MAT150 - Summer 2023

Phd. Anabela Romina Turlione
Digipen, Bilbao

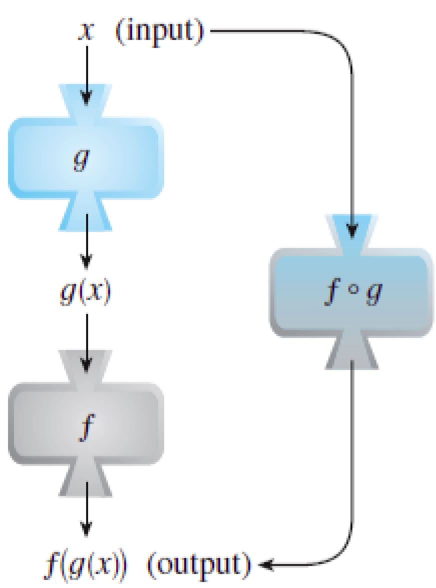
Domain and range of composite functions

Composition

Given two functions $f(x)$ and $g(x)$, the **composite function** $f \circ g$ is defined by

$$(f \circ g)(x) = f(g(x)) \tag{1}$$

- Domain of $f \circ g$: it is the set of all x in the domain of g such that $g(x)$ is in the domain of f.



Exercise 1

If $f(x) = \sqrt{x}$ and $g(x) = \sqrt{2-x}$, find each function and its domain.

(a) $f \circ g$

(b) $g \circ f$

(c) $f \circ f$

(d) $g \circ g$

Solution:

(a) $f \circ g = \sqrt[4]{2-x}$, Domain: $(-\infty, 2]$

(b) $g \circ f = \sqrt{2-\sqrt{x}}$, Domain: $[0, 4]$

(c) $f \circ f = \sqrt[4]{x}$, Domain: $[0, \infty)$

(d) $g \circ g = \sqrt{2-\sqrt{2-x}}$, Domain: $[-2, 2]$

- In the study of the domain of a composite function, first of all, simplify the function as much as possible.

Exercise 2: Find the domain of f, g and $f\big(g(x)\big)$

$$f(x) = \frac{x-2}{x-1}, \quad g(x) = \frac{x+2}{x+1} \tag{2}$$

Exercise 3:

Find the domain of the following curves.

1. $y = \sqrt{2x-1}$

1. $y = \sqrt{3-t} - \sqrt{2-t}$

1. $y = \frac{\sqrt{1-x^2}}{x-2}$

1. $y = \frac{\sqrt{x^2-4}}{\ln(x-1)}$

1. $y = \arccos\left(\frac{x}{3}\right)$

1. $y = \frac{\ln(x-1)}{\ln x - 1}$

1. $y = \sec(3x)$

1. $y = \tan(2x-\pi)$

1. $ \left\lbrace
 \begin{array}{ll}
 x=\frac{1}{t-1} \\
 y=\ln(t)
 \end{array} \right.$

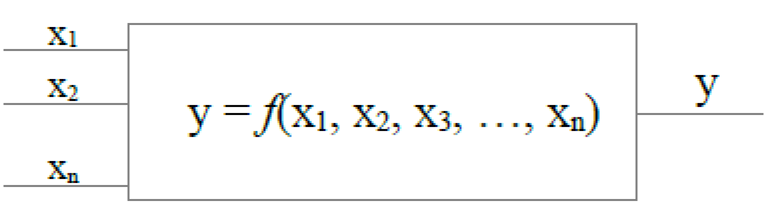
1. $(y+2)^2 + (x-1)^2 = 1$

1. $y^2 - 4y + 1 = 2x$

1. $x^2 - 2x + 3y^2 + 6y + 1 = 0$

Surfaces in R3

A multi-variable function is rule that assigns to some entry(ies) an output.

