numerik, yang menyelesaikan r , jika a diberikan. Dengan fungsi itu kita dapat memplot jantung yang berubah sebagai permukaan parametrik.

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

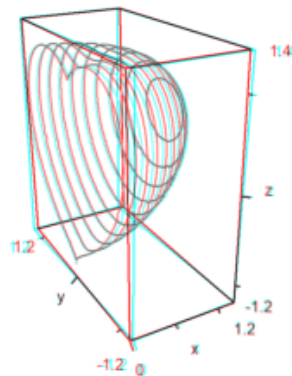


Berikut ini adalah plot 3D dari gambar di atas yang diputar di sekitar sumbu z. Kami mendefinisikan fungsi, yang mendeskripsikan objek.

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

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

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



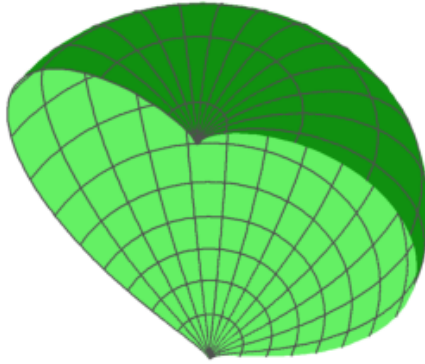
Latihan Soal

1. Gambarlah permukaan benda putar dari kurva

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

dengan memutar kurva ini di sekitar sumbu y. Gunakan parameter r dan a untuk mendefinisikan permukaan.
JAWAB:

```
>ekspresi &= (x^2 + y^2 - 1)^3 - x^2 * y^3;
>function fr(r, a) &= ekspresi with [x = r * cos(a), y = r * sin(a)] | trigreduce;
>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=green, zoom=4, amb=0, max=0.7, grid=12, height=50°):
```



2. Buatlah plot 3D dari objek yang dihasilkan dengan memutar kurva jantung di sekitar sumbu z. Definisikan fungsi

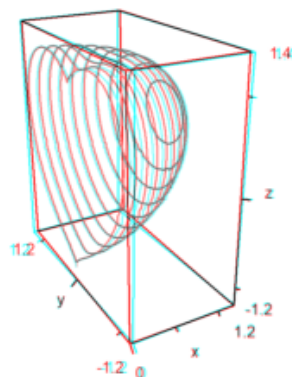
$$f(x, y, z) \text{ sebagai } (x^2 + y^2 + z^2 - 1)^3 - r * z^3.$$

JAWAB:

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

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

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



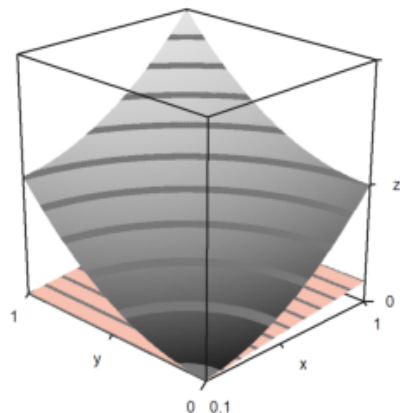
Fungsi `plot3d` bagus untuk dimiliki, tetapi tidak memenuhi semua kebutuhan. Selain rutinitas yang lebih mendasar, Anda bisa mendapatkan plot berbingkai dari objek apa pun yang Anda suka.

Meskipun Euler bukan program 3D, Euler dapat menggabungkan beberapa objek dasar. Kami mencoba untuk memvisualisasikan paraboloid dan garis singgung-nya.

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

Sekarang `framedplot()` menyediakan bingkai, dan menyetel tampilan.

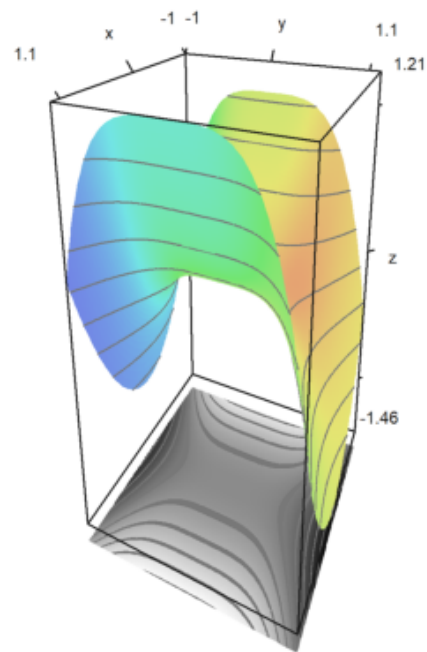
```
>framedplot("myplot",[0.1,1,0,1,0,1],angle=-45°, ...  
> center=[0,0,-0.7],zoom=6):
```



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



Latihan Soal

1. Buatlah plot 3D dari fungsi

$$z = x^2 + y^2 (\text{paraboloid})$$

dan tambahkan garis singgung pada titik (1,1) dengan persamaan garis singgung $z=2(x-1)+2(y-1)+2$.

JAWAB:

```
>function myplot ...
```

```
    y = 0:0.01:1;  
    x = (0.1:0.01:1)'; ...  
endfunction
```

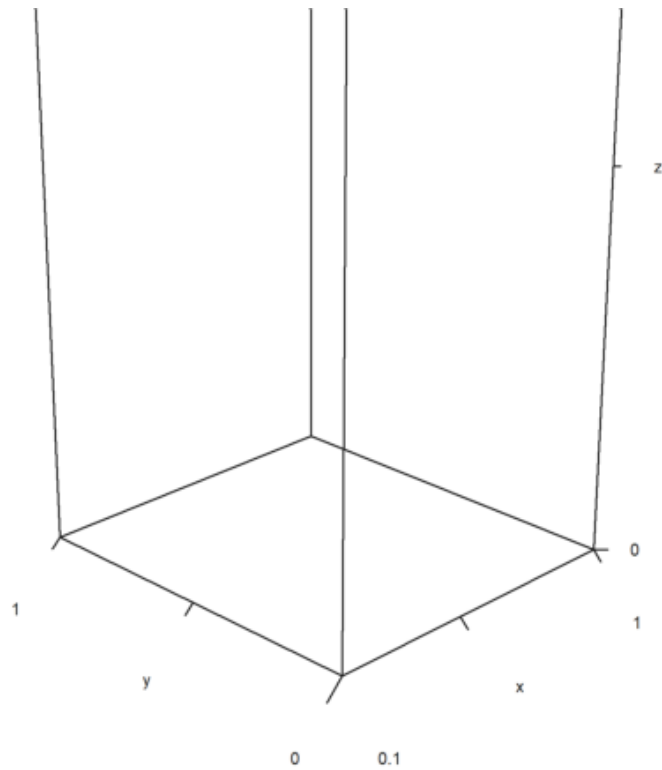
```
>plot3d(x, y, 2*(x - 1) + 2*(y - 1) + 2, <scale, <frame, >hue, hues=0.5, >contour, color=...
```



```
>h = holding(1);  
>plot3d(x, y, x^2 + y^2, <scale, <frame, >contour, >hue):
```



```
>holding(h);
>framedplot("myplot", [0.1, 1, 0, 1, 0, 2], angle=-45°, center=[0, 0, -0.7], zoom=6):
```



2. Buatlah plot bidang kontur dari fungsi

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

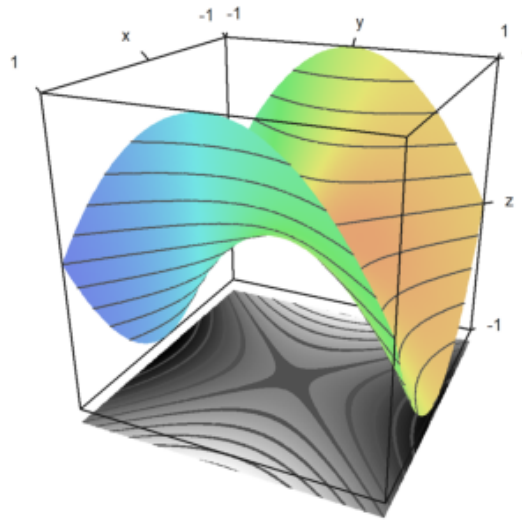
dalam rentang $-1 \leq x \leq 1$ dan $-1 \leq y \leq 1$. Tambahkan juga plot permukaan pada bidang kontur tersebut.

