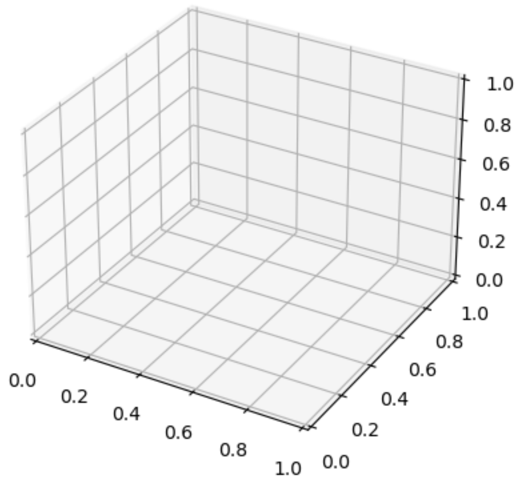
In [1]:
```
#mpplot3d is used for plottig 3d graphs
from mpl_toolkits import mplot3d
```

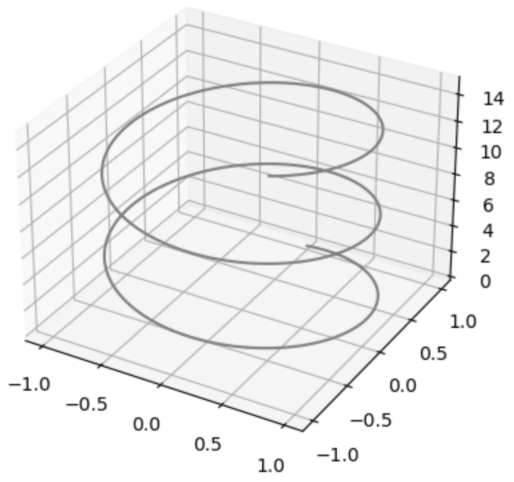
In [2]:
```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:
```
fig=plt.figure()
ax=plt.axes(projection="3d")
```



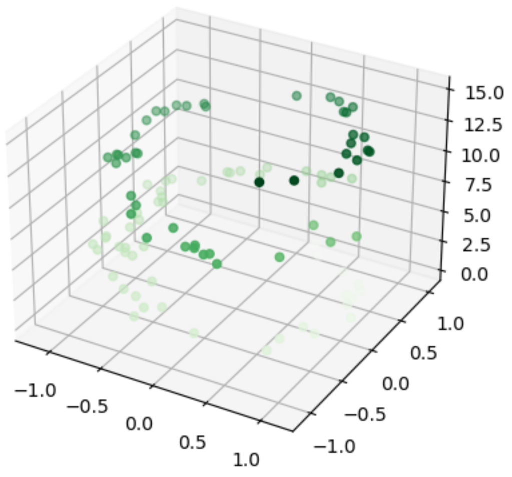
In [4]:
```
ax=plt.axes(projection="3d")
zline=np.linspace(0,15,1000)
xline=np.sin(zline)
yline=np.cos(zline)
ax.plot3D(xline,yline,zline,"gray")
```

Out[4]:
```
[<mpl_toolkits.mplot3d.art3d.Line3D at 0x12b789a3fa0>]
```



In [5]:
```
ax=plt.axes(projection="3d")
zdata=15*np.random.random(100)
xdata=np.sin(zdata)+0.1*np.random.randn(100)
ydata=np.cos(zdata)+0.1*np.random.randn(100)
ax.scatter3D(xdata,ydata,zdata,c=zdata,cmap='Greens')
```

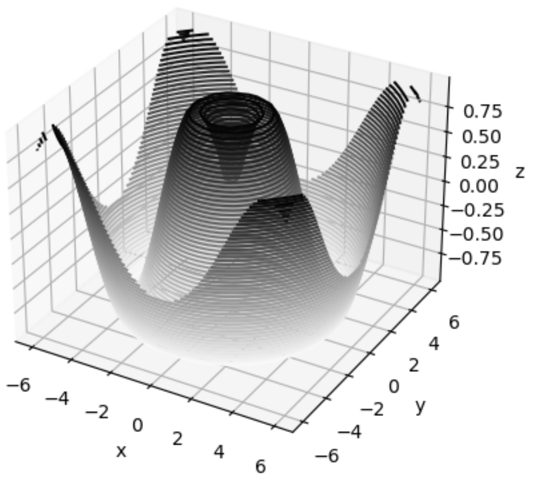
Out[5]:
```
<mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x12b78a41850>
```



In [6]:
```
#3d contour plots
def f(x,y):
    return np.sin(np.sqrt(x**2+y**2))
x=np.linspace(-6,6,30)
y=np.linspace(-6,6,30)
X,Y=np.meshgrid(x,y)
Z=f(X,Y)
```

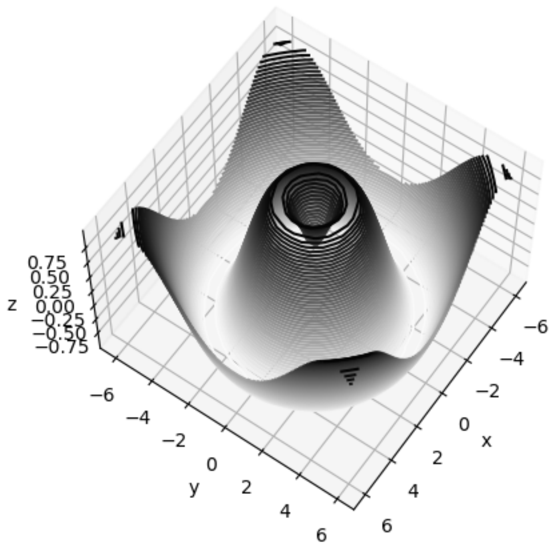
In [7]:
```
fig=plt.figure()
ax=plt.axes(projection='3d')
ax.contour3D(X,Y,Z,50,cmap="binary")
ax.set_xlabel("x")
ax.set_ylabel("y")
