

Nama : Rifina Rizki Alfitasari
NIM : 22305141047
Kelas : Matematika B 2022

Menggambar Plot 3D dengan EMT

Ini adalah pengantar untuk plot 3D dalam Euler. Kami memerlukan plot 3D untuk memvisualisasikan fungsi dari dua variabel.

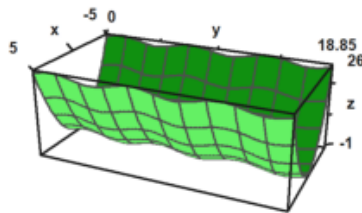
Euler menggambar fungsi-fungsi seperti ini menggunakan algoritma pengurutan untuk menyembunyikan bagian-bagian di latar belakang. Secara umum, Euler menggunakan proyeksi sentral. Secara default, proyeksi ini dari kuadran positif x-y menuju titik asal $x=y=z=0$, tetapi sudut $=0^\circ$ mengarah ke arah sumbu y. Sudut pandangan dan tinggi dapat diubah.

Euler dapat menggambar

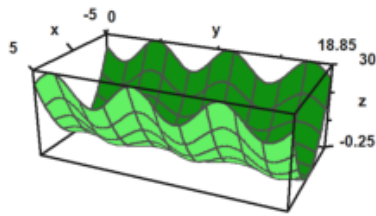
- permukaan dengan shading dan garis level atau rentang level,
- awan titik-titik,
- kurva parametrik,
- permukaan implisit.

Plot 3D dari sebuah fungsi menggunakan perintah `plot3d`. Cara termudah adalah dengan 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):
```

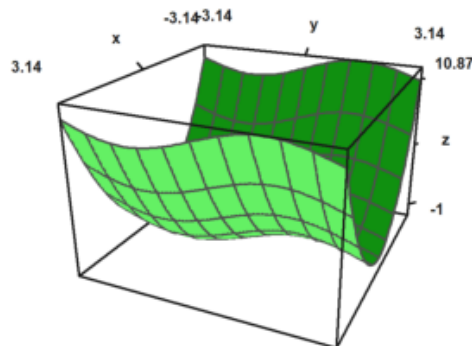


```
>plot3d("x^2+x*sin(y)",-5,5,0,6*pi):
```



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+sin(y)", r=pi):
```



Fungsi dari Dua Variabel

Untuk grafik suatu fungsi, gunakan

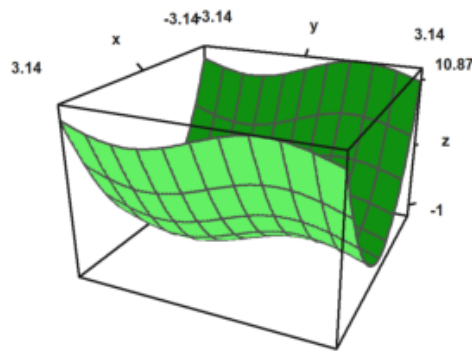
- ekspresi sederhana dalam x dan y ,
- nama fungsi dari dua variabel,
- atau matriks data.

Pengaturan default adalah kisi kawat yang terisi dengan warna yang berbeda di kedua sisi. Perlu diperhatikan bahwa jumlah interval kisi default adalah 10, tetapi plot menggunakan jumlah kotak 40×40 yang default untuk membentuk permukaan. Ini dapat diubah.

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

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

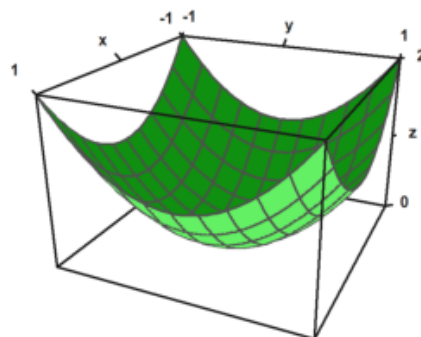
```
>plot3d("x^2+y^2"):
```



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

- kiri, kanan, atas, bawah: mengubah sudut pandang
- +, - : memperbesar atau memperkecil
- a: menghasilkan gambar anaglif (lihat di bawah)
- l: mengaktifkan atau menonaktifkan sumber cahaya (lihat di bawah)
- spasi: mengembalikan ke pengaturan awal
- enter: mengakhiri interaksi

```
>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: sebuah kotak simetris di sekitar (0,0).
- n: jumlah subinterval untuk plot.

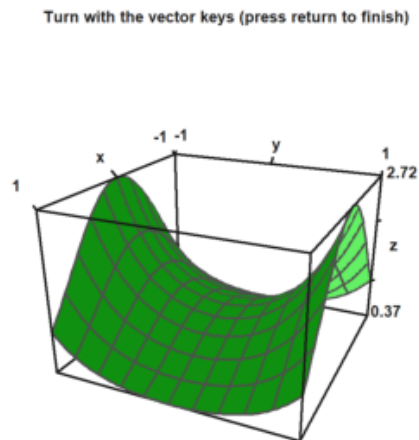
Terdapat beberapa parameter untuk mengubah skala fungsi atau mengubah tampilan grafik.

`fscale`: mengubah skala nilai-nilai fungsi (default adalah `<fscale>`).

`skala`: angka atau vektor 1x2 untuk mengubah skala dalam arah x dan y.

`bingkai`: jenis bingkai (default 1).

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



Tampilan dapat diubah dengan berbagai cara yang berbeda.

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

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

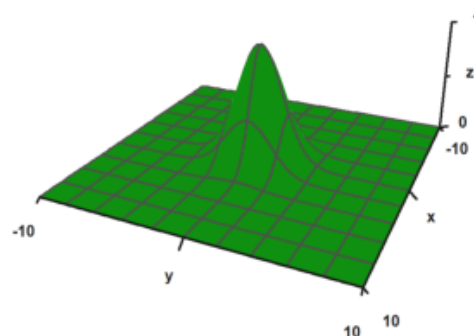
```
>view
```

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

Jarak yang lebih dekat memerlukan zoom yang lebih kecil. Efeknya lebih mirip dengan lensa sudut lebar.

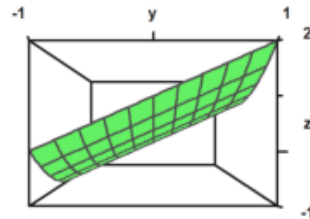
Pada contoh berikut, sudut=0 dan tinggi=0 dilihat dari sumbu y negatif. Label sumbu untuk y disembunyikan dalam kasus ini.

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



Plot selalu melihat ke tengah kubus plot. Anda dapat memindahkan tengahnya 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 tersebut diubah skala untuk masuk ke dalam kubus satuan untuk ditampilkan. Jadi, tidak perlu mengubah jarak atau zoom tergantung pada ukuran plot. Label-label mengacu pada ukuran sebenarnya, namun. Jika Anda memamatkannya dengan `scale=false`, Anda perlu berhati-hati agar plot tetap muat ke dalam jendela plotting, dengan mengubah jarak pandang atau zoom, dan memindahkan pusatnya.

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

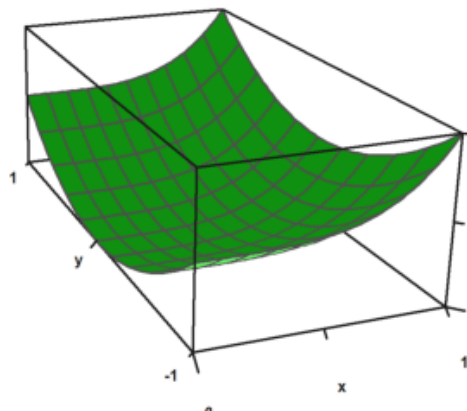
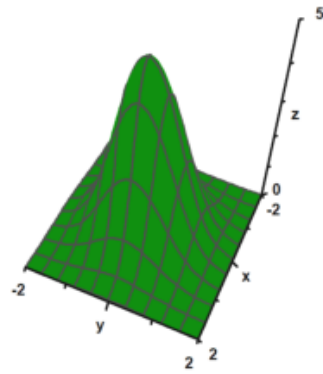
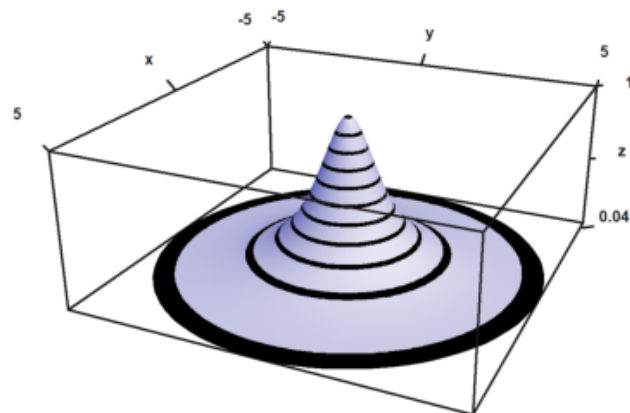


Diagram polar juga tersedia. Parameter `polar=true` menggambar diagram polar. Fungsi harus tetap menjadi fungsi dari x dan y . Parameter `fscale` mengukur fungsi dengan skala sendiri. Jika tidak, fungsi akan diukur untuk sesuai ke dalam sebuah kubus.

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



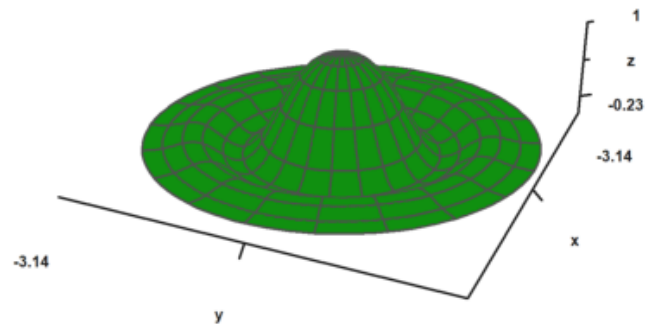
```
>function f(r) := exp(-r/2)*cos(r); ...
>plot3d("f(x^2+y^2)",>polar,scale=[1,1,0.4],r=pi,frame=3,zoom=4):
```



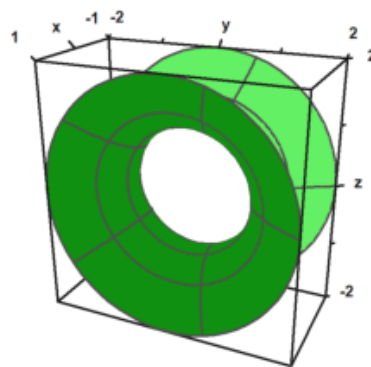
Parameter rotate memutar suatu fungsi dalam sumbu x sekitar sumbu x.

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

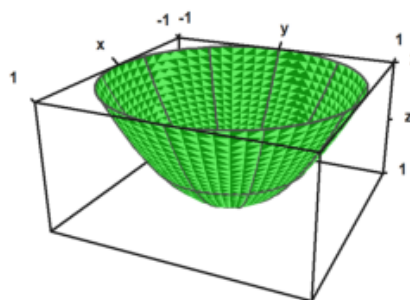
```
>plot3d("x^2+1",a=-1,b=1,rotate=true,grid=5):
```



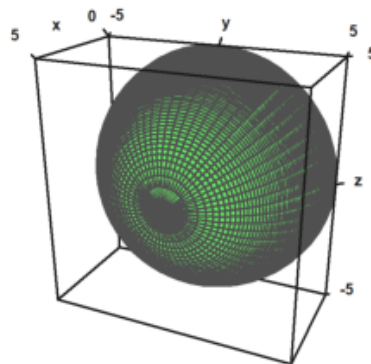
```
>plot3d("x^2+1",a=-1,b=1,rotate=2,grid=5):
```



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

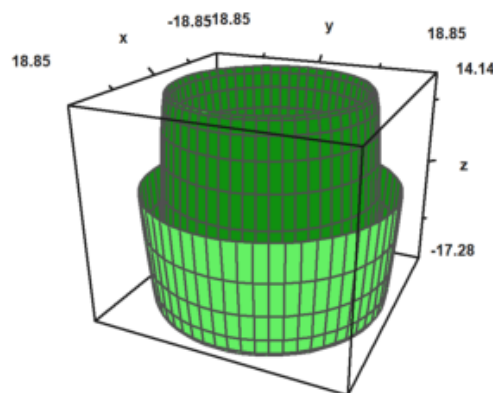


