

MAT 305: Mathematical Computing

3d plots

John Perry

University of Southern Mississippi

Fall 2013

Outline

- ① 3d objects
- ② 3d plots of functions
- ③ Summary

3d objects

3d plots of
functions

Summary

Outline

- 1 3d objects
- 2 3d plots of functions
- 3 Summary

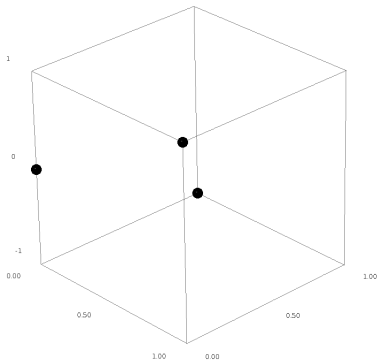
The `point3d()` command

`point3d((x,y,z), options)` where

- (x,y,z) is a 3-tuple (the location in \mathbb{R}^3 of this point)
 - can send list of points $[(x_1,y_1,z_1), (x_2,y_2,z_2), \dots]$
- *options* include
 - `rgbcolor`
 - `size` (*not* `pointsize`; default is 5)
 - `opacity` (more on this later)

Example

```
sage: point3d([(0,0,0), (0,1,-1), (1,0,1)],  
             rgbcolor=(0,0,0), size=10)
```



grab image and rotate!

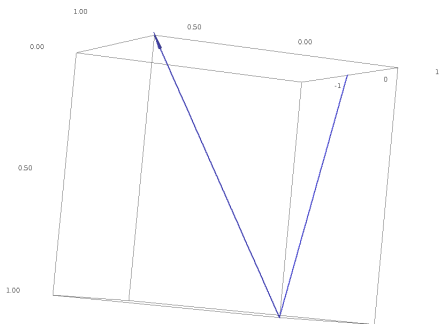
The `line3d()` command

`line3d`($[(x_1, y_1, z_1), (x_2, y_2, z_2), \dots]$, *options*) where

- $[(x_1, y_1, z_1), (x_2, y_2, z_2), \dots]$ is a list of *at least two* points
 - more than two points? consecutive lines
- *options* include
 - `rgbcolor`
 - `thickness`
 - `arrow_head=True` for arrow on final point
 - `opacity` (more on this later)

Example

```
sage: line3d([(0,0,0), (0,1,-1), (1,0,1)],  
             thickness=2, arrow_head=True)
```



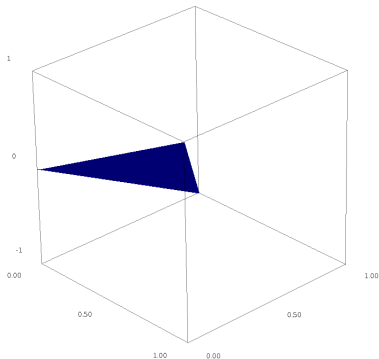
The `polygon3d()` command

`polygon3d([$(x_1, y_1, z_1), (x_2, y_2, z_2), \dots$], options)` where

- [$(x_1, y_1, z_1), (x_2, y_2, z_2), \dots$] is a list of *at least two* points
 - fewer than 2? nothing drawn
- *options* include
 - `rgbcolor`
 - `opacity` (more on this later)

Example

```
sage: polygon3d([(0,0,0), (0,1,-1), (1,0,1)])
```



controls whether you can “see through” the object

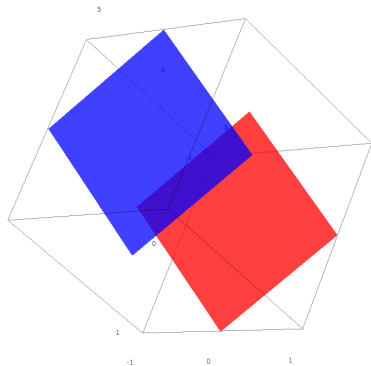
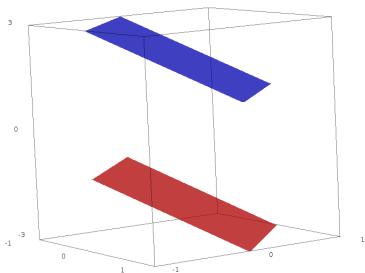
- ranges from 0 to 1
- 0: completely translucent (invisible)
- 1: completely opaque
- useful when combining many objects

Example: two parallel planes

```
sage: p1 = polygon3d([(1,0,1), (0,1,1), (-1,0,3), (0,-1,3)],  
                    opacity=0.75)
```

```
sage: p2 = polygon3d([(1,0,-3), (0,1,-3), (-1,0,-1), (0,-1,-1)],  
                    rgbcolor=(1,0,0), opacity=0.75)
```

```
sage: p1 + p2
```



3d objects

3d plots of
functions

Summary

Outline

① 3d objects

② 3d plots of functions

③ Summary

Variables

- “standard” 2d: y depends on x
 - can define otherwise if necessary
 - Q as functions of t

- “standard” 3d: z depends on x, y
 - can define otherwise
 - x given (unless changed); must define at least y

The `plot3d()` command

`plot3d(f(x,y), (x, xmin, xmax), (y, ymin, ymax),
options)`

where

- $f(x,y)$ is a function of x and y
- *options* include
 - `adaptive=True` for a better-looking graph (slower)
 - `mesh=True` for mesh grid lines
 - `dots=True` to show dots at grid points
 - `color`
 - `opacity`

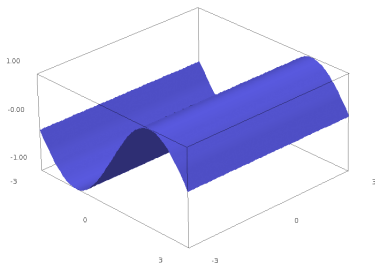
Example 1

3d objects

3d plots of
functions

Summary

```
sage: plot3d(sin(x), (x,-3,3), (y,-3,3))
```



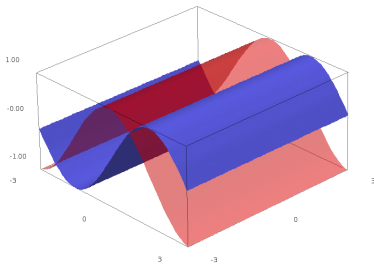
grab image and rotate!