ax.set_zlabel("z")
```

Out[7]:
```
Text(0.5, 0, 'z')
```



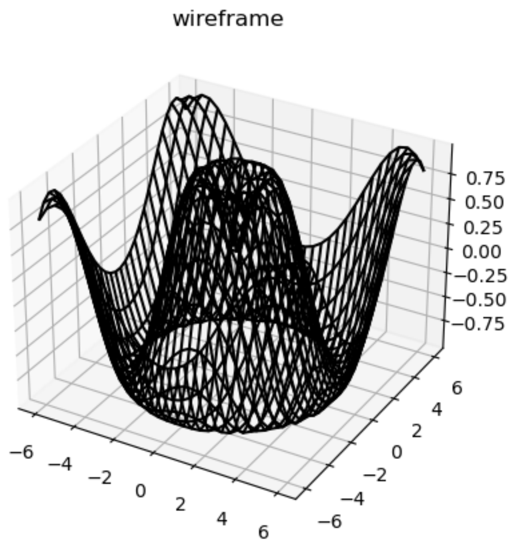
In [8]:
```
#viweing using viewinit
ax.view_init(60,35)#60 rotaion in x-y plane,35 degrees rotated counter clockwise in z plane
fig
```

Out[8]:



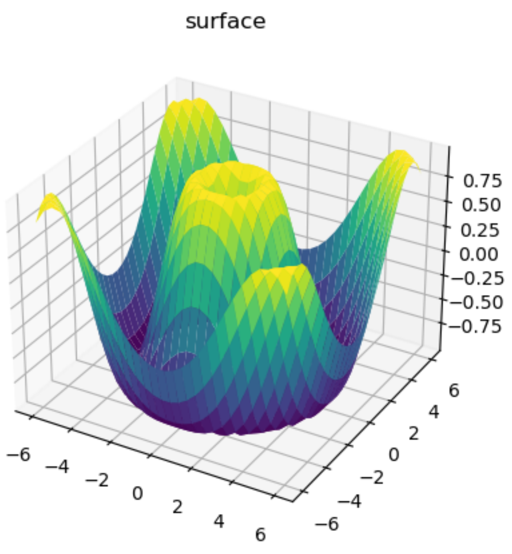
In [9]:
```
#wireframe
fig=plt.figure()
ax=plt.axes(projection="3d")
ax.plot_wireframe(X,Y,Z,color="black")
ax.set_title("wireframe")
```

Out[9]:
```
Text(0.5, 0.92, 'wireframe')
```



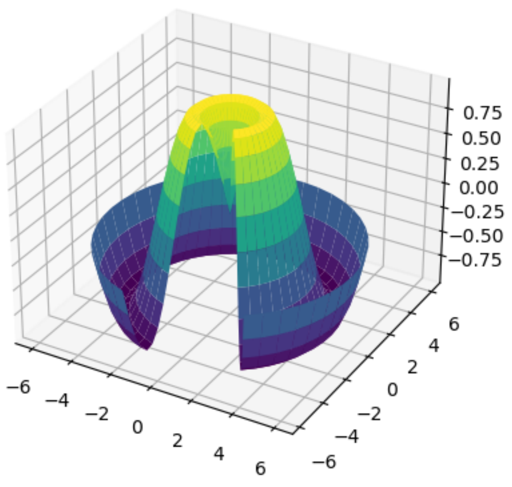
In [10]:
```
#each fac eof wireframe is a filled polygon and we can use color map for even better visualisation
ax=plt.axes(projection="3d")
ax.plot_surface(X,Y,Z,rstride=1,cstride=1,cmap="viridis",edgecolor="none")
ax.set_title("surface")
```

Out[10]:
```
Text(0.5, 0.92, 'surface')
```

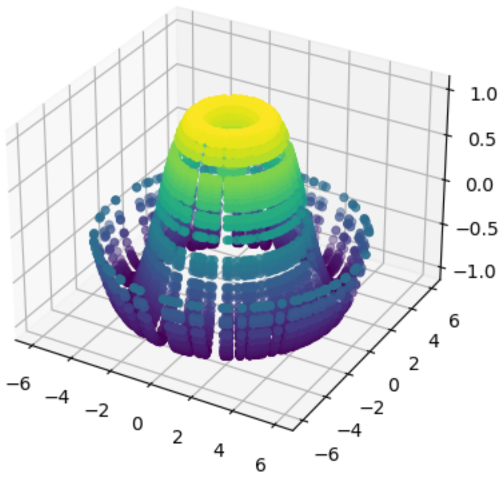


In [11]:
```
r=np.linspace(0,6,20)
theta=np.linspace(-0.9*np.pi,0.8*np.pi,40)
r,theta=np.meshgrid(r,theta)
X=r*np.sin(theta)
Y=r*np.cos(theta)
Z=f(X,Y)
ax=plt.axes(projection="3d")
ax.plot_surface(X,Y,Z,rstride=1,cstride=1,cmap="viridis",edgecolor="none")
```

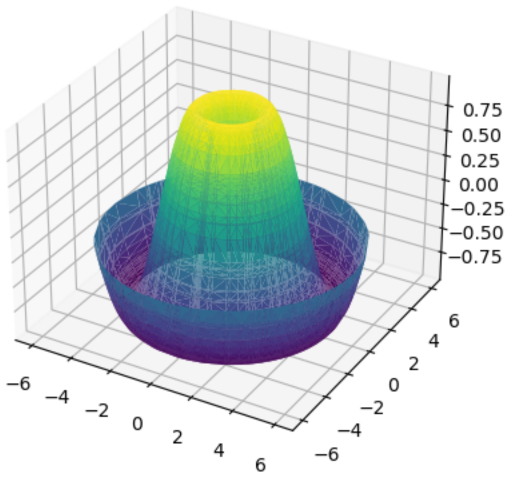
Out[11]:
```
<mpl_toolkits.mplot3d.art3d.Poly3DCollection at 0x12b791482e0>
```



In [13]:
