

Stack Overflow is a community of 4.7 million programmers, just like you, helping each other.

Join them; it only takes a minute:

[Sign up](#)

Join the Stack Overflow community to:

Ask programming
questions

Answer and help
your peers

Get recognized for your
expertise

plot rotation of a parametrised algebraic surface

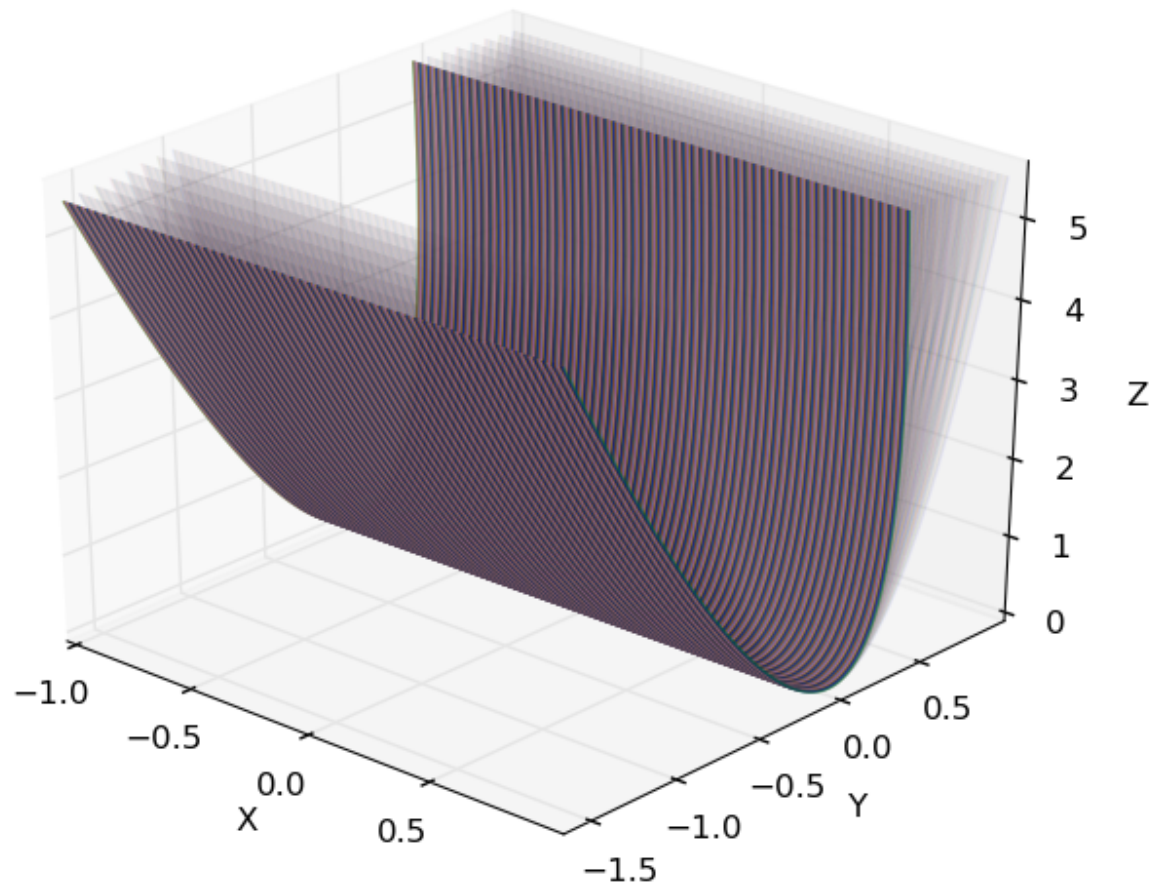
I'd like to plot a rotated parabolic cylinder. Like holding one point of the plane of symmetry with two fingers and let this plane rotate about this point.

There is the [rotation matrix](#) (for the y-axis).