JAWAB :

```
>x = -1:0.02:1; y = x';
>z = x^2 - y^2;
>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):
```

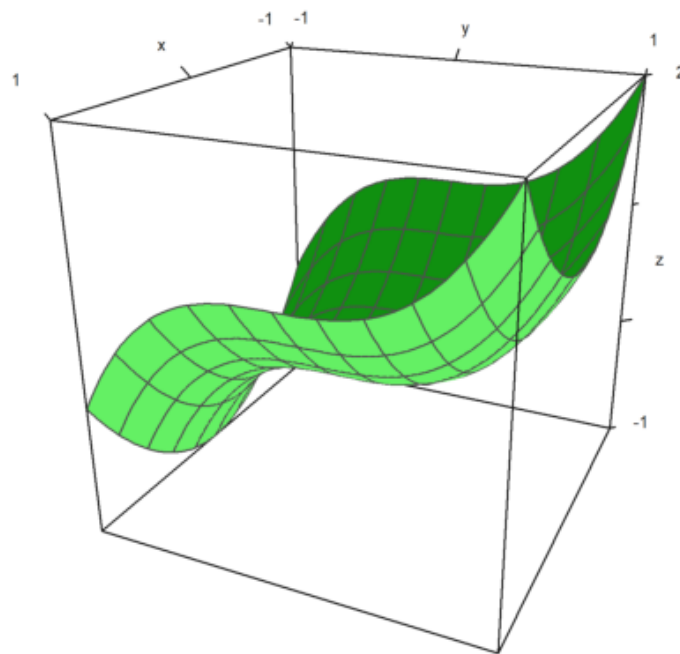


Animasi

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

Harap pelajari sumber rotasi untuk melihat lebih detail.

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



Latihan Soal

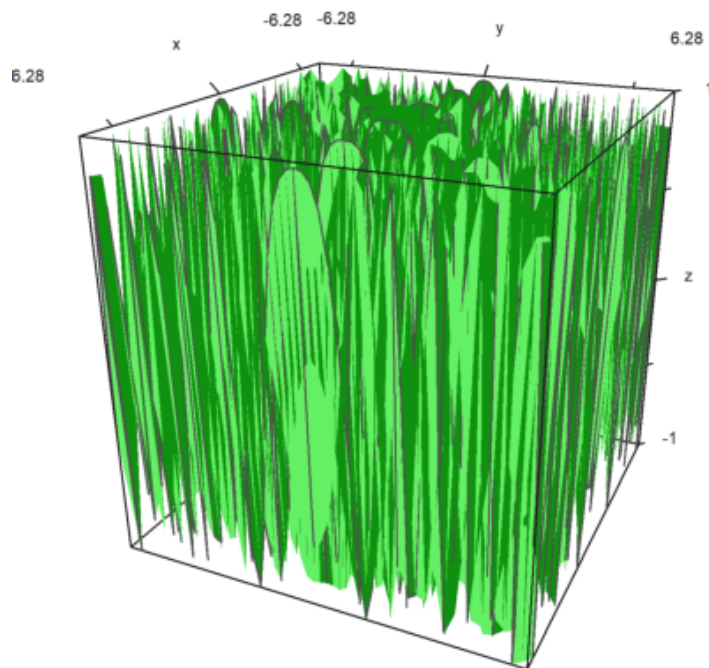
1. Buatlah animasi 3D dari fungsi

$$z = \sin(x^2 + y^2)$$

yang berputar di sekitar sumbu z. Tampilkan rotasi ini dengan menggunakan fungsi rotate.

JAWAB :

```
>function testplot() := plot3d("sin(x^2 + y^2)", xmin=-2*pi, xmax=2*pi, ymin=-2*pi, ymax=2*pi)
>rotate("testplot");
>testplot() :
```

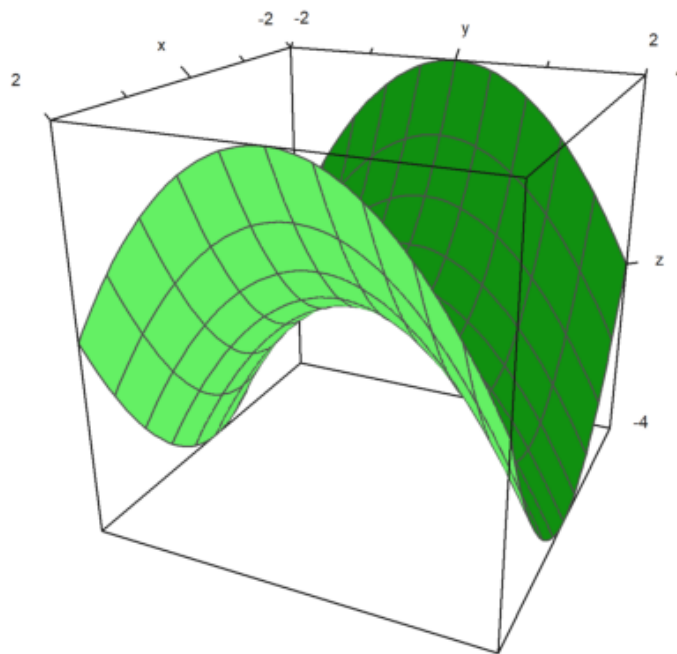
2. Buatlah animasi rotasi untuk permukaan

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

dengan batas $-2 \leq x, y \leq 2$. Tampilkan hasilnya dalam animasi.

JAWAB :

```
>function testplot() := plot3d("x^2 - y^2", xmin=-2, xmax=2, ymin=-2, ymax=2, zmin=-4, zmax=4)
>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" Povray ke dalam jalur lingkungan, atau menyetel 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 default adalah eulerhome (), biasanya c: \ Users \ Username \ Euler. Povray menghasilkan file PNG, yang dapat dimuat oleh Euler ke dalam notebook. Untuk membersihkan file-file ini, gunakan povclear ().

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

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

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

Perhatikan bahwa alam semesta Povray memiliki sistem koordinat lain. Antarmuka ini menerjemahkan semua koordinat ke sistem Povray. Jadi Anda dapat terus berpikir dalam sistem koordinat Euler dengan z menunjuk

ke atas secara vertikal, sumbu x , y , z di tangan kanan. Anda perlu memuat file povray.

```
>load povray;
```

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

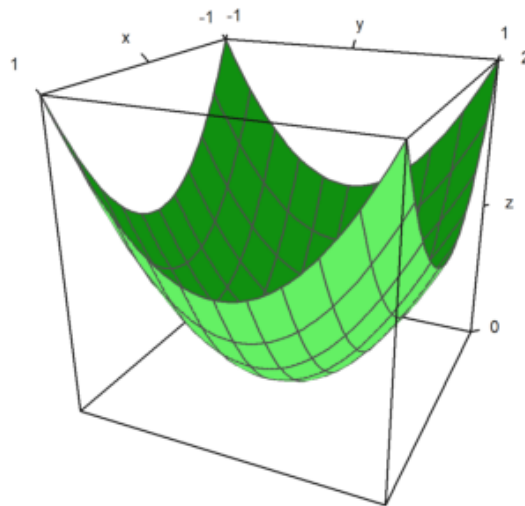
```
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

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

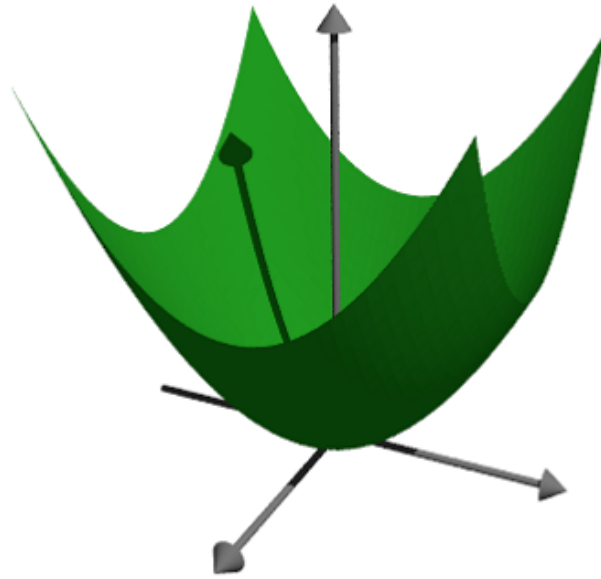
Untuk pertamakali, kami memplot fungsi sederhana. Perintah berikut menghasilkan file povray di direktori pengguna Anda, dan menjalankan Povray untuk menelusuri file ini.