```
>plot3d("x*sin(x)",a=0,b=6pi,rotate=2):
```



Berikut adalah plot dengan tiga fungsi.

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



Plot Kontur

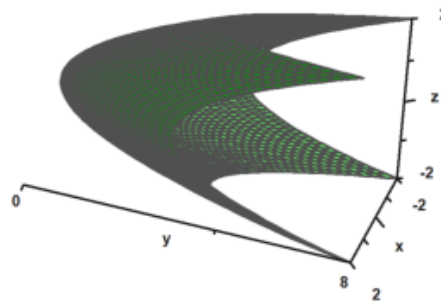
Untuk plot ini, Euler menambahkan garis-garis grid. Namun, mungkin juga digunakan garis-garis level dan tampilan warna tunggal atau tampilan warna spektral. Euler dapat menggambar ketinggian fungsi pada plot dengan shading. Pada semua plot 3D, Euler dapat menghasilkan anaglif merah/sian.

- >hue: Mengaktifkan shading ringar daripada kawat.
- >contour: Menampilkan garis kontur otomatis pada plot.
- level=... (atau levels): Vektor nilai untuk garis kontur.

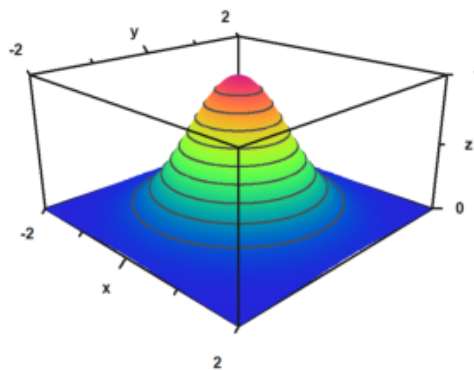
Defaultnya adalah level="auto", yang menghitung beberapa garis level secara otomatis. Seperti yang Anda lihat dalam plot, level sebenarnya adalah rentang level.

Gaya default dapat diubah. Untuk plot kontur berikutnya, kami menggunakan grid yang lebih halus dengan 100x100 titik, menyesuaikan fungsi dan plot, serta 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):
```



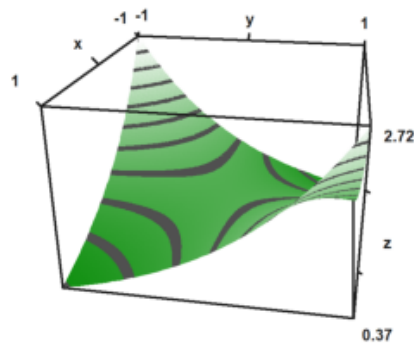
Bayangan default menggunakan warna abu-abu. Tetapi juga tersedia rentang warna spektral.

- >spektral: Menggunakan skema spektral default

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

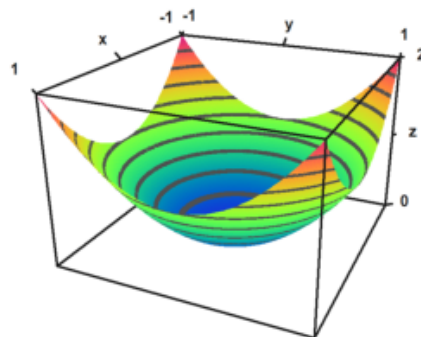
Untuk plot berikutnya, kami menggunakan skema spektral default dan meningkatkan jumlah titik untuk mendapatkan tampilan yang sangat halus.

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



Selain garis level otomatis, kita juga dapat menetapkan nilai garis level. Ini akan menghasilkan garis level yang tipis, bukan rentang level.

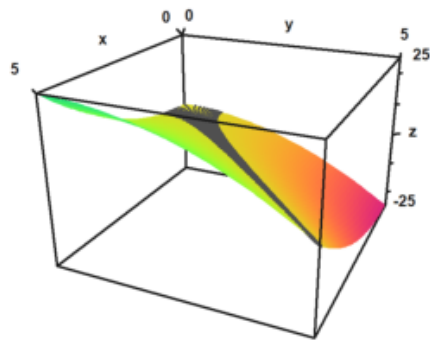
```
>plot3d("x^2-y^2",0,5,0,5,level=-1:0.1:1,color=redgreen):
```



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

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

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



Pada contoh berikut, kita memplot himpunan, di mana

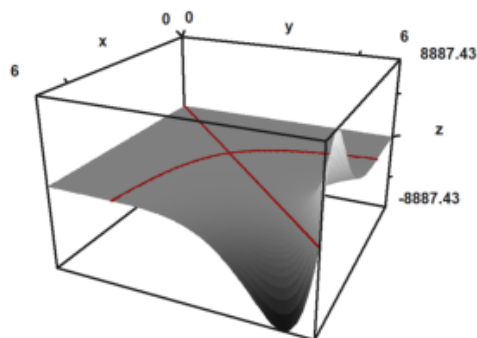
$$f(x, y) = x^y - y^x = 0$$

Kami menggunakan satu garis tipis untuk garis level.