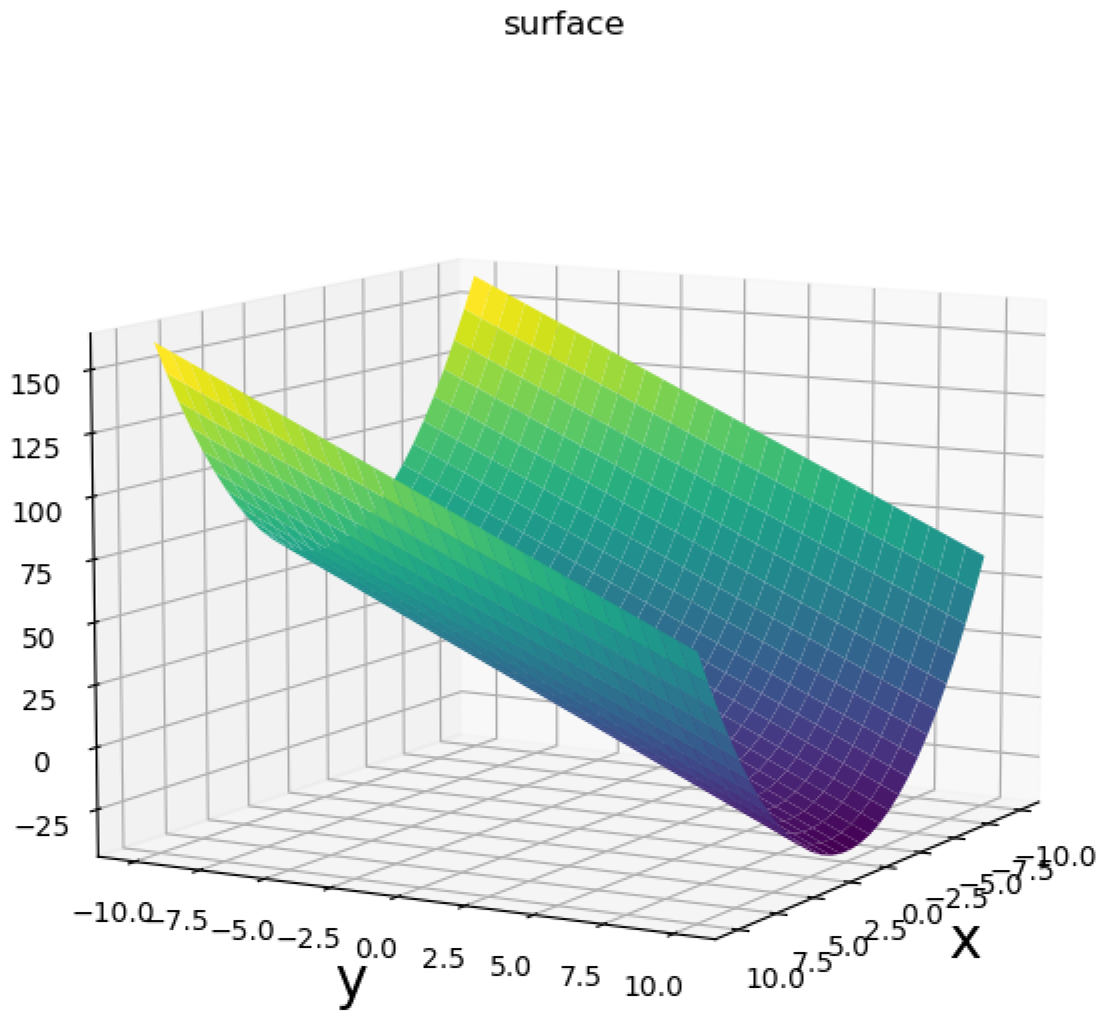


We are going to study the particular case in which $y \in \Re$, then $f : x_1, x_2, \ldots, x_n \in \Re^n \rightarrow \Re$

Example: $z = f(x, y) = x^2 - 5y + 10$

In [29]:
```
import matplotlib.pyplot as plt
import numpy as np

def plot_z1():
    x = np.linspace(-10,10,30)
    y = np.linspace(-10,10,30)
    X, Y = np.meshgrid(x,y)


    Z = X**2-5*Y+10


    ax = plt.axes(projection='3d')
    ax.plot_surface(X, Y, Z, rstride=1, cstride=1,
                    cmap='viridis', edgecolor='none')

    plt.xlabel('x', fontsize=20)
    plt.ylabel('y', fontsize=20)

    plt.rcParams['figure.figsize'] = [7, 7]
    ax.set_title('surface');
    ax.view_init(10, 30)
```

In [30]:
```
plot_z1()
```



In the same way than in 2D we can define some curves that are not functions by means of the implicit equation $f(x, y) = 0$, in 3D we can use $f(x, y, z) = 0$.