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 batal untuk menghentikan pertanyaan lebih lanjut. Anda mungkin harus menekan OK di jendela Povray untuk mengetahui dialog start-up Povray.

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

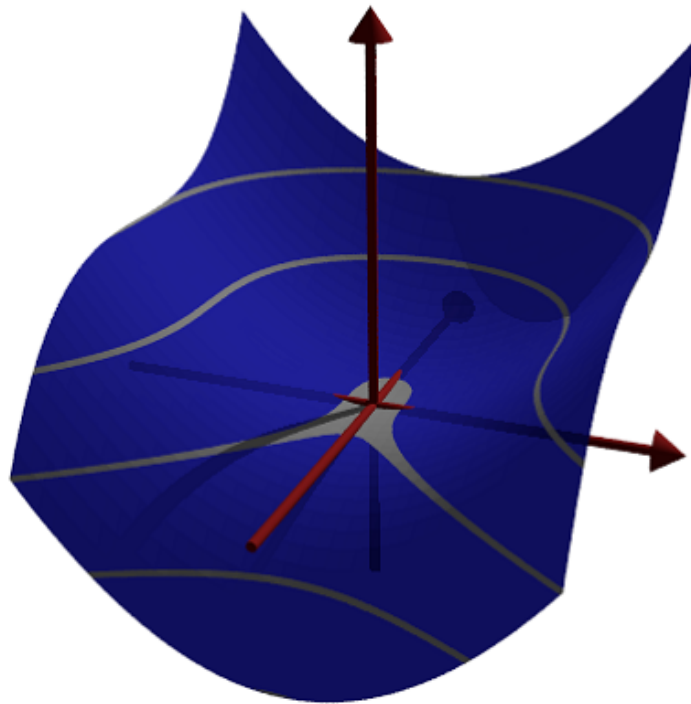


```
>pov3d("x^2+y^2",zoom=3);
```



Kita bisa membuat fungsinya transparan dan menambahkan hasil akhir lainnya. Kita juga bisa menambahkan garis level ke plot fungsi.

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



Terkadang perlu untuk mencegah penskalaan fungsi, dan menskalakan fungsi dengan tangan.
Kami memplot himpunan titik di bidang kompleks, di mana hasil kali jarak ke 1 dan -1 sama dengan 1.

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



Latihan Soal

1. Plotlah fungsi

$$z = \sin(x) * \cos(y)$$

dalam ruang 3D. Tambahkan garis level pada plot dengan rentang level dari -1 hingga 1
JAWAB :

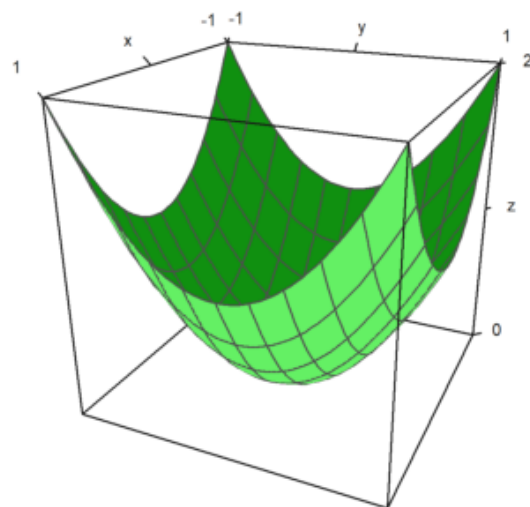
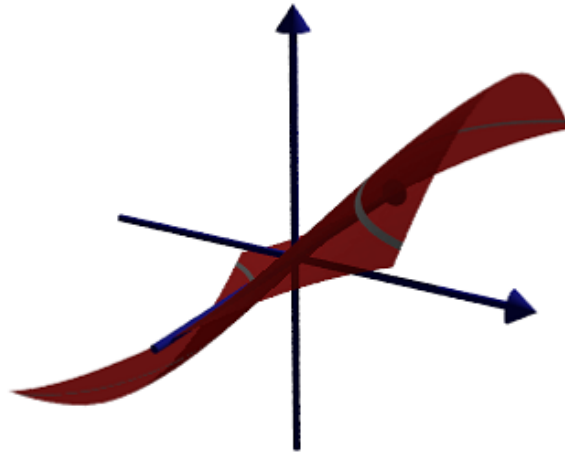
```
>f(x, y) := sin(x) * cos(y);
```

```
Variable or function x not found.
```

```
Error in:
```

```
f(x, y) := sin(x) * cos(y); ...
      ^
```

```
>pov3d("sin(x)*cos(y)", axiscolor=blue, angle=30°, ...
>look=povlook(red, 0.3), level=-1:0.5:1, zoom=3):
```



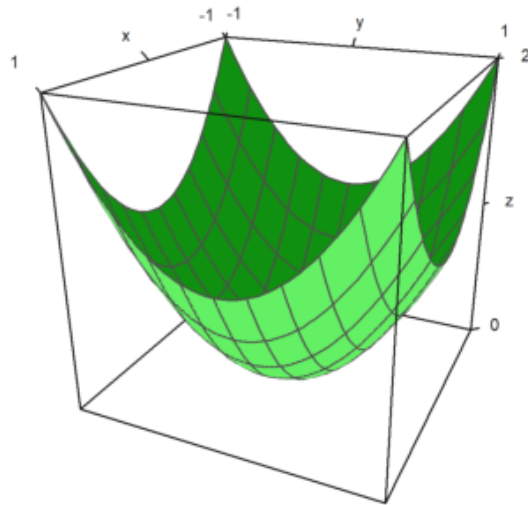
2. Buatlah permukaan dari fungsi

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

dan tambahkan vektor normal pada setiap titik permukaan. Gunakan rentang x dan y dari -2 hingga 2.
JAWAB :

```
>Z &= x^2 + y^2;  
>dx &= diff([x, y, Z], x);  
>dy &= diff([x, y, Z], y);  
>N &= crossproduct(dx, dy);  
>NX &= N[1]; NY &= N[2]; NZ &= N[3];  
>x = -2:0.5:2;  
>y = x';  
>pov3d(x, y, Z(x, y), angle=15°, ...  
>xv=NX(x, y), yv=NY(x, y), zv=NZ(x, y), <shadow):
```





Merencanakan dengan Koordinat

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

Dalam contoh ini, kita memutar fungsi di sekitar sumbu z .

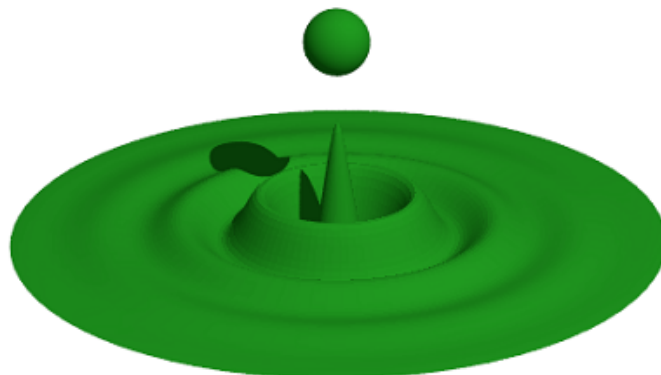
```
>function f(x) := x^3-x+1; ...
>x=-1:0.01:1; t=linspace(0,2pi,8)'; ...
>Z=x; X=cos(t)*f(x); Y=sin(t)*f(x); ...
>pov3d(X,Y,Z,angle=40°,height=20°,axis=0,zoom=4,light=[10,-5,5]);
```