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

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

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

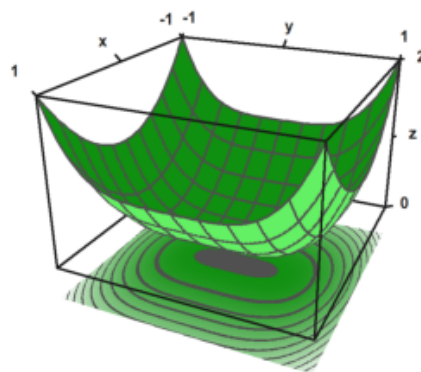


Berikut beberapa gaya lainnya. Kami selalu mematikan bingkai, dan menggunakan berbagai skema warna untuk plot dan kisi.

```

>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):

```



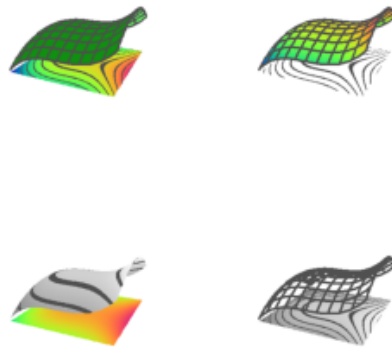
Ada beberapa skema spektral lainnya, bernomor dari 1 hingga 9. Namun, Anda juga dapat menggunakan warna=nilai, di mana nilai

- spektral: untuk rentang dari biru ke merah
- putih: untuk rentang yang lebih lemah
- kuningbiru, unguhijau, birukuning, merahhijau
- birukuning, unguhijau, kuningbiru, merahhijau

```

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

```

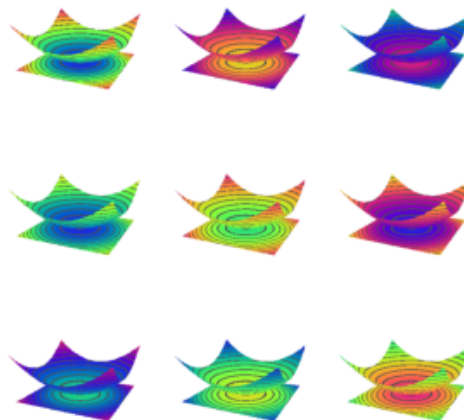


Sumber cahaya dapat diubah dengan tombol 'l' dan tombol kursor selama interaksi pengguna. Ini juga dapat diatur dengan parameter.

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

Perhatikan bahwa program tidak membedakan antara sisi-sisi dari plot. Tidak ada bayangan. Untuk ini, Anda memerlukan Povray.

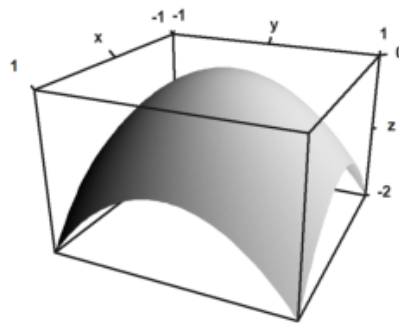
```
>plot3d("-x^2-y^2", ...
> hue=true,light=[0,1,1],amb=0,user=true, ...
> title="Press l and cursor keys (return to exit)":
```



Parameter warna mengubah warna permukaan. Warna garis level juga bisa 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):
```

Press i and cursor keys (return to exit)



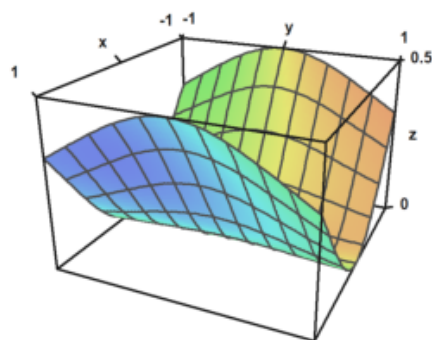
Warna 0 memberikan efek pelangi yang istimewa.

```
>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-objek tersebut. Fitur-fitur dari plot3d termasuk plot implisit. Plot ini menunjukkan himpunan nol dari sebuah 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: potongan sejajar dengan bidang y-z
- implisit=2: potongan sejajar dengan bidang x-z
- implisit=4: potongan sejajar dengan bidang x-y

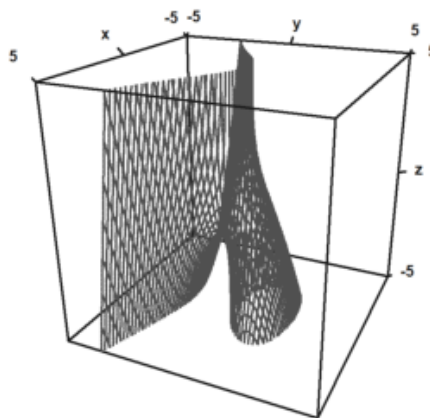
Tambahkan nilai-nilai ini, jika Anda suka. Dalam contoh ini, kita membuat plot

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

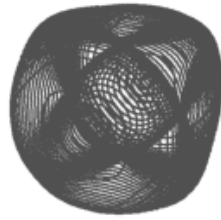
```
>plot3d("x^2+y^3+z*y-1",r=5,implicit=3):
```

```
>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):
```



Plot Data 3D

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 mengasumsikan bahwa (t,s) berjalan melalui grid persegi. Sebagai hasilnya, Anda dapat membuat gambar persegi panjang dalam ruang.

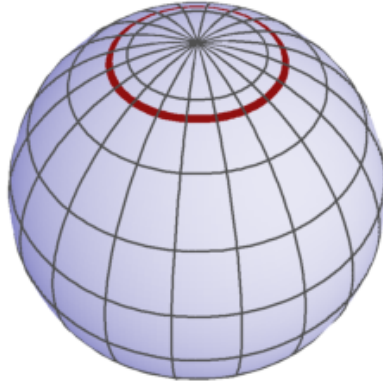
Anda dapat menggunakan bahasa matriks Euler untuk menghasilkan koordinat secara efektif.

Pada contoh berikut, kita menggunakan vektor nilai t dan vektor kolom nilai s untuk parameterisasi permukaan bola. Dalam gambar, kita dapat menandai wilayah-wilayah, dalam kasus kami adalah wilayah 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 contohnya yaitu grafik suatu fungsi.

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

Namun, kita dapat membuat berbagai macam permukaan. Berikut adalah permukaan yang sama sebagai fungsi:

$$x = yz$$

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

Dengan usaha lebih, kita dapat menghasilkan banyak permukaan.

Pada contoh berikut, kita membuat tampilan yang teduh dari bola yang distorsi. Koordinat biasa untuk bola tersebut adalah

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

dengan

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

Kita merubahnya dengan faktor

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

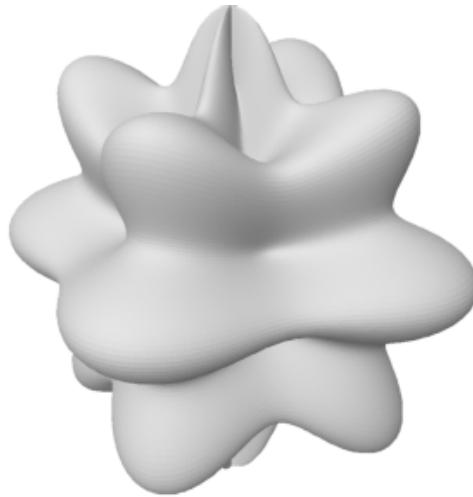
```
>t=linspace(0,2pi,320); s=linspace(-pi/2,pi/2,160)'; ...
>d=1+0.2*(cos(4*t)+cos(8*s)); ...
>plot3d(cos(t)*cos(s)*d,sin(t)*cos(s)*d,sin(s)*d,hue=1, ...
> light=[1,0,1],frame=0,zoom=5):
```

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

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

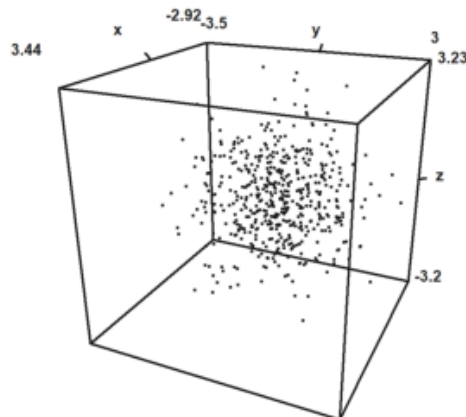
Gayanya sama seperti di plot2d dengan points=true;

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

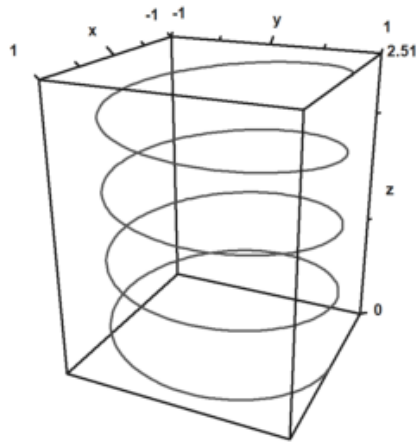


Dimungkinkan juga untuk memplot kurva dalam 3D. Dalam hal ini, lebih mudah untuk menghitung terlebih dahulu titik-titik kurva. Untuk kurva pada bidang kita menggunakan barisan koordinat dan parameter `wire=true`.