And the parametrised surface is:

(u, v, u^2)

I've already plotted this surface for rotations about the x-axis:



Below is the modified(!) code to plot the rotations about the y-axis.

```
from math import cos,sin,pi
import matplotlib.pyplot as plt
from matplotlib import cm

from mpl_toolkits.mplot3d import Axes3D
```

```

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.auto_scale_xyz([0, 500], [0, 500], [0, 0.15])

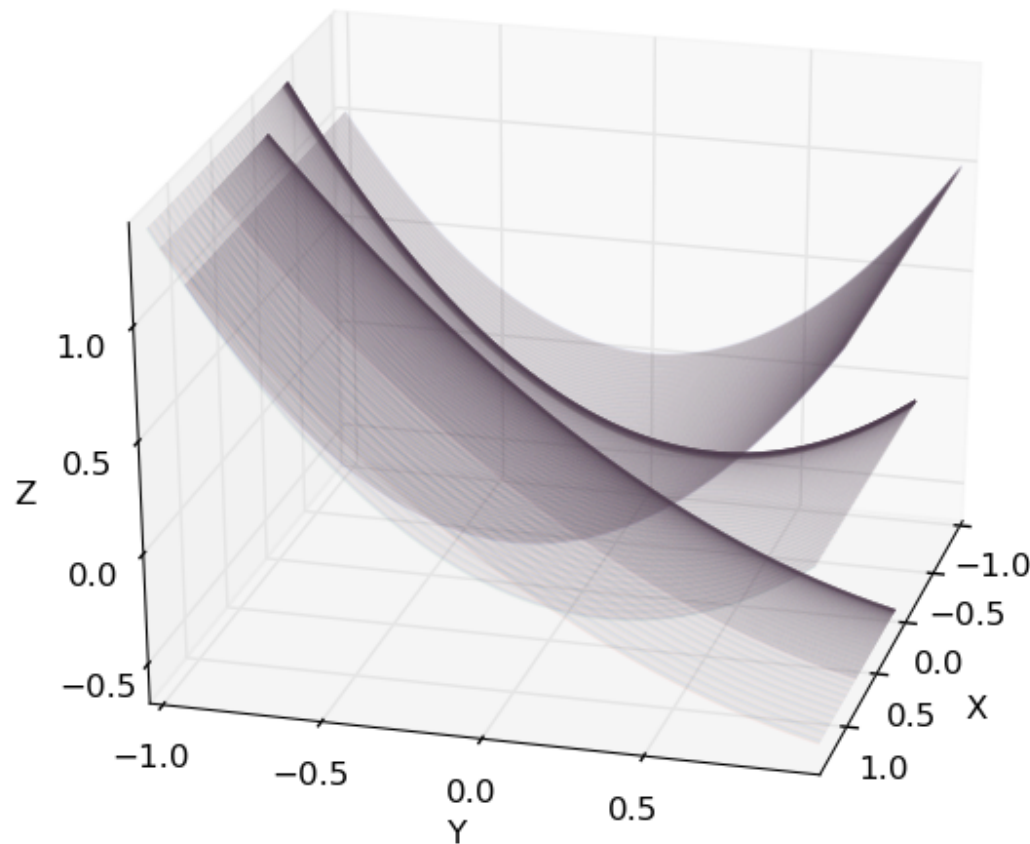
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')

Us = np.arange(-1, 1, 0.005)
Vs = np.arange(-1, 1, 0.005)

for ch in [0.2*i*pi for i in range(3)]:
    Xs = Us*cos(ch) + sin(ch)*(Us**2)
    Ys = Vs
    Xs, Ys = np.meshgrid(Xs, Ys)
    Zs = -Us*sin(ch) + cos(ch)*(Us**2)
    Axes3D.plot3D(ax,Xs,Ys,Zs, alpha=0.05)

```

And it doesn't work. It produces nonsense. This is the output i get:



The operations are identical, so i really can't see why it doesn't work. Can anyone tell me what i'm doing wrong?

I need to make this work in order to just look at the way algebraic surfaces transform when subjected to certain rotations. The code above is a crudely simplified snippet of what i am currently debugging. So the answer i am hoping for regards this particular code.