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

Kami juga menunjukkan, bagaimana objek tambahan dapat ditambahkan ke adegan pov3d. Untuk pembuatan objek, lihat contoh berikut. Perhatikan bahwa plot3d menskalakan plot, sehingga 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=5,axis=0,add=povsphere([0,0,0.5],0.1,povlook(green)), ...
> w=500,h=300);
```



Dengan metode naungan lanjutan Povray, sangat sedikit titik yang dapat menghasilkan permukaan yang sangat halus. Hanya di perbatasan dan dalam bayangan, triknya mungkin menjadi jelas.

Untuk ini, 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 menjadi x dan y dari ini dan mengambil produk silang sebagai normal.

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

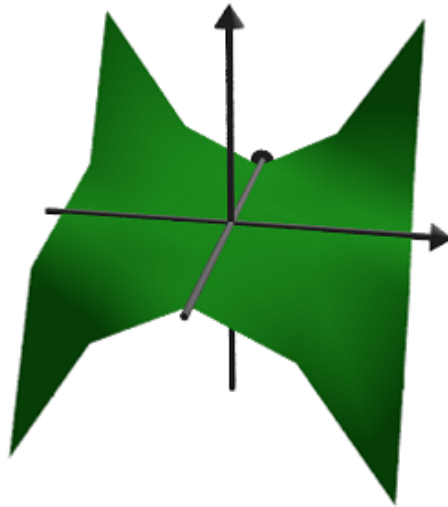
Kami mendefinisikan normal sebagai produk silang dari turunan ini, dan mendefinisikan fungsi koordinat.

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

$$\begin{matrix} 3 & 2 & 2 \\ [-2x^3y, -3x^2y^2, 1] \end{matrix}$$

We use only 25 points.

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



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

See: Examples\Trefoil Knot | Trefoil Knot

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

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

Kemudian dua vektor turunannya menjadi x dan y.

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

Sekarang normal, yang merupakan produk persilangan dari dua turunannya.

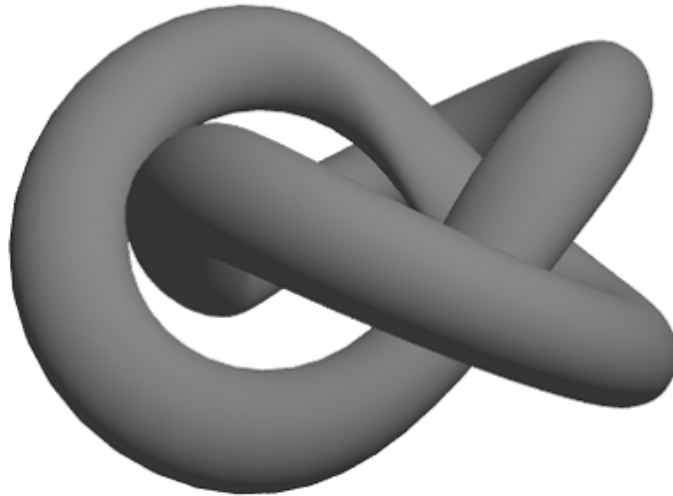
```
>dn &= crossproduct(dx,dy);
```

Kami sekarang mengevaluasi semua ini secara numerik.

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

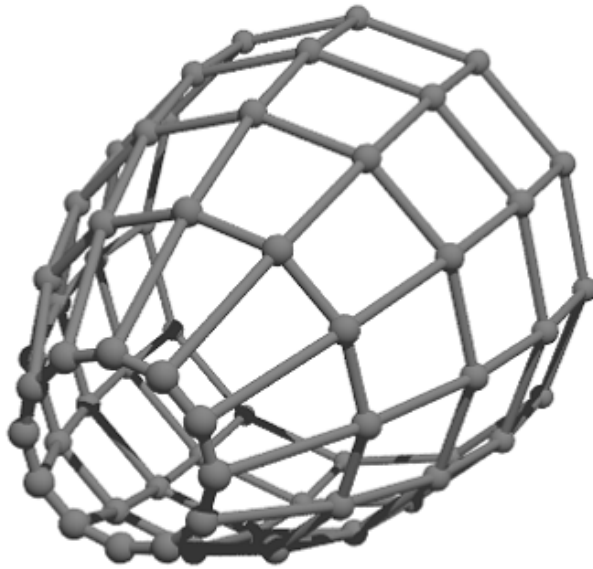
Vektor normal adalah evaluasi dari ekspresi simbolik $dn[i]$ untuk $i = 1, 2, 3$. Sintaks untuk ini adalah `&"expresi"` (parameter). Ini adalah alternatif metode pada contoh sebelumnya, di mana kita mendefinisikan ekspresi simbolik NX, NY, NZ terlebih dahulu.

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



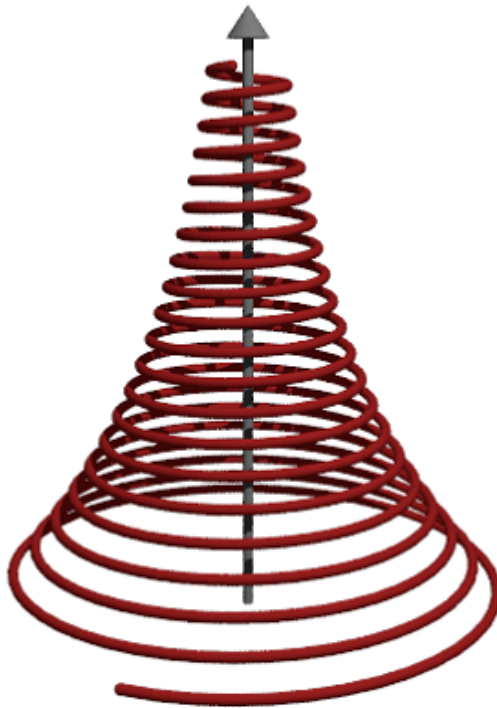
Kami juga dapat membuat grid dalam 3D.

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



Dengan `povgrid ()`, kurva dimungkinkan.

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



Latihan Soal

1. Plotlah fungsi

$$f(x) = x^3 - x + 1$$

yang diputar di sekitar sumbu z. Tampilkan hasilnya dengan sudut pandang tertentu.

JAWAB :

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



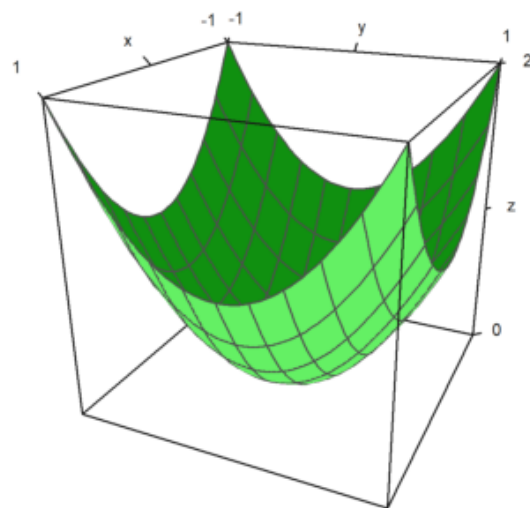
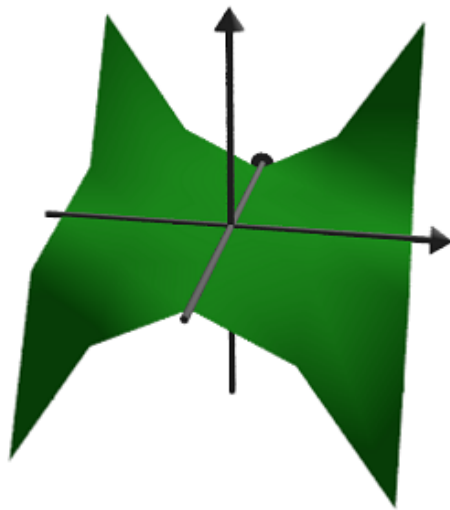
2. Plotlah fungsi

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

di dalam ruang 3D. Hitunglah normal di setiap titik permukaan dan tampilkan dengan vektor normal pada plot.

JAWAB :