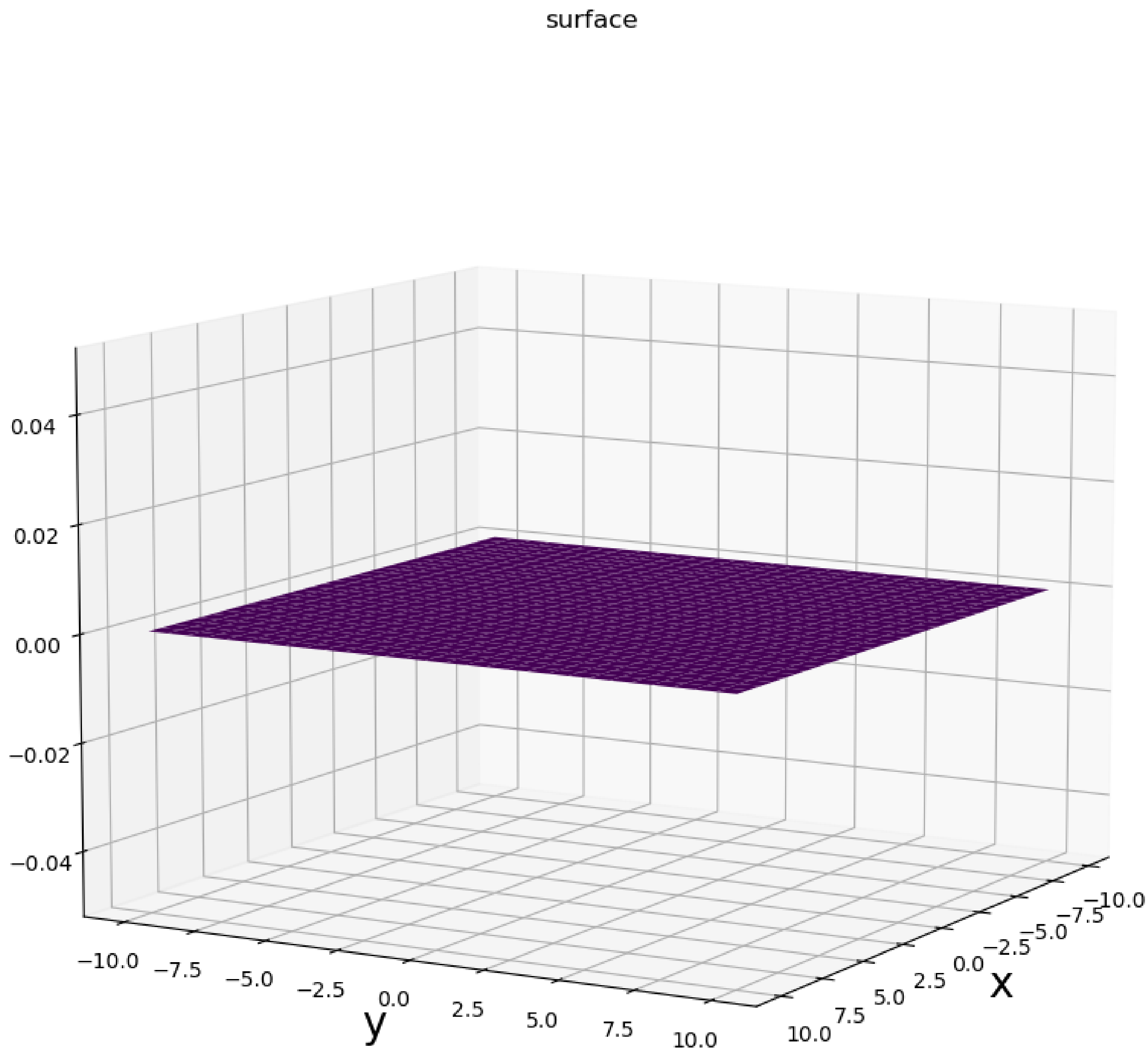
Basic Surfaces

Planes: $A(x - x_0) + B(y - y_0) + C(z - z_0) = 0$, $A,\ B,\ C$ real constants, $x,\ y\ and\ z \in \Re$

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

def plot_z2(A,B,C,x0,z0,y0):
    x = np.linspace(-10,10,30)
    y = np.linspace(-10,10,30)
    X, Y = np.meshgrid(x,y)