[python](#) [3d](#) [matplotlib](#)



2 Answers

Here is another version which uses your explicit rotation equations:

```
import matplotlib as mpl
mpl.use('TkAgg')
from math import cos,sin,pi
import numpy as np
from mpl_toolkits.mplot3d import axes3d
import matplotlib.pyplot as plt

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.auto_scale_xyz([0, 500], [0, 500], [0, 0.15])

ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')

U = np.arange(-1, 1, 0.005)
V = np.arange(-1, 1, 0.005)
def animate():
    for ch in [0.2*i*pi for i in range(3)]:
        Us, Vs = np.meshgrid(U, V)
        Xs = Us*cos(ch) + sin(ch)*(Us**2)
        Ys = Vs
        Zs = -Us*sin(ch) + cos(ch)*(Us**2)
        tmp = ax.plot_surface(Xs,Ys,Zs, alpha=0.05)
        fig.canvas.draw()
        tmp.remove()

win = fig.canvas.manager.window
fig.canvas.manager.window.after(100, animate)
plt.show()
```

answered Sep 18 '12 at 13:26

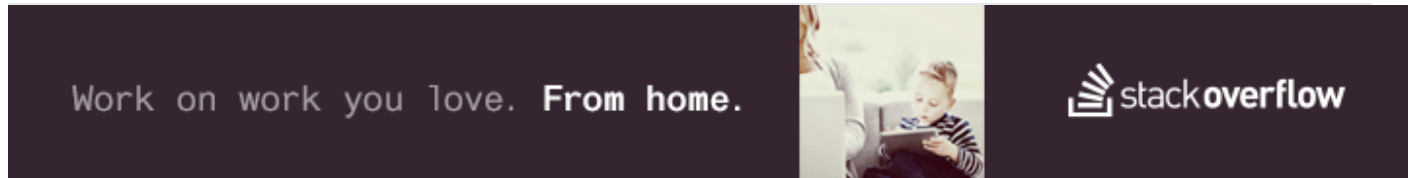


matplotlib.use() must be called *before* pylab,matplotlib.pyplot, or matplotlib.backends... if warn:
 warnings.warn(_use_error_msg) Traceback (most recent call last): File "<stdin>", line 1, in <module> File
 "/usr/lib/python2.7/dist-packages/spyderlib/widgets/externalshell/sitecustomize.py", line 493, in runfile
 execfile(filename, namespace) File "/home/michael/Courses/Introduction to Programming/src/untitled0.py",
 line 96, in <module> fig.canvas.manager.window.after(100, animate) AttributeError: 'QMainWindow' object
 has no attribute 'after' – [Michael](#) Sep 18 '12 at 16:20

this is the error message i get from this code – [Michael](#) Sep 18 '12 at 16:21

Your Spyder IDE is interfering with valid Python code. Open a terminal and run the script there: python
 untitled0.py . – [unutbu](#) Sep 18 '12 at 17:39

ok, but this is the rotation i managed to execute. my question above is for the rotational axis perpendicular
 to this one. now i included the false output i get into my first post. – [Michael](#) Sep 19 '12 at 11:24



The [rotate_axes3d_demo](#) does it with [ax.view_init](#) this way:

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import axes3d
plt.ion()

sin = np.sin
cos = np.cos
pi = np.pi

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.auto_scale_xyz([0, 500], [0, 500], [0, 0.15])

ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')

Us, Vs = np.meshgrid(np.linspace(-1, 1, 200), np.linspace(-1, 1, 200))
ax.plot_surface(Us, Vs, Us**2)
```

```
for angle in range(0, 360):  
    ax.view_init(angle, 30)  
    plt.draw()
```

answered Sep 18 '12 at 12:40



unutbu

366k

46

660

782

this code only produces one uniform parabolic cylinder. also edited my initial post. – [Michael](#) Sep 18 '12 at 12:49

That's odd; I get a rotating surface. – [unutbu](#) Sep 18 '12 at 12:51