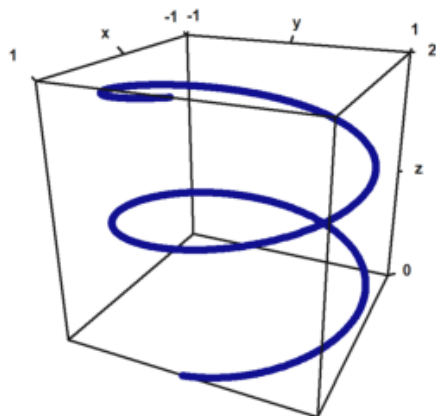
```
>t=linspace(0,8pi,500); ...
>plot3d(sin(t),cos(t),t/10,>wire, zoom=3):
```



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

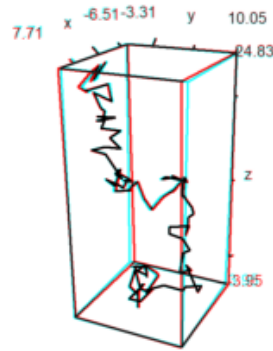


```
>X=cumsum(normal(3,100)); ...
> plot3d(X[1],X[2],X[3],>anaglyph,>wire):
```



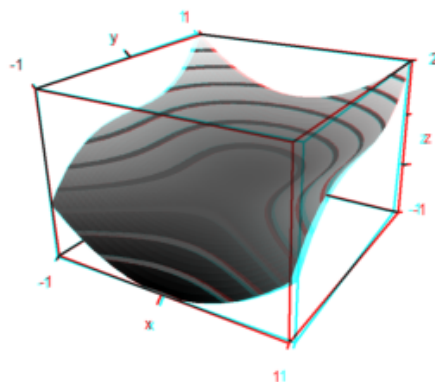
EMT juga dapat membuat plot dalam mode anaglyph. Untuk melihat plot seperti itu, Anda memerlukan kacamata berwarna merah/sian.

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



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

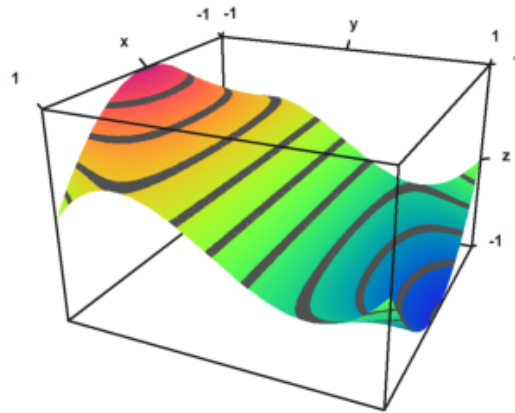
```
>plot3d("x^2*y^3-y",>spectral,>contour, zoom=3.2) :
```



Euler juga dapat memplot permukaan yang diparameterisasi, jika parameternya adalah nilai x , y , dan z dari gambar kotak persegi panjang di ruang tersebut.

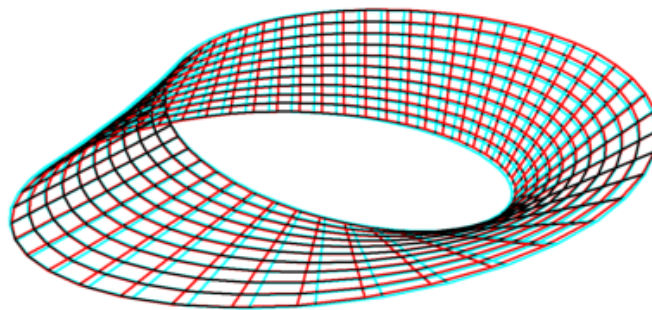
Untuk demo berikut, kami menyiapkan parameter u - dan v -, dan menghasilkan koordinat ruang dari parameter tersebut.

```
>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 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):
```



Grafik Statistik

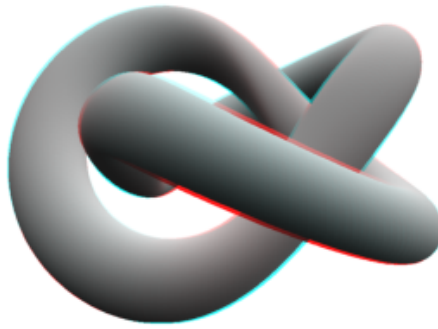
Grafik batang juga memungkinkan. Untuk ini, kita harus menyediakan

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

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

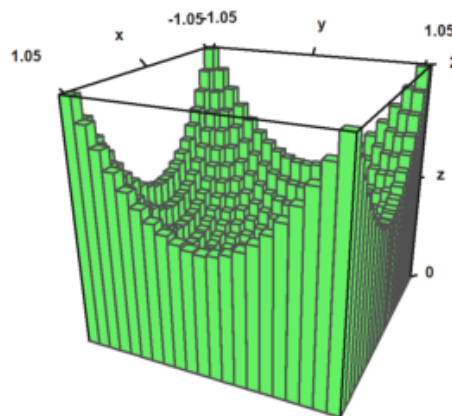
Dalam contoh ini, kita pertama-tama menghitung nilai-nilainya. Kemudian kita menyesuaikan x dan y, sehingga vektor-vektor tersebut berpusat pada nilai yang digunakan.

```
>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):
```



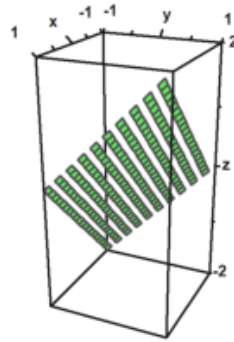
Dimungkinkan untuk membagi plot suatu permukaan menjadi dua bagian atau lebih.

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

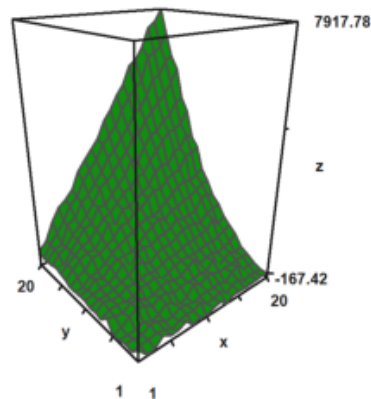


Jika Anda memuat atau menghasilkan matriks data M dari sebuah file dan perlu membuat plotnya dalam tiga dimensi (3D), Anda dapat mengubah skala matriks tersebut menjadi $[-1,1]$ dengan menggunakan perintah "scale(M)", atau mengubah skala matriks dengan menggunakan perintah ">zscale". Ini dapat digabungkan dengan faktor-faktor skala individu yang diterapkan secara 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):
```

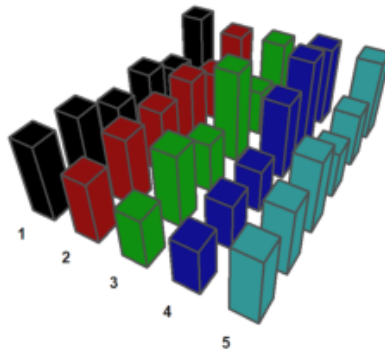


```
>Z=intrandom(5,100,6); v=zeros(5,6); ...
>loop 1 to 5; v[#]=getmultiplicities(1:6,Z[#]); end; ...
>columnsplot3d(v',scols=1:5,ccols=[1:5]):
```



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 hati sekitar sumbu y. Berikut adalah ungkapan yang mendefinisikan bentuk hati:

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

Selanjutnya, kami mengatur

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

```
>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 r, jika a diberikan. Dengan fungsi tersebut kita dapat memplot jantung yang diputar sebagai permukaan parametrik.

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

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

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

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



```

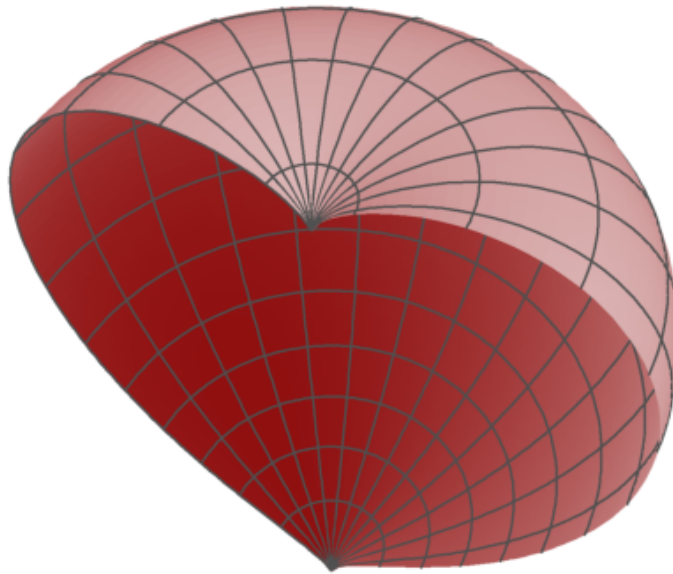
r=x^2+y^2;
return (r+z^2-1)^3-r*z^3;
endfunction

```

```

>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=[10,100,60],>anaglyph):

```



Plot 3D Khusus

Fungsi plot3d ini bagus untuk dimiliki, tetapi tidak memenuhi semua kebutuhan. Selain rutinitas yang lebih dasar, mungkin Anda bisa mendapatkan plot bingkai dari objek apa pun yang Anda sukai.