    Z = z0-(A*(X-x0)+B*(Y-y0))/C

    ax = plt.axes(projection='3d')
    ax.plot_surface(X, Y, Z, rstride=1, cstride=1,
                    cmap='viridis', edgecolor='none')

    plt.xlabel('x', fontsize=20)
    plt.ylabel('y', fontsize=20)

    plt.rcParams['figure.figsize'] = [10, 10]
```

```
    ax.set_title('surface');
    ax.view_init(10, 30)
```

In [33]:
```
A,B,C,x0,y0,z0 =0,0,1,0,0,0

plot_z2(A,B,C,x0,z0,y0)
```



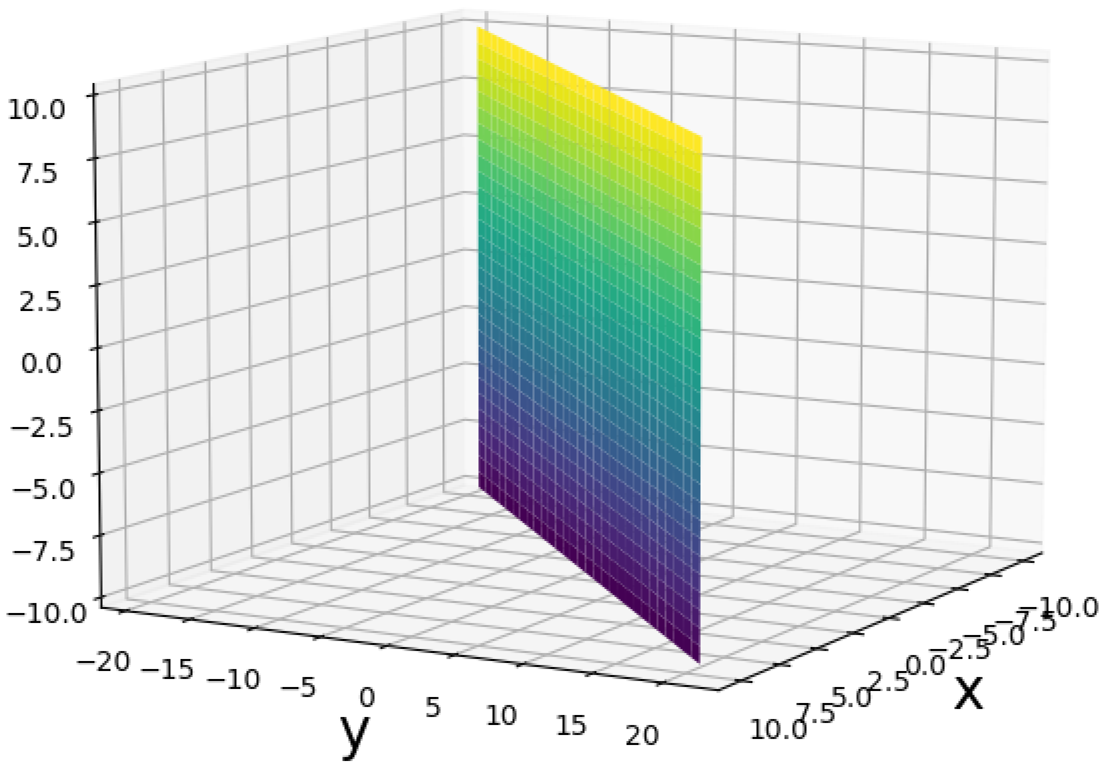
In [12]:
```
import matplotlib.pyplot as plt
import numpy as np

def plot_z3(A,B):
    x = np.linspace(-10,10,30)
    z = np.linspace(-10,10,30)
    X, Z = np.meshgrid(x,z)

    Y = A*X+B

    ax = plt.axes(projection='3d')
    ax.plot_surface(X, Y, Z, rstride=1, cstride=1,
                    cmap='viridis', edgecolor='none')
```