```
>Z &= x^2 * y^3;
>dx &= diff([x, y, Z], x);
>dy &= diff([x, y, Z], y);
>N &= crossproduct(dx, dy);
>NX &= N[1]; NY &= N[2]; NZ &= N[3]; N;
>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):
```

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

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

Pertama kita tentukan tiga silinder, dan simpan dalam string di Euler.

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

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

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

```
>c2
```

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

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

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

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

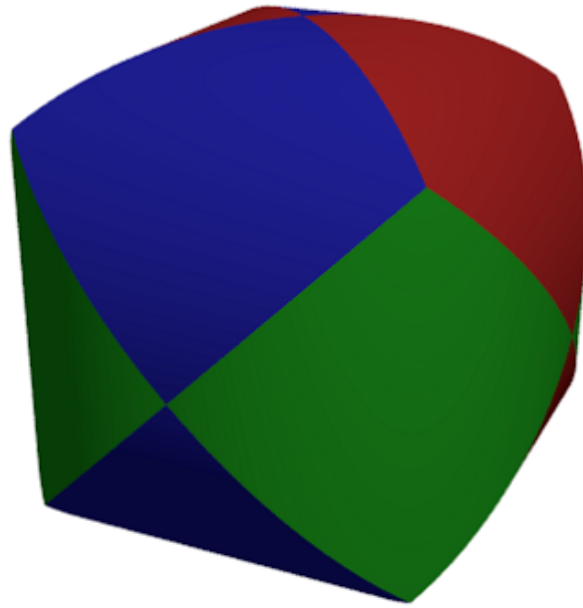
```
texture { pigment { color rgbf <0.101961,0.2,0.301961,0.1> } }  
finish { ambient 0.5 }
```

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

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

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

```
>povend;
```



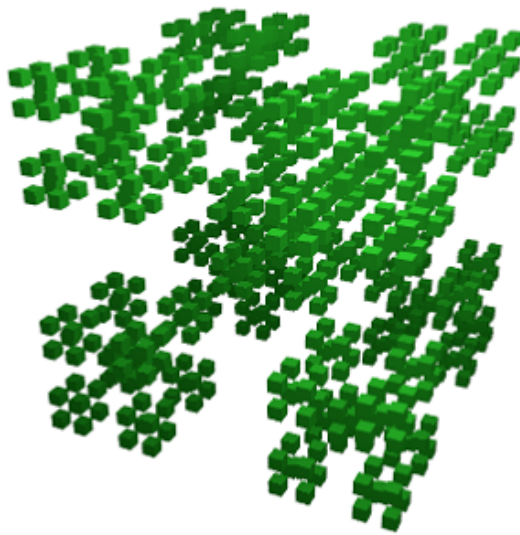
Fungsi berikut menghasilkan fraktal secara rekursif.

Fungsi pertama menunjukkan, bagaimana Euler menangani objek Povray sederhana. Fungsi `povbox()` mengembalikan string, 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 yang lain. Seperti persimpangan, ada bagian dari objek CSG Povray.

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

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

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

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

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

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

Kami menghasilkan kubus kedua, dan memutar serta menskalakannya sedikit.

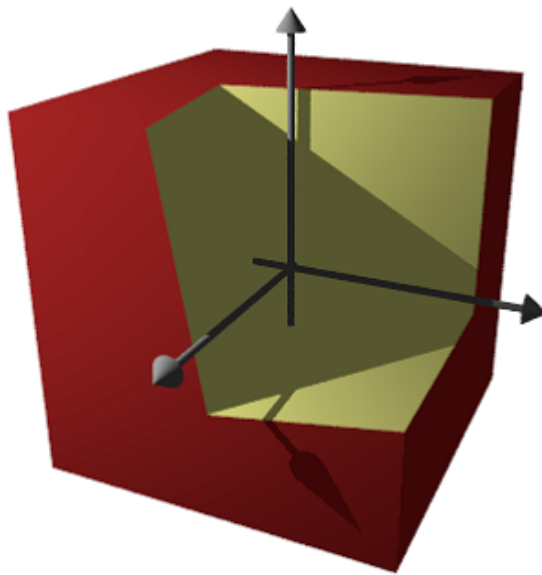
```
>c2=povobject ("mycube",povlook (yellow),translate=[1,1,1], ...  
> rotate=xrotate (10°)+yrotate (10°), scale=1.2);
```

Kemudian kita ambil perbedaan 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();
```



Latihan Soal

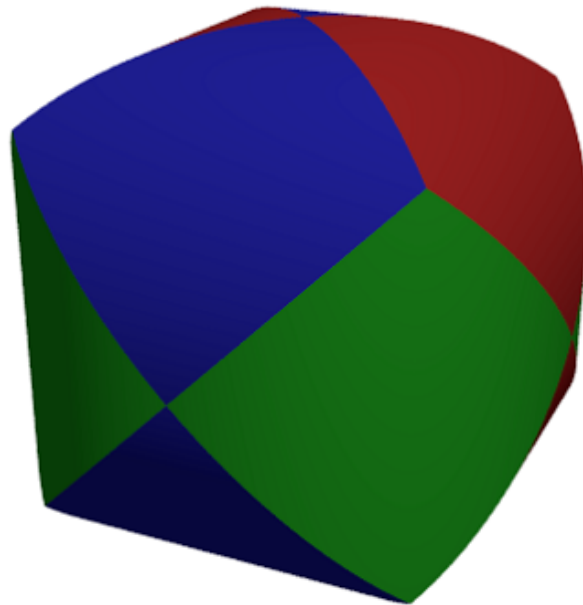
1. Buatlah tiga silinder dengan warna merah, hijau, dan biru yang saling berpotongan. Tampilkan objek-objek ini dan simpan hasilnya ke dalam file Povray.

JAWAB :

```

>povstart (zoom=4);
>c1 = povcylinder(-povx, povx, 1, povlook(red));
>c2 = povcylinder(-povy, povy, 1, povlook(green));
>c3 = povcylinder(-povz, povz, 1, povlook(blue));
>writeln(povintersection([c1, c2, c3]));
>povend();

```



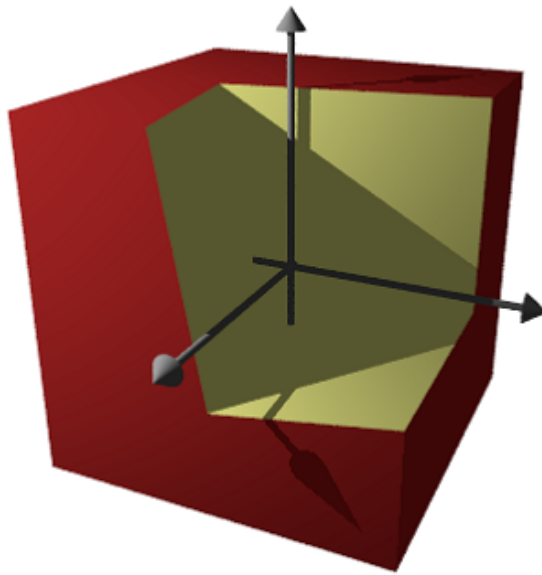
2. Buatlah dua kubus yang saling berpotongan, satu kubus berwarna merah dan yang lainnya berwarna kuning. Kubus kedua harus sedikit dipindahkan dan diputar. Tampilkan hasil potongan objek tersebut.