Meskipun Euler bukanlah program 3D, namun bisa menggabungkan beberapa objek dasar. Kami mencoba untuk memvisualisasikan sebuah paraboloid beserta tangannya.

```

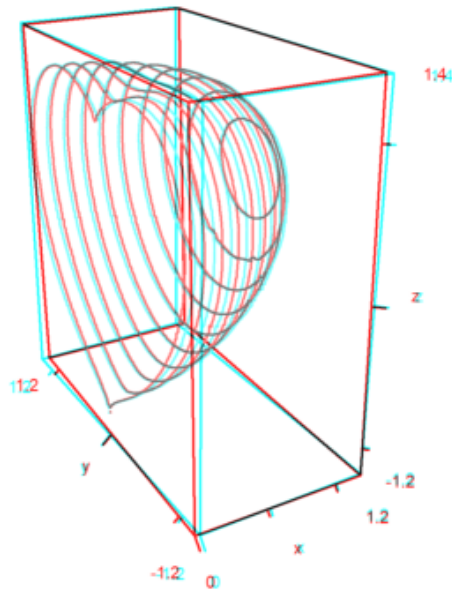
>function myplot ...

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 bingkai, dan mengatur tampilan.

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

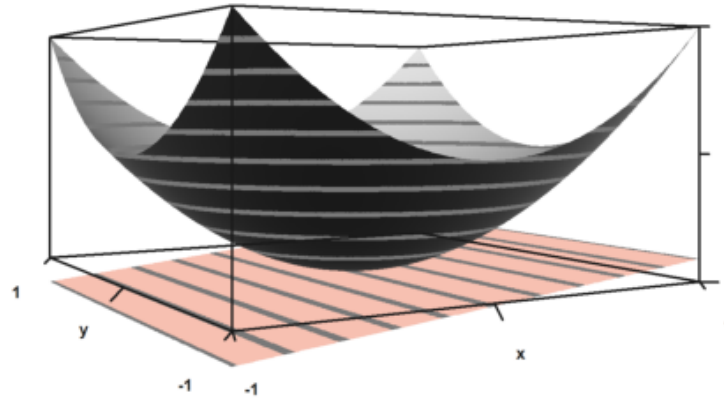


Dengan cara yang sama, Anda dapat memplot bidang kontur secara manual. Perhatikan bahwa `plot3d()` menyetel jendela ke `fullwindow()` secara default, tetapi `plotcontourplane()` berasumsi demikian.

```
>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):
```



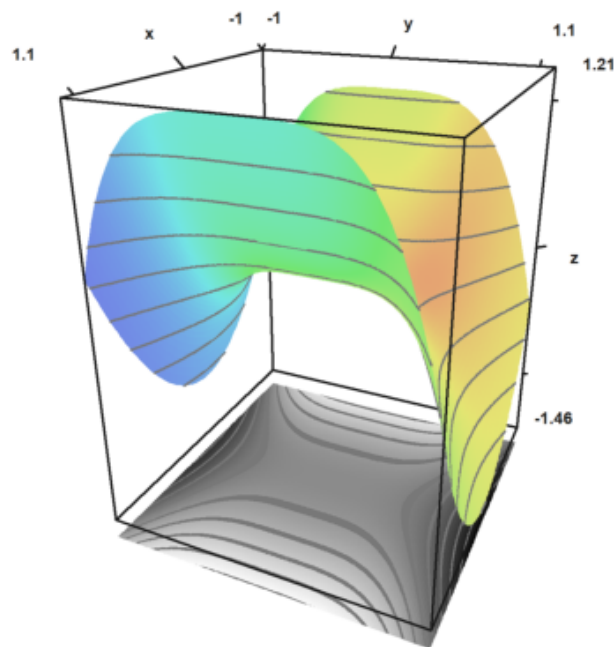
Animasi

Euler dapat menggunakan bingkai untuk melakukan perhitungan awal animasi.

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

Silakan pelajari sumber kode dari fungsi `rotate` untuk melihat lebih banyak detailnya.

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



Menggambar dengan Povray

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

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

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

Fungsi pov3d mirip 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 secara otomatis memulai raytracer, dan memuat tampilan ke dalam buku catatan Euler.

Selain pov3d(), ada banyak fungsi lain yang menghasilkan objek Povray. Fungsi-fungsi ini mengembalikan string-string yang berisi kode Povray untuk objek-objek tersebut. Untuk menggunakan fungsi-fungsi ini, mulailah berkas Povray dengan povstart(). Kemudian gunakan writeln(...) untuk menulis objek-objek ke dalam berkas tampilan. Akhirnya, selesaikan berkas dengan povend(). Secara default, raytracer akan dijalankan, dan PNG akan dimasukkan ke dalam buku catatan Euler.

Fungsi objek memiliki parameter yang disebut "look", yang memerlukan 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 sumbu z mengarah ke atas secara vertikal, dan sumbu x, y, z dalam arah searah jarum jam.

Anda perlu memuat berkas povray.

```
>load povray;
```

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

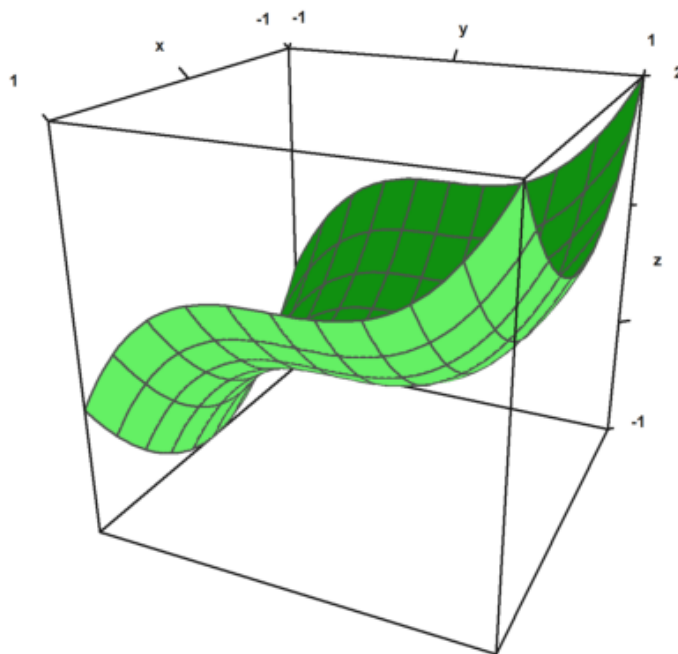
```
>defaultpovray="C:\Program Files\POV-Ray\v3.7\bin\pvengine.exe"
```

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

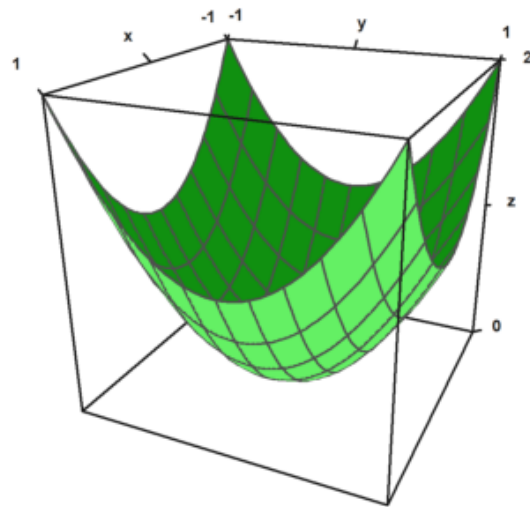
Untuk kesan pertama, kita membuat plot dari sebuah fungsi sederhana. Perintah berikut menghasilkan sebuah file povray di direktori pengguna Anda, dan menjalankan Povray untuk melakukan ray tracing terhadap file ini.

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

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

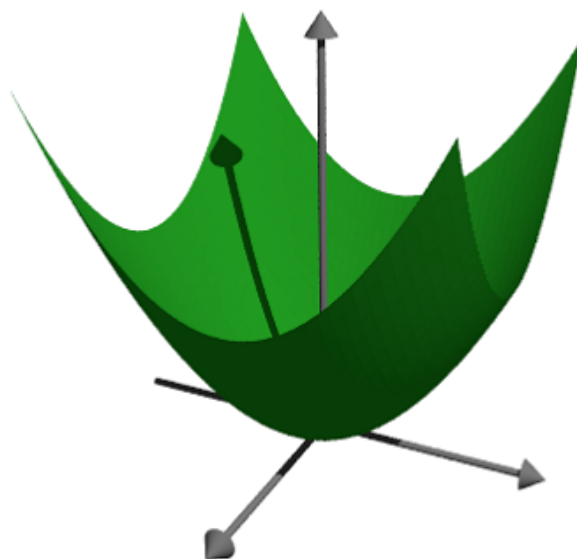