```
    plt.xlabel('x', fontsize=20)
    plt.ylabel('y', fontsize=20)

    plt.rcParams['figure.figsize'] = [7, 7]
    ax.set_title('surface');
    ax.view_init(10, 30)
```

$y = 2x + 1$

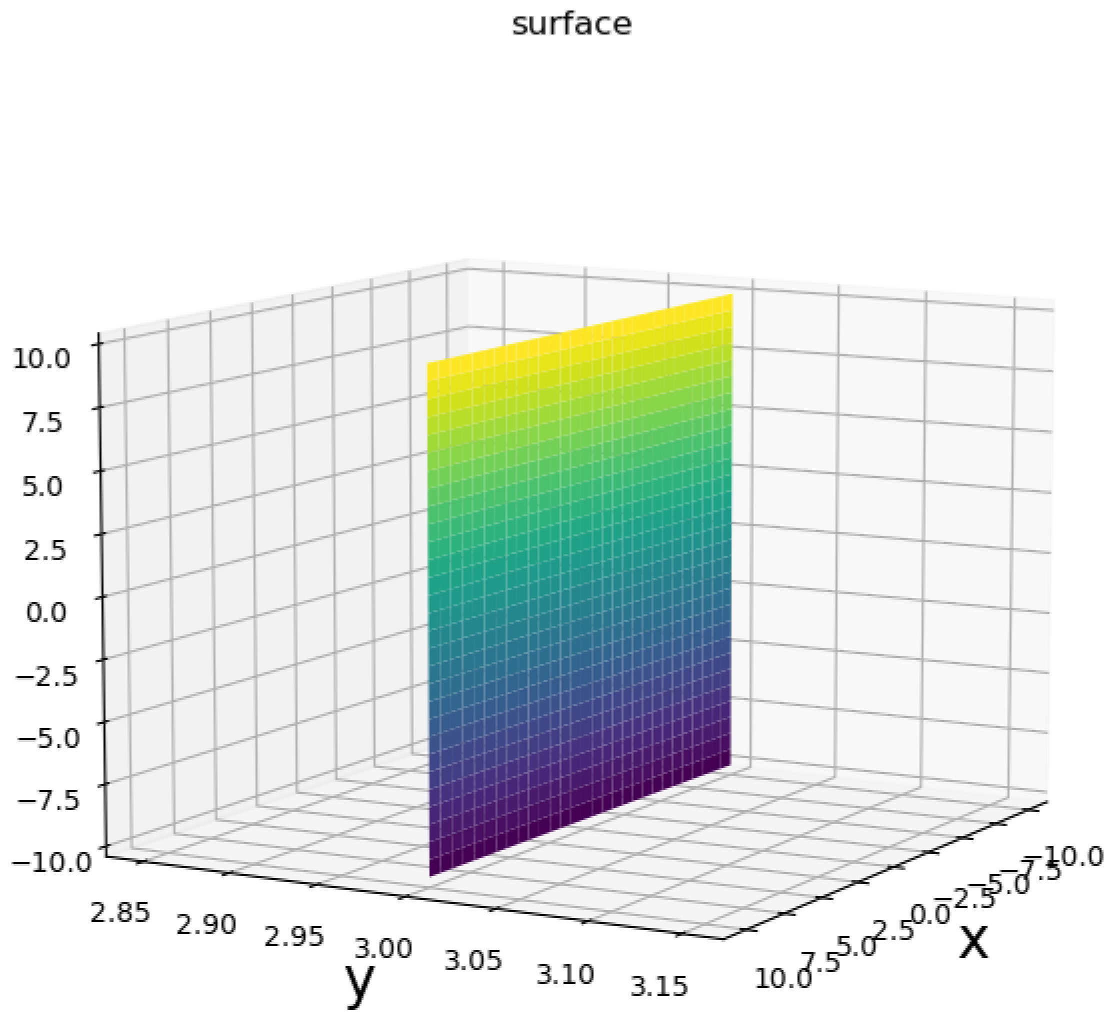
In [14]:
```
plot_z3(2,1)
```

surface



$y = 3$

In [15]:
```
plot_z3(0,3)
```



Cylinders

- In general, $f(x, y) = 0$ (parallel to z), $g(x, z) = 0$ (parallel to y) or $h(y, z) = 0$ (parallel to x).
- According to their **section area** the cylinders are named "circular cylinders, elliptical cylinders, parabolic cylinders, etc.
- Common to all of them, is that their intersection with planes parallel to their axis of symmetry will return in lines.

In [16]:

```
import matplotlib.pyplot as plt
import numpy as np

def plot_z4(r1,r2):

    z = np.linspace(0,10,50)
    theta = np.linspace(0,2*np.pi,50)


    THETA, Z = np.meshgrid(theta, z)