JAWAB:

```

>povstart (light=[5,-5,5], fade=10);
>povdefine("mycube", povbox(-1, 1));
>c1 = povobject("mycube", povlook(red));
>c2 = povobject("mycube", povlook(yellow), translate=[1, 1, 1], ...
> rotate=xrotate(10°) + yrotate(10°), scale=1.2);
>writeln(povdifference(c1, c2));
>writeAxis(-1.2, 1.2, axis=1);
>writeAxis(-1.2, 1.2, axis=2);
>writeAxis(-1.2, 1.2, axis=4);
>povend();

```



Fungsi Implisit

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

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

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

```
>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) +p
>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);
```

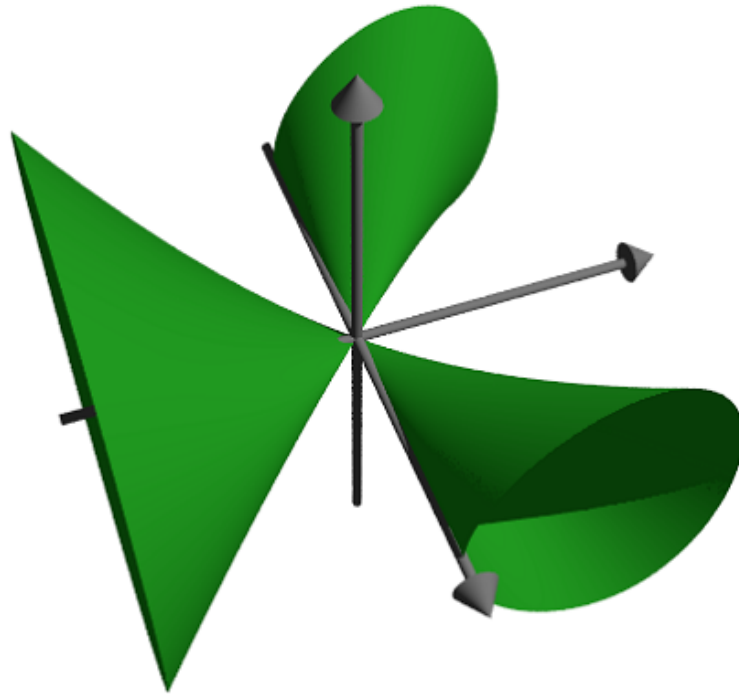
```
>povstart (angle=25°,height=10°);
>writeln(povsurface("pow(x,2)+pow(y,2)*pow(z,2)-1",povlook(blue),povbox(-2,2,"")));
>povend();
```



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

Buat permukaan implisit. Perhatikan sintaks yang berbeda dalam ekspresi tersebut.

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

Latihan Soal

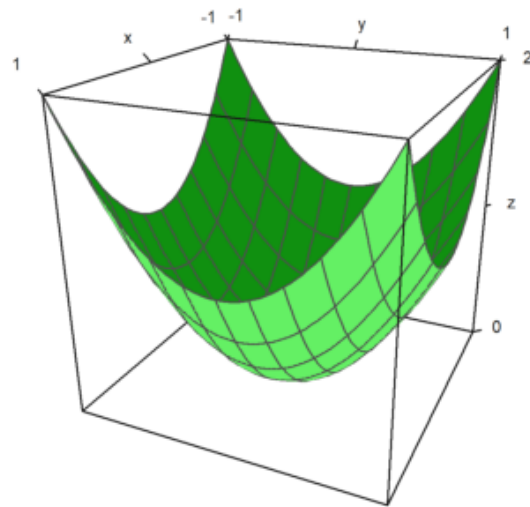
1. Buatlah permukaan implisit untuk persamaan

$$(x^2 + y^2 + z^2 - 1) * (x^2 + y^2 - z^2) = 0.$$

Tampilkan permukaan ini dengan warna merah.

JAWAB :

```
>povstart(angle=60°, height=50°, zoom=5);
>writeln(povsurface("(pow(x,2) + pow(y,2) + pow(z,2) - 1) * (pow(x,2) + pow(y,2) - pow(z,2) = 0);
>writeAxes();
>writeAxes():
```



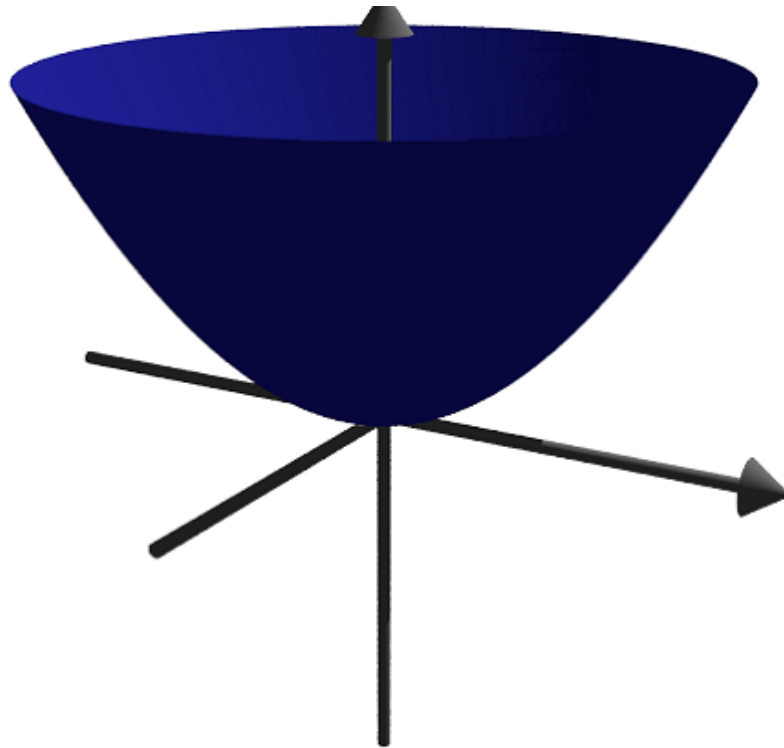
2. Buatlah permukaan implisit untuk persamaan

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

yang merepresentasikan paraboloid. Tampilkan permukaan ini dengan warna biru dan tambahkan sumbu untuk referensi.

JAWAB:

```
>povstart (angle=30°, height=20°, zoom=5);
>writeln(povsurface("z - pow(x,2) - pow(y,2)", povlook(blue)));
>writeAxes();
>povend();
```



Objek Jaring

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

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

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

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

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

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

Sekarang kami mendefinisikan dua disk, yang akan berpotongan dengan permukaan.

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

Tulis permukaan dikurangi dua cakram.

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

Tuliskan dua persimpangan tersebut.

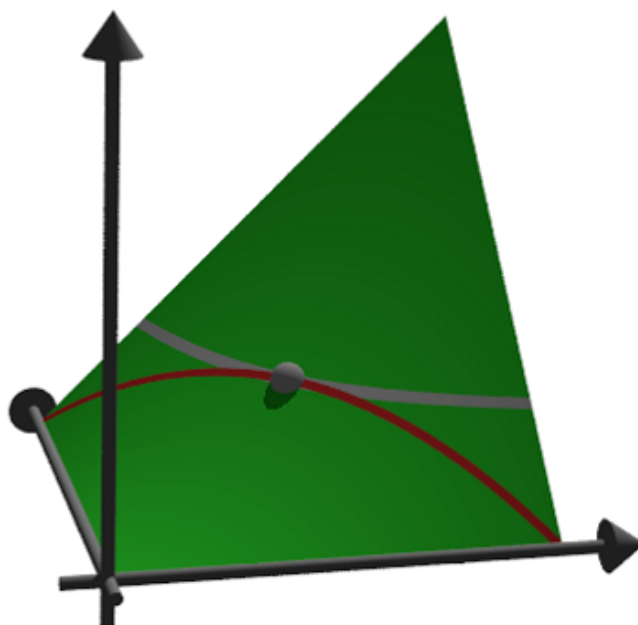
```
>writeln(povintersection([mesh,c1],povlook(red)); ...  
>writeln(povintersection([mesh,l1],povlook(gray)));
```

Tulis titik maksimal.

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

Tambahkan sumbu dan selesai.

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



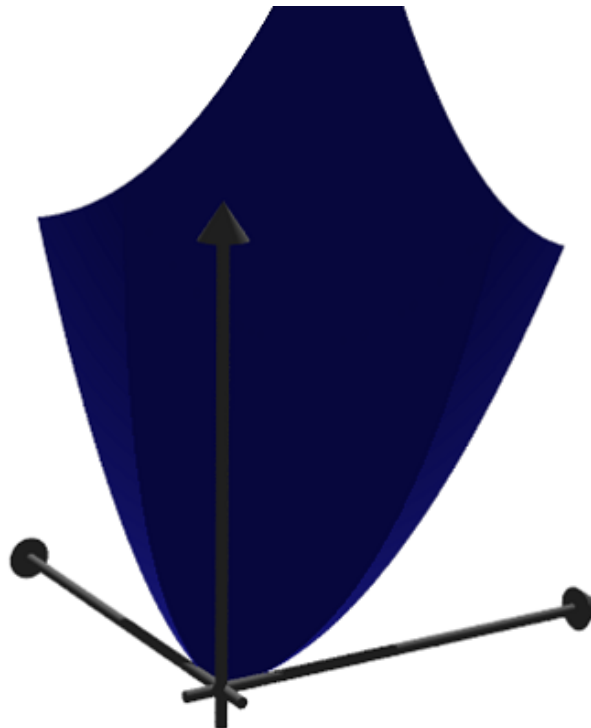
1. Buatlah objek mesh yang mewakili permukaan

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

untuk x dan y dalam interval $[0, 1]$. Tambahkan sebuah disk yang berpotongan dengan permukaan tersebut pada posisi $(0.5, 0.5, 0)$ dengan jari-jari 0.3. Tampilkan hasil pemotongan dalam warna biru.