```
>pov3d("x^2+y^2",zoom=3);
```



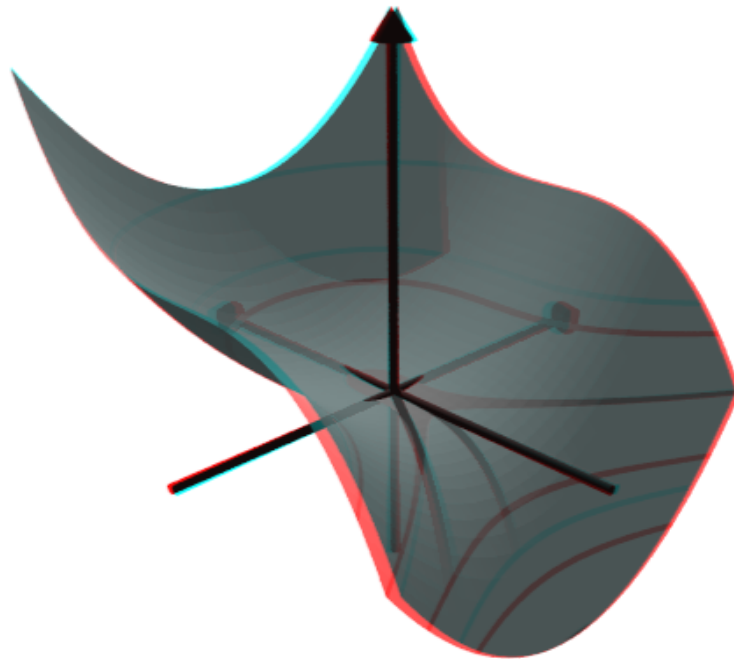
Kita dapat membuat fungsinya transparan dan menambahkan penyelesaian lainnya. Kita juga dapat menambahkan garis level ke plot fungsi.

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



Terkadang perlu untuk mencegah penskalaan fungsi, dan menskalakan fungsi secara manual. Kita memplot himpunan titik pada bidang kompleks, dimana 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);
```

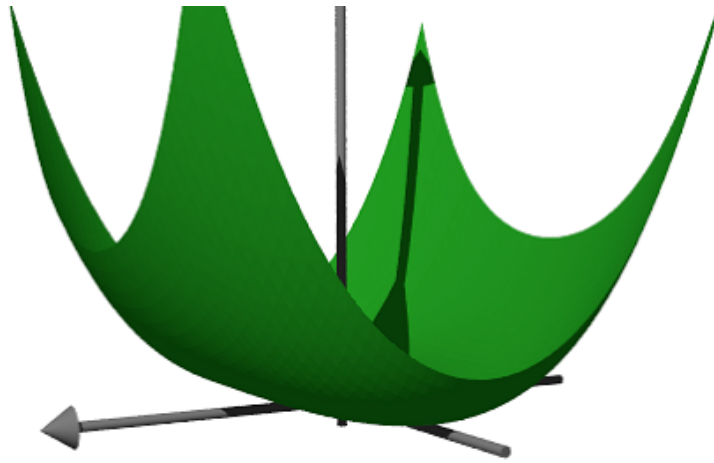


Plotting dengan Koordinat

Daripada menggunakan fungsi, kita dapat melakukan plotting dengan menggunakan koordinat. Seperti pada `plot3d`, kita memerlukan tiga matriks untuk mendefinisikan objek.

Pada contoh ini, kita memutar sebuah fungsi sekitar sumbu z.

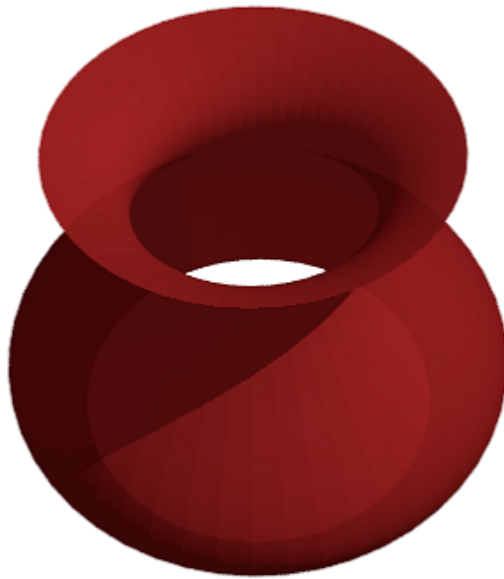
```
>function f(x) := x^3-x+1; ...
>x=-1:0.01:1; t=linspace(0,2pi,50)'; ...
>Z=x; X=cos(t)*f(x); Y=sin(t)*f(x); ...
>pov3d(X,Y,Z,angle=40°,look=povlook(red,0.1),height=50°,axis=0,zoom=4,light=[10,5,15]);
```



Dalam contoh berikut, kita memplot gelombang teredam. Kita menghasilkan gelombang tersebut dengan bahasa matriks Euler.

Kita juga menunjukkan bagaimana objek tambahan dapat ditambahkan ke dalam sebuah adegan pov3d. Untuk menghasilkan objek-objek tersebut, lihat contoh-contoh berikutnya. Perlu diperhatikan bahwa plot3d mengubah skala plot sehingga cocok dalam 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 shading canggih Povray, hanya sedikit titik dapat menghasilkan permukaan yang sangat halus. Hanya di batas-batas dan dalam bayangan trik ini mungkin menjadi jelas. Untuk ini, kita perlu menambahkan vektor normal di setiap titik matriks.

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

$$x^2 y^3$$

Persamaan permukaannya adalah $[x,y,Z]$. Kami menghitung dua turunan dari x dan y dan mengambil perkalian silangnya sebagai normal.

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

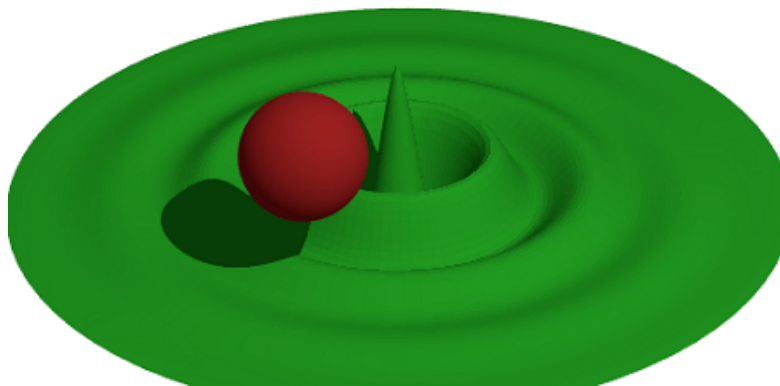
Kami mendefinisikan normal sebagai produk silang dari turunan ini, dan mendefinisikan fungsi koordinat.

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

$$[-\cos(x), 6\sin(3y), 1]$$

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 adalah simpul Trefoil yang dibuat oleh A. Busser dalam Povray. Terdapat versi yang ditingkatkan dari ini dalam contoh-contoh.

Lihat: [Examples\Trefoil Trefoil](#) | [Simpul Trefoil](#)

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

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

Kemudian kedua vektor turunan ke x dan y.

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

Sekarang normalnya, yaitu perkalian silang kedua turunannya.

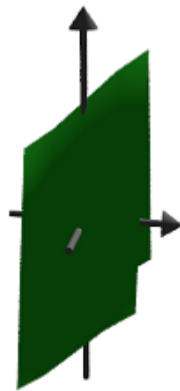
```
>dn &= crossproduct(dx,dy);
```

Kami sekarang mengevaluasi semua ini secara numerik.

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

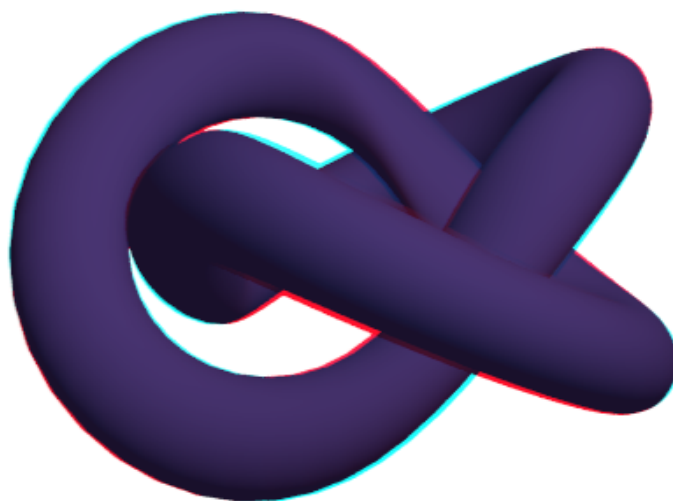
Vektor normal adalah evaluasi ekspresi simbolik $dn[i]$ untuk $i=1,2,3$. Sintaksnya adalah `&"ekspresi"(parameter)`. Ini adalah alternatif dari metode pada contoh sebelumnya, di mana kita mendefinisikan ekspresi simbolik NX , NY , NZ terlebih dahulu.

```
>pov3d(X(x,y),Y(x,y),Z(x,y),>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));
```



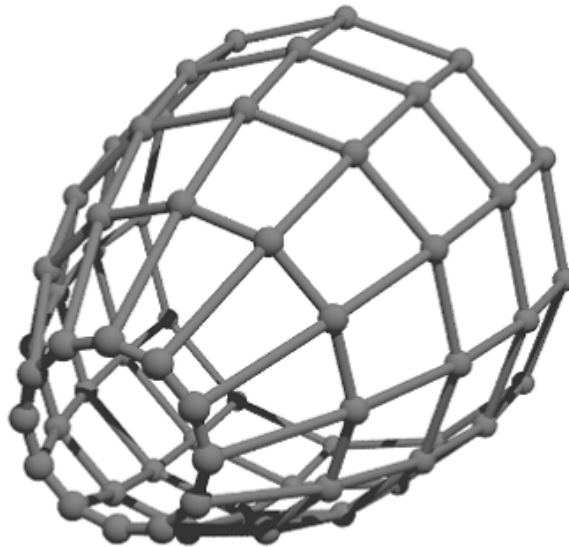
Kami juga dapat menghasilkan grid dalam 3D.