```
theta=2*np.pi*np.random.random(100)
r=6*np.random.random(100)
r,theta=np.meshgrid(r,theta)
X=np.ravel(r*np.sin(theta))#ravel is used for converting multi dimensional array to 1D arrays
Y=np.ravel(r*np.cos(theta))
Z=f(X,Y)
ax=plt.axes(projection="3d")
ax.scatter(X,Y,Z,c=Z,cmap="viridis",linewidth=0.5);
```

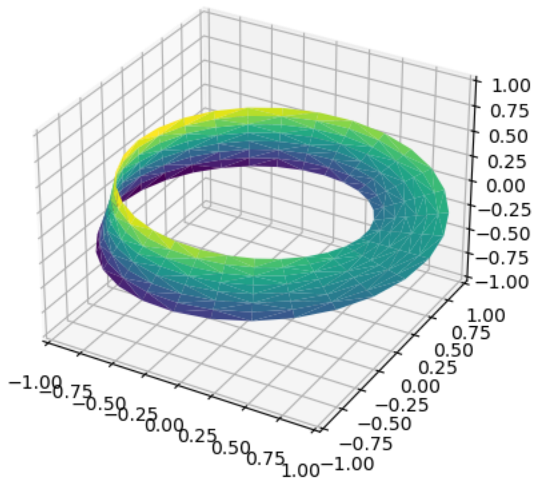


In [14]:
```
ax = plt.axes(projection='3d')
ax.plot_trisurf(X,Y,Z,cmap='viridis', edgecolor='none')#plotting triangular plots using plot_trisurf
```



In [17]:
```
#mobius loop
theta=np.linspace(0,2*np.pi, 30)#strip has 360 angle
w=np.linspace(-0.25, 0.25, 8)#w represents width of strip
w,theta=np.meshgrid(w,theta)
phi=0.5*theta#twist in the loop
r=1+w*np.cos(phi)
x=np.ravel(r*np.cos(theta))
y=np.ravel(r*np.sin(theta))
z=np.ravel(w*np.sin(phi))
from matplotlib.tri import Triangulation
tri=Triangulation(np.ravel(w),np.ravel(theta))#triangluation of the two parameters we defined
ax=plt.axes(projection="3d")
ax.plot_trisurf(x,y,z,triangles=tri.triangles,cmap='viridis',linewidths=0.2)#triangles argument is given
ax.set_xlim(-1,1)#setting limits on axis
ax.set_ylim(-1,1)
ax.set_zlim(-1,1)
```

Out[17]:
```
(-1.0, 1.0)
```


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00