Example 2: color and opacity

```
sage: p1 = plot3d(sin(x), (x,-3,3), (y,-3,3))
```

```
sage: p2 = plot3d(cos(x), (x,-3,3), (y,-3,3),  
                rgbcolor=(1,0,0), opacity=0.5)
```

```
sage: p1+p2
```



Something marginally useful

Plot $z = \sin x \cos y$ and the tangent plane at
 $(x_0, y_0, z_0) = \left(\frac{\pi}{6}, \frac{\pi}{3}, \frac{1}{4}\right)$. Make the plane red and translucent.

3d objects

3d plots of
functions

Summary

Something marginally useful

3d objects

3d plots of
functions

Summary

Plot $z = \sin x \cos y$ and the tangent plane at $(x_0, y_0, z_0) = \left(\frac{\pi}{6}, \frac{\pi}{3}, \frac{1}{4}\right)$. Make the plane red and translucent.

Tangent plane is

$$z = f_x(x_0, y_0) \cdot (x - x_0) + f_y(x_0, y_0) \cdot (y - y_0) + z_0$$

where

- f_x, f_y , are “partial derivatives”

$$f_x = \frac{\partial}{\partial x} f(x, y): x \text{ variable, } y \text{ “constant”}$$

- (x_0, y_0, z_0) point

Something marginally useful

Plot $z = \sin x \cos y$ and the tangent plane at $(x, y, z) = \left(\frac{\pi}{6}, \frac{\pi}{3}, \frac{1}{4}\right)$.
Make the plane red and translucent.

3d objects

3d plots of
functions

Summary

Something marginally useful

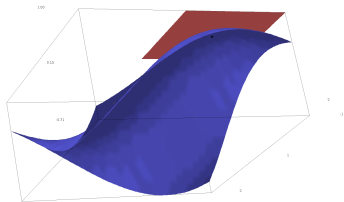
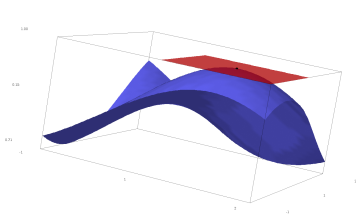
Plot $z = \sin x \cos y$ and the tangent plane at $(x, y, z) = \left(\frac{\pi}{6}, \frac{\pi}{3}, \frac{1}{4}\right)$.
Make the plane red and translucent.

```
sage: f(x,y) = sin(x)*cos(y)
sage: p1 = plot3d(f, (x,-pi/4,3*pi/4), (y,-pi/4,3*pi/4))
sage: dfx = diff(f,x)
sage: dfy = diff(f,y)
sage: a=pi/2; b = 0
sage: tanplane = dfx(a,b)*(x-a) + dfy(a,b)*(y-b) + f(a,b)
sage: p2 = plot3d(tanplane, (x,pi/4,3*pi/4), (y,-pi/4,pi/4),
                 rgbcolor=(1,0,0), opacity=0.75)
sage: p3 = point3d((a,b,f(a,b)),rgbcolor=(0,0,0),size=10)
sage: p1+p2+p3
```

f_x
 f_y

...and you get...

The graduating seagull!



Let's add a normal vector

```
sage: fz(x,y,z) = f(x,y) - z
```

```
sage: fgrad = fz.gradient()
```

(Think about why I had to subtract z)

```
sage: fgrad
```

```
(x,y,z) |--> (cos(x)*cos(y), -sin(x)*sin(y), -1)
```

```
sage: c = f(a,b)
```

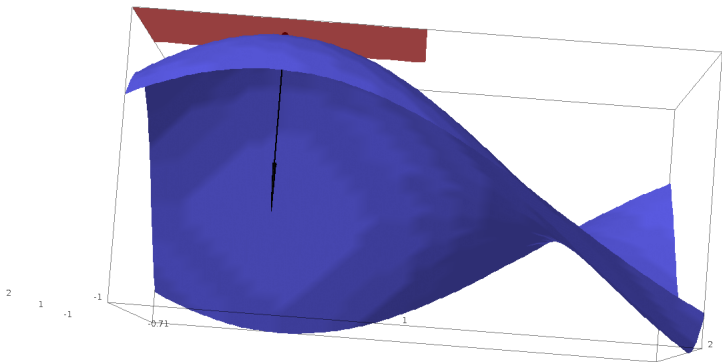
```
sage: dx, dy, dz = fgrad(a,b,c)
```

```
sage: nvec = line3d([(a,b,c), (a+dx,b+dy,c+dz)],  
                    rgbcolor=(0,0,0), thickness=2,  
                    arrow_head=True)
```

```
sage: p1 + p2 + p3 + nvec
```

...and you get...

Result



3d objects

3d plots of
functions

Summary

Outline

① 3d objects

② 3d plots of functions

③ Summary

Summary

- Sage offers many ways to plot 3d objects and functions
 - plots can be rotated
 - images can be saved

- adjusting opacity allows one to see through an object