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



Dengan povgrid(), kurva dimungkinkan.

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



Objek Povray

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

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

Pertama kita mendefinisikan tiga silinder, dan menyimpannya dalam string di Euler. Fungsi povx() dll. hanya mengembalikan vektor [1,0,0], yang dapat digunakan sebagai gantinya.

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

String tersebut 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 pada objek dalam tiga warna berbeda.

Hal ini dilakukan oleh `povlook()`, yang mengembalikan 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)
```

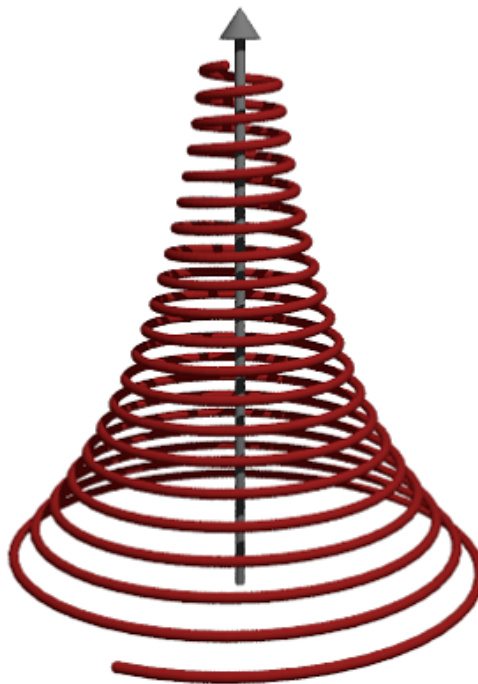
```
texture { pigment { color rgbf <0.101961,0.2,0.301961,0.1> } }  
finish { ambient 0.5 }
```

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

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

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

```
>povend;
```



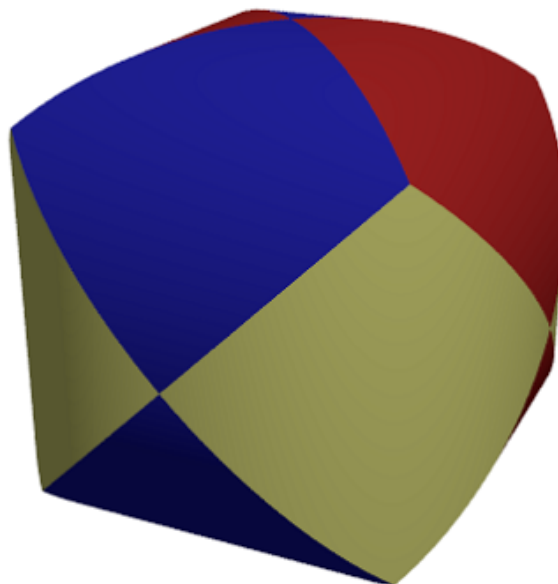
Fungsi berikut menghasilkan fraktal secara rekursif.

Fungsi pertama menunjukkan bagaimana Euler menangani objek Povray sederhana. Fungsi `povbox()` mengembalikan string, 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 pemisahan satu objek dari objek lainnya. Seperti persimpangan, ada bagian dari objek CSG di Povray.

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

Untuk demonstrasi ini, kita mendefinisikan sebuah objek dalam Povray, daripada menggunakan sebuah string dalam Euler. Definisi ditulis ke file secara langsung.

Koordinat sebuah kotak -1 hanya 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 membuat kubus kedua, dan memutar serta menskalakannya sedikit.

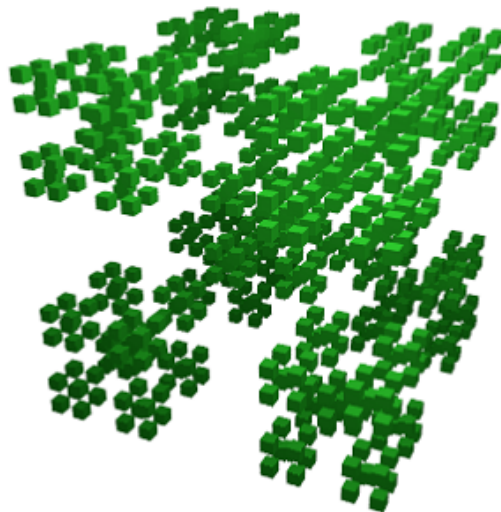
```
>c2=povobject ("mycube",povlook (yellow),translate=[1,1,1], ...  
> rotate=xrotate(10°)+yrotate(10°), scale=1.2);
```

Lalu kita ambil selisih kedua benda tersebut.

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

Sekarang tambahkan tiga sumbu.

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



Fungsi Implisit

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

Sintaks untuk fungsinya sedikit berbeda. Anda tidak dapat menggunakan 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);
```

```
Function povstart not found.
Try list ... to find functions!
Error in:
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();
```

```
Function povlook not found.
Try list ... to find functions!
Error in:
... 2)-pow (c,2),2)+pow (pow (y,2)-1,2) )-d",povlook (red) ) ); povend(); ...
      ^
```



```
>povstart (angle=25°,height=10°);
```

```
Function povstart not found.  
Try list ... to find functions!  
Error in:  
povstart (angle=25°,height=10°); ...  
      ^
```

```
>writeln(povsurface ("pow(x,2)+pow(y,2)*pow(z,2)-1",povlook(blue),povbox(-2,2,"")));
```

```
Function povlook not found.  
Try list ... to find functions!  
Error in:  
... face ("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 sintaksis yang berbeda dalam ekspresi.

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



Objek Mesh

Dalam contoh ini, kami akan menunjukkan bagaimana cara membuat objek mesh, dan menggambarkannya dengan informasi tambahan.

Kami ingin memaksimalkan xy dengan syarat $x+y=1$ 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 bentuk string seperti sebelumnya, karena terlalu besar. Jadi, kita mendefinisikan objek dalam file Povray menggunakan declare. Fungsi povtriangle() melakukannya secara otomatis. Ini dapat menerima vektor normal seperti pov3d().

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

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

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

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

Tulis permukaannya dikurangi kedua cakram.

```
>writeln(povdifference(mesh,povunion([cl,ll]),povlook(green)));
```

Tuliskan kedua perpotongan tersebut.

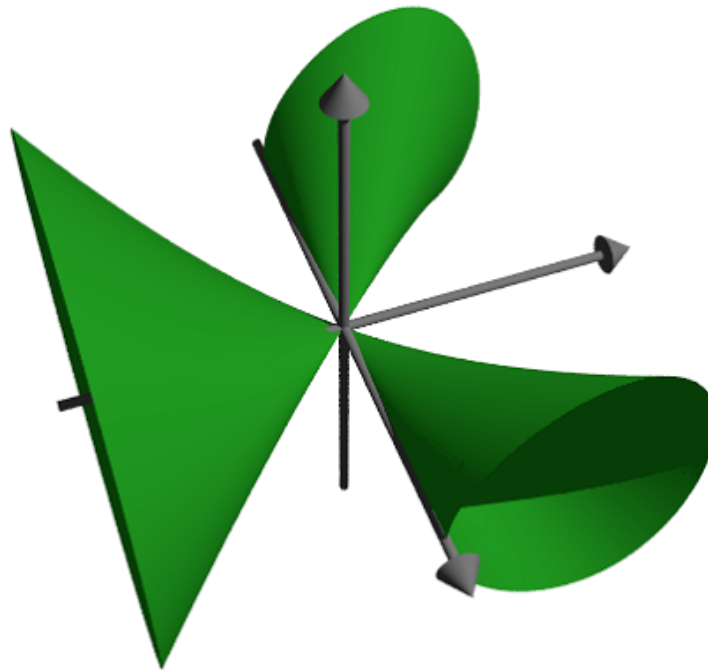
```
>writeln(povintersection([mesh,cl],povlook(red))); ...  
>writeln(povintersection([mesh,ll],povlook(gray)));
```

Tulis poin 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();
```

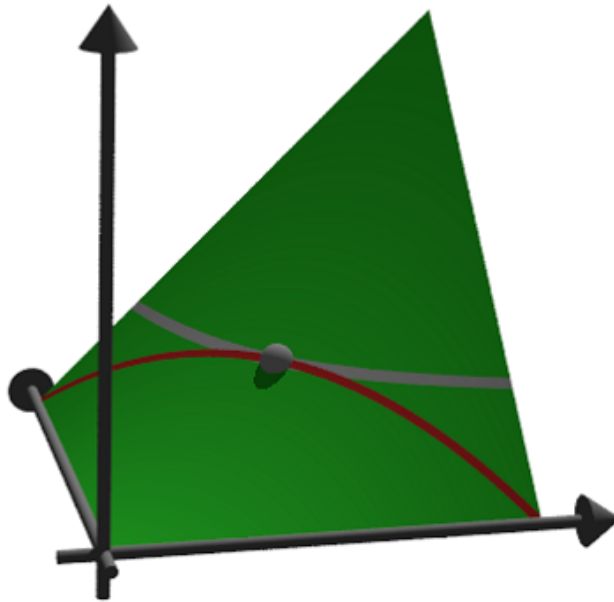


Anaglif dalam Povray

Untuk menghasilkan anaglif 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 load-anaglyph().