JAWAB :

```
>povstart(angle=-30°, center=[0.5, 0.5, 0.5], zoom=6);
>x = 0:0.02:1; y = x';
>z = x.^2 + y.^2;
>vx = -2*x;
>vy = -2*y;
>vz = ones(size(x));
>mesh = povtriangles(x, y, z, "", vx, vy, vz);
>disk = povdisc([0.5, 0.5, 0], [0, 0, 1], 0.3);
>writeln(povdifference(mesh, disk, povlook(blue)));
>writeAxes(0, 1, 0, 1, 0, 1, d=0.015);
>povend();
```



2. Buatlah objek mesh untuk fungsi

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

untuk x dan y dalam interval $[-1, 1]$. Tambahkan dua cakram yang berpotongan dengan permukaan tersebut: satu di $(0,0,0)$ dengan jari-jari 0.5 dan yang lainnya di $(0,0,0.5)$ dengan jari-jari 0.2. Tampilkan hasil pemotongan dengan warna ungu dan tunjukkan titik maksimum pada permukaan.

JAWAB:

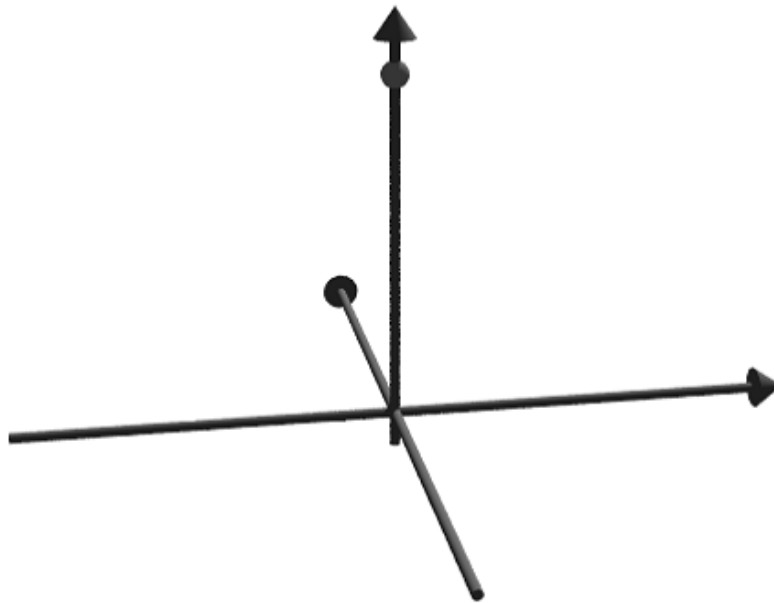
```
>povstart(angle=-10°, center=[0, 0, 0.5], zoom=5);
>x = -1:0.02:1; y = x';
>z = 1 - x.^2 - y.^2;
>vx = -2*x;
>vy = -2*y;
>vz = ones(size(x));
>mesh = povtriangles(x, y, z, "", vx, vy, vz);
>disk1 = povdisc([0, 0, 0], [0, 0, 1], 0.5);
>disk2 = povdisc([0, 0, 0.5], [0, 0, 1], 0.2);
>writeln(povdifference(mesh, povunion([disk1, disk2]), povlook(purple)));
```

Variable or function purple not found.

Error in:

```
... ce(mesh, povunion([disk1, disk2]), povlook(purple))); ...
      ^
```

```
>writeln(povpoint([0, 0, 1], povlook(gray), size=2*defaultpointsize));
>writeAxes(-1, 1, -1, 1, 0, 1, d=0.015);
>povend();
```



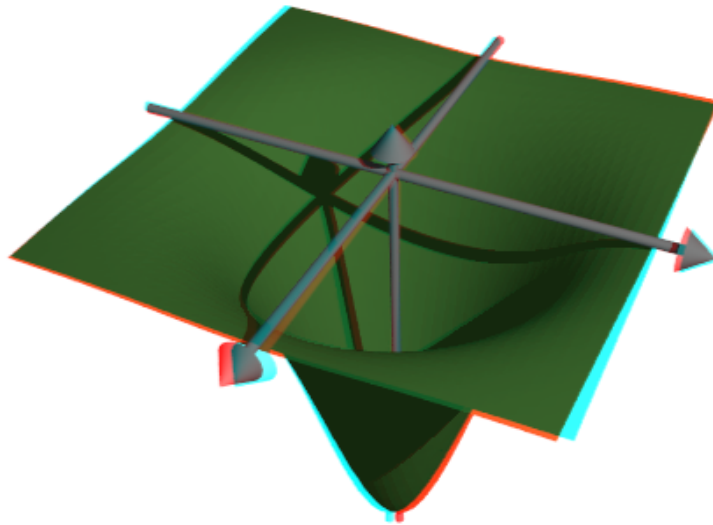
Anaglyph di Povray

Untuk menghasilkan anaglyph untuk kaca mata 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 `load-anaglyph ()`.

Tentu saja, Anda memerlukan kaca mata 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 adegan dengan objek, Anda perlu memasukkan pembuatan adegan ke dalam fungsi, dan menjalankannya dua kali dengan nilai yang berbeda untuk parameter anaglyph.

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

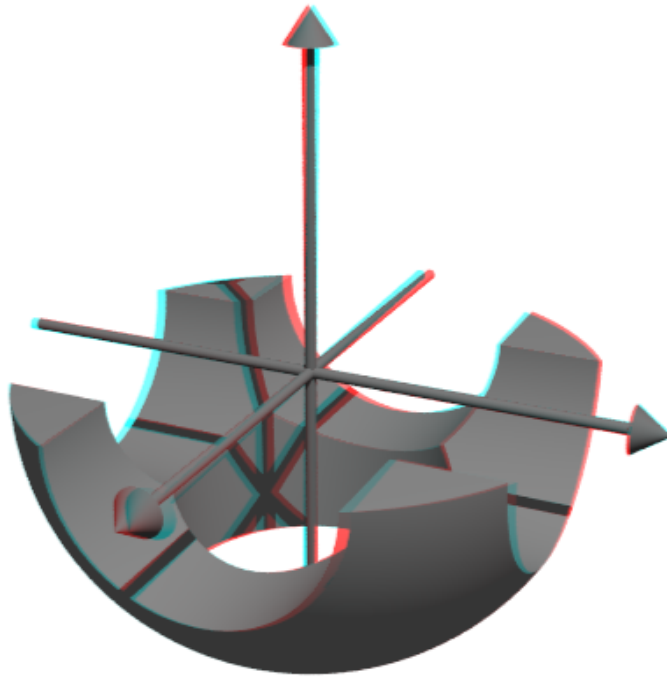
```

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. Parameternya seperti di povstart () dan povend () digabungkan.

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

Mendefinisikan Objek Sendiri

Antarmuka povray Euler berisi banyak objek. Tetapi Anda tidak dibatasi untuk ini. Anda dapat membuat objek sendiri, yang menggabungkan objek lain, atau merupakan objek yang sama sekali baru.

Kami mendemonstrasikan torus. Perintah Povray untuk ini adalah "torus". Jadi kami mengembalikan string dengan perintah ini dan parameternya. Perhatikan bahwa torus selalu berpusat pada asalnya.

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

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

Here is our first torus.

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

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

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

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

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

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

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

```
>povend();
```

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

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

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

```
> povend ();
```

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

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

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

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



Berikut adalah contoh yang lebih lengkap. Kami menyelesaikannya

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

dan menunjukkan poin 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.

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

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

Yes, it has.

Next we define two objects. The first is the plane

$$a \cdot x \leq b$$

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

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

Kemudian kami mendefinisikan perpotongan dari semua setengah spasi dan sebuah 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
```

We can now plot the scene.

```
>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); ...
```

The following is a circle around the optimum.

