JPL09 - 3d graphics

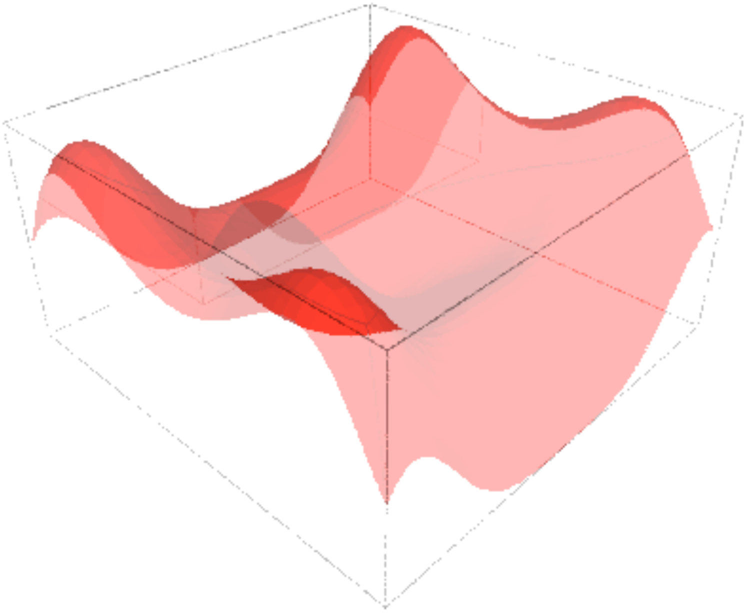
3D Graphics

The plot3d command draws 3d plots of functions of two variables.

```
var('x,y')
P = plot3d(sin(x^2)-cos(y^2), (x,-2,2), (y,-2,2), color='red',
opacity=0.8, mesh=True)
P.show()
```

In case you ever have any problem with the above (say java isn't setup in your web browser), the option viewer='tachyon' will use a raytracer to draw a png image. That *always works* no matter what.

```
P.show(viewer='tachyon')
```



You can pass several options to the plot3d command, e.g., mesh, opacity, color, etc. Try modifying each of the parameters below and seeing what happens.

```
plot3d(sin(x^2)-cos(y^2), (x,-2,2), (y,-2,2), color='blue',
opacity=0.8, mesh=True,
          aspect_ratio=[1,1,1])
```

When drawing 3d plots there are two other critically important functions: **parametric_plot3d** and **implicit_plot3d.** We illustrate each below. In each case, try changing parameters and see what happens. The reference manual has dozens of examples of parametric 3d plots.

```
var('u,v')
f_x = sin(u) / (sqrt(2) + sin(v))
f_y = sin(u) / (sqrt(2) + cos(v))
f_z = cos(u) / (1 + sqrt(2))
parametric_plot3d([f_x, f_y, f_z], (u, -pi, pi), (v, -pi, pi),
                  frame=False, color="green", plot_points=[50,50])
```

```
var('x,y,z')
T = RDF(golden_ratio)
p = 2 - (cos(x + T*y) + cos(x - T*y) + cos(y + T*z) + cos(y - T*z) +
cos(z - T*x) + cos(z + T*x))
r = 4.77
implicit_plot3d(p, (x, -r, r), (y, -r, r), (z, -r, r),
plot_points=60)
```

Just like with 2d graphics, you can combine together multiple 3d plots:

```
var('x,y')
```

```
P1 = plot3d(sin(x*y), (x,-2,2), (y,-2,2), opacity=0.5)
P2 = plot3d(cos(x*y), (x,-2,2), (y,-2,2), opacity=0.7, color='red')
P3 = sphere((0,0,1))
P1 + P2 + P3
```

```
S = sphere(size=.5, color='yellow')
from sage.plot.plot3d.shapes import Cone
S += Cone(.5, .5, color='red').translate(0,0,.3)
S += sphere((.45,-.1,.15), size=.1, color='white') +
sphere((.51,-.1,.17), size=.05, color='black')
S += sphere((.45, .1,.15),size=.1, color='white') + sphere((.51,
.1,.17), size=.05, color='black')
S += sphere((.5,0,-.2),size=.1, color='yellow')
f(x,y) = exp(x/5)*cos(y)
P = plot3d(f,(-5,5),(-5,5), color='red')
cape_man = P.scale(.2) + S.translate(1,0,0)
cape_man.show(aspect_ratio=[1,1,1], frame=False)
```

Finally, here is a list of 3d primitives. Play around with plotting them:

- arrow3d
- line3d
- tetrahedron, cube, octahedron, dodecahedron, icosahedron
- sphere
- list_plot3d
- text3d
- polygon3d
- point3d

A bunch of random spheres:

```
S
=[sphere((random(),random(),random()),0.1*random(),color=hue(random()))
 for _ in range(30)]
show(sum(S), aspect_ratio=[1,1,1])
```

For example, here is a 3d random walk.

```
p = [0,0,0]
v = [p]
for i in range(200):
   p = copy(p)
   for j in range(3):
      p[j] += random()-0.5
   v.append(p)
line3d(v)
```