Tentu saja, Anda memerlukan kacamata merah/cyan untuk melihat contoh-contoh berikut dengan benar. Fungsi pov3d() memiliki sakelar sederhana untuk menghasilkan anaglif.

```
>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 berbeda untuk parameter anaglyph.

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

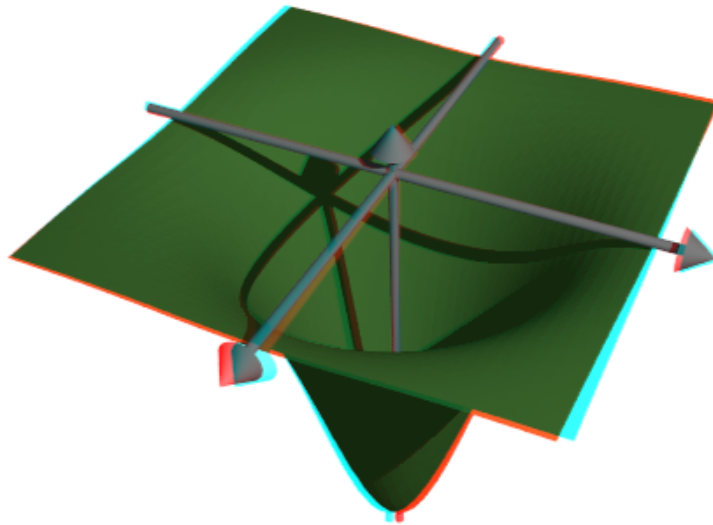
```

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 gabungan povstart() dan povend().

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



Mendefinisikan Objek sendiri

Antarmuka povray Euler berisi banyak objek. Namun Anda tidak dibatasi pada hal ini. Anda dapat membuat objek sendiri, yang menggabungkan objek lain, atau merupakan objek yang benar-benar 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 titik asal.

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

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

Ini torus pertama kami.

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

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

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

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

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

Sekarang kita tempatkan objek-objek tersebut ke dalam sebuah adegan. Untuk tampilannya 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)) ); ...
```

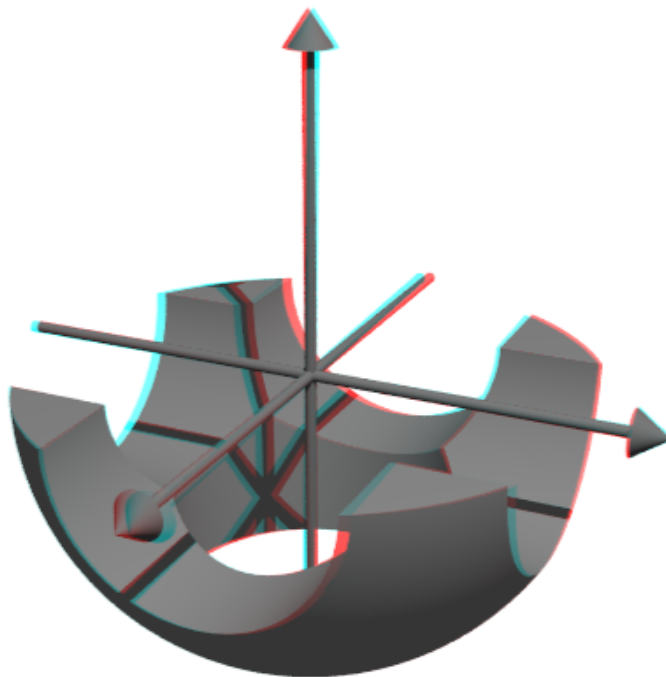
```
>povend();
```

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

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

jika ada yang tidak berfungsi. Ini akan membuat jendela Povray tetap terbuka.

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



Berikut adalah contoh yang lebih rinci. Kami menyelesaikan

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

dan menunjukkan titik-titik yang memungkinkan dan optimum 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 punya solusinya.

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

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

Ya, itu sudah ada.

Selanjutnya, kita mendefinisikan dua objek. Yang pertama adalah bidang

$$a \cdot x \leq b$$

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

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

Kemudian kita mendefinisikan perpotongan semua setengah ruang 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
```

Sekarang kita dapat merencanakan adegannya.

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

Berikut ini adalah lingkaran di sekitar optimal.

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

Dan kesalahan ke arah 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=5°)); ...
>povend();
```

Contoh lain

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

Lihat: Contoh/ Dandelin Spheres

Lihat: Contoh/ Donat Math

Lihat: Contoh/ Trefoil Knot

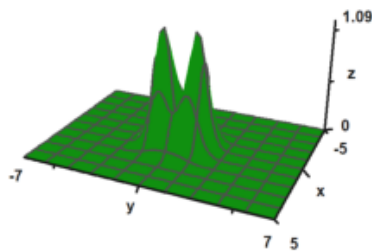
Lihat: Contoh/ Optimisasi dengan Scaling Afinitas

Latihan Soal

1. 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 $\text{scale}=[1,2], \text{xmin}=-5, \text{xmax}=5, \text{ymin}=-7, \text{ymax}=7, \text{frame}=3$.

2. Gambarlah grafik dari fungsi

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

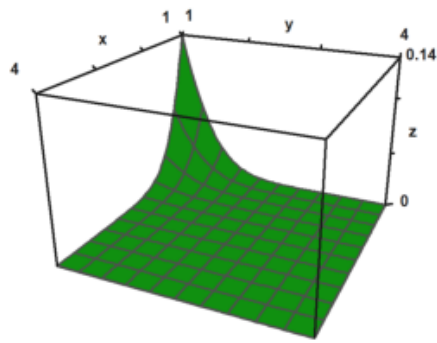
dengan syarat :

$$1 \leq x \leq 4$$

dan

$$1 \leq y \leq 4$$

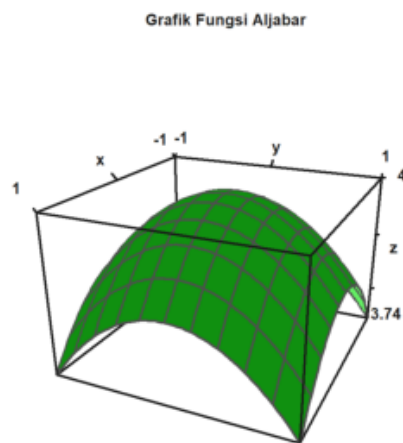
```
>function n(x,y) := E^(-(x^2+y^2));  
>plot3d("n", a=1, b=4, c=1, d=4) :
```

3. Gambarlah grafik dari fungsi berikut.

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

```
>plot3d("(16-x^2-y^2)^(1/2)",>user, ...
>title= "Grafik Fungsi Aljabar"):
```



Penjelasan :
misalkan

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

dan perhatikan bahwa

$$z \geq 0$$

Jika kedua ruas dikuadratkan dan sederhanakan, akan diperoleh 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)

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

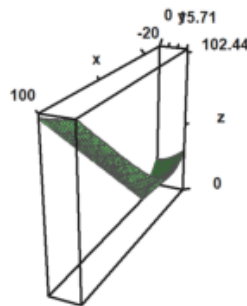
(berupa lingkaran dengan pusat(0,0) dan jari-jari 4)
 -bidang XOZ(y=0)

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

(berupa lingkaran dengan pusat(0,0) dan jari-jari 4)
 4. Gambarlah grafik berikut

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

```
>function g(x,y) &= sqrt(x^2+2*y^2);
>plot3d("g");
>plot3d("g",-20,100,0,5*pi):
```

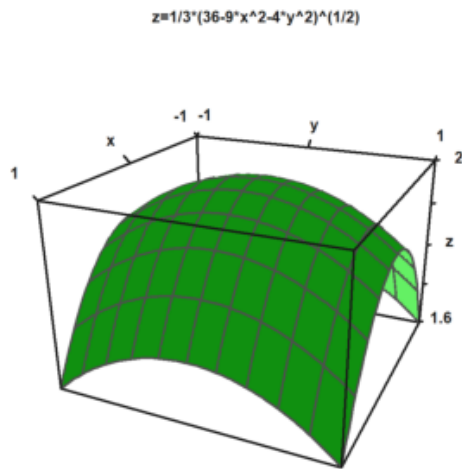


Dari gambar grafik di atas dapat dilihat bahwa secara default gambar plot menggunakan sumbu x dengan rentang nilai dari -20 sampai dengan 100.

5. 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)