–Terjemahan

Berikut ini adalah lingkaran di sekitar optimal.

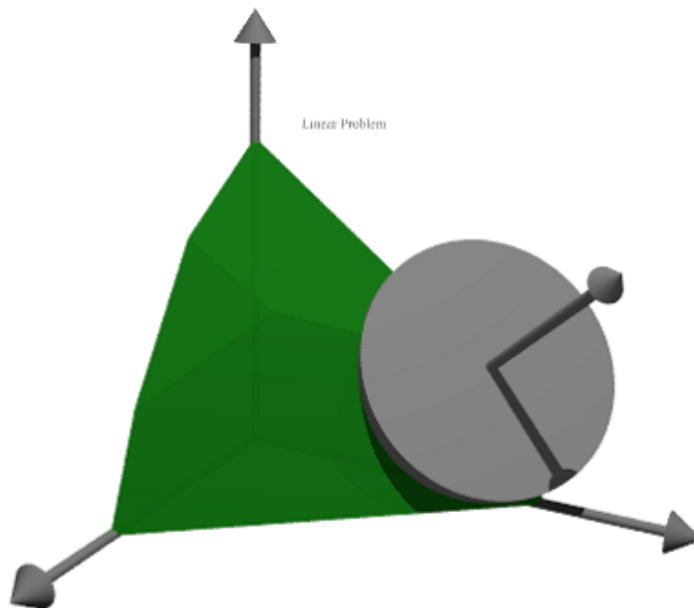
```
>writeln(povintersection([povsphere(x,0.5),povplane(c,c.x')], ...
> povlook(red,0.9)));
```

Dan ada kesalahan di arah optimal.

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

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

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



Contoh Lainnya

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

See: Examples/Dandelin Spheres

See: Examples/Donat Math

See: Examples/Trefoil Knot

See: Examples/Optimization by Affine Scaling

Latihan Soal

1. Buatlah objek torus dengan radius luar 1.0 dan radius dalam 0.3, dan tempatkan objek ini di koordinat (1, 0, 0). Tampilkan objek ini dengan Phong shading.

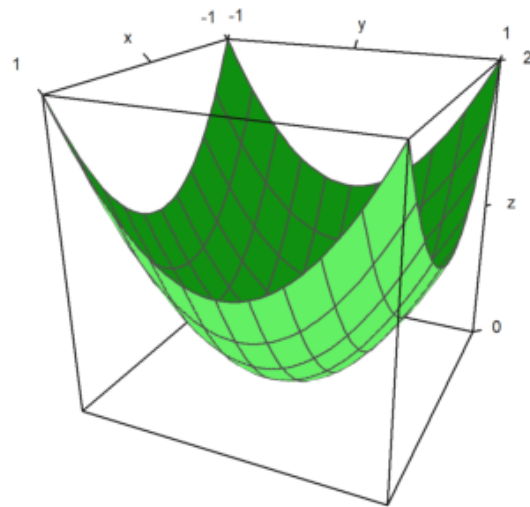
JAWAB :

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

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

```
>t1 = povdonat(1.0, 0.3); ...  
>povstart(center=[1, 0, 0], angle=0°, zoom=3); ...  
>writeln(povobject(t1, povlook(green, phong=1))); ...  
>povend() :
```





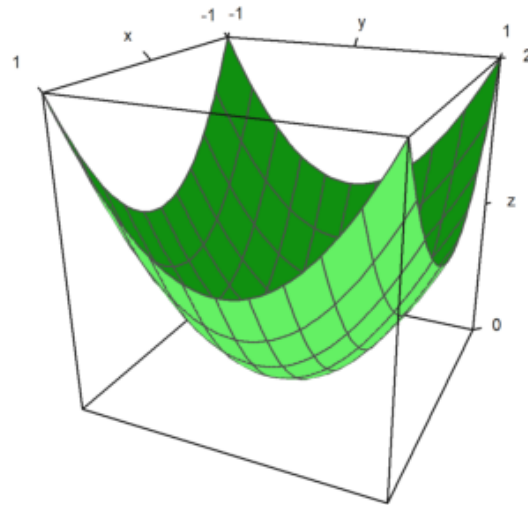
2. Buatlah objek torus dengan radius luar 1.0 dan radius dalam 0.4, dan tempatkan objek ini di koordinat (0, 0, 0).

JAWAB:

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

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

```
>t1 = povdonat(1.0, 0.4);
>povstart(center=[0, 0, 0], angle=0°, zoom=3); ...
>writeln(povobject(t1, povlook(green))):
```



3. Buatlah dua objek kubus dengan ukuran 1 yang berada pada titik (0, 0, 0) dan (1, 1, 1). Gunakan warna biru untuk kubus pertama dan merah untuk kubus kedua.

JAWAB:

```
>function povbox(center, size) ...
```

```
    $ return "box {" + center + ", " + size + "}";
    $endfunction
```

```
>cube1 = povbox([0, 0, 0], [1, 1, 1]);
```

```
Need matrices or strings for |
Try "trace errors" to inspect local variables after errors.
povbox:
    return "box {" + center + ", " + size + "}";
Error in:
cube1 = povbox([0, 0, 0], [1, 1, 1]); ...
      ^
```

```
>cube2 = povbox([1, 1, 1], [1, 1, 1]);
```

```
Need matrices or strings for |
Try "trace errors" to inspect local variables after errors.
povbox:
    return "box {" + center + ", " + size + "}";
Error in:
cube2 = povbox([1, 1, 1], [1, 1, 1]); ...
      ^
```

```
>povstart(center=[0.5, 0.5, 0.5], angle=45°, zoom=3); ...
>writeln(povobject(cubel, povlook(blue))); ...
>writeln(povobject(cube2, povlook(red))); ...
>povend()
```

Variable or function cubel not found.

Error in:

```
... 0.5], angle=45°, zoom=3); writeln(povobject(cubel, povlook(blue ...
^
```

4. Buatlah objek bola dengan jari-jari 0.5 yang terletak di koordinat (0, 0, 0) dan tampilkan dengan efek refleksi.
JAWAB:

```
>function povsphere(radius, look="") ...
```

```
$ return "sphere {" + radius + look + "}";
$endfunction
```

```
>ball = povsphere(0.5, povlook(green, reflection=0.8));
>povstart(center=[0, 0, 0], angle=0°, zoom=3);
>writeln(ball);
>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);
```

5. Buatlah dua objek silinder dengan tinggi 1 dan jari-jari 0.3 yang terletak pada posisi (0, 0, 0) dan (1, 0, 0).
Tampilkan dengan warna kuning dan ungu.

JAWAB:

```
>function povcylinder(radius, height, look="") ...
```

```
$ return "cylinder {" + radius + ", " + height + look + "}";
$endfunction
```

```
>cylinder1 = povcylinder(0.3, 1, povlook(yellow));
>cylinder2 = povcylinder(0.3, 1, povlook(purple));
```

Variable or function purple not found.

Error in:

```
cylinder2 = povcylinder(0.3, 1, povlook(purple)); ...
^
```



```
>povstart(center=[0.5, 0, 0], angle=45°, zoom=3);
>writeln(povobject(cylinder1, povobject(translate=[0, 0, 0])));
```

Function povobject needs at least one argument!

Use: povobject (object: string {, look: string, translate: none, rotate: none, scale: no

Error in:

```
... vobject(cylinder1, povobject(translate=[0, 0, 0])); ...
                                     ^
```

```
>writeln(povobject(cylinder2, povobject(translate=[1, 0, 0])));
```

Variable or function cylinder2 not found.

Error in:

```
writeln(povobject(cylinder2, povobject(translate=[1, 0, 0]))); ...
                                     ^
```

```
>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);
```