    X=r1*np.cos(THETA)
    Y=r2*np.sin(THETA)
```

```
    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')
    ax.plot_surface(X, Y, Z,alpha=0.5)

    plt.xlabel('x', fontsize=20)
    plt.ylabel('y', fontsize=20)

    plt.rcParams['figure.figsize'] = [7, 7]
#     ax.set_title('surface');
    ax.view_init(40, 40)
    ax.set_xlim([-3.5,3.5])
    ax.set_ylim([-3.5,3.5])
```

Circular cylinder

$$(x-x_0)^2+(y-y_0)^2=R^2 \tag{3}$$

Example:

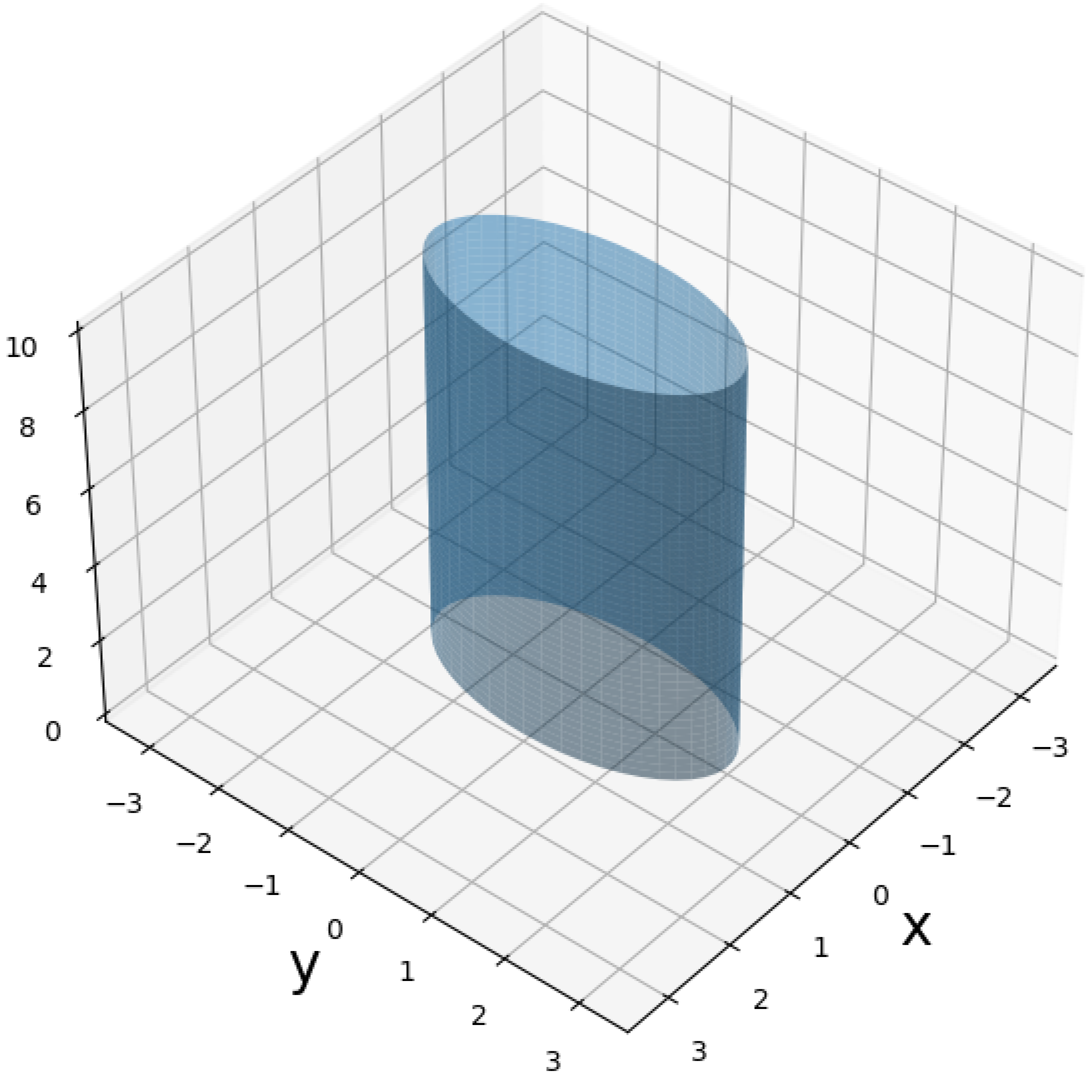
$$x^2+y^2=1 \tag{4}$$

Parametric equation

We use cylindrical coordinates:

$$\begin{cases} x = r\cos(\phi) \\ y = r\sin(\phi) \quad ,\ \phi\ in\ [0,2\pi],\ t\ in\ [0,L] \\ z = t \end{cases} \tag{5}$$

In [35]:
```
plot_z4(1,2)
```



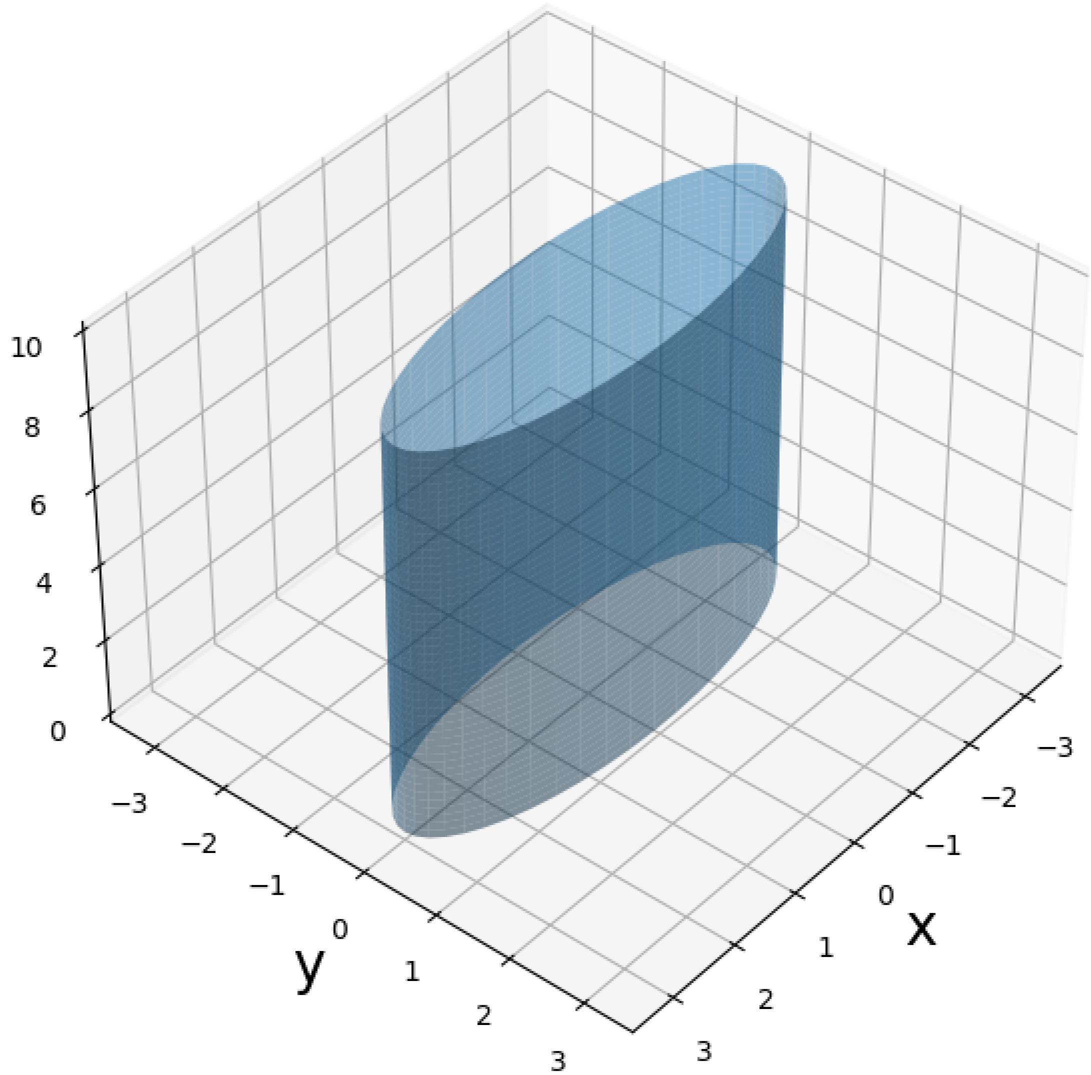
Elliptical cylinder

$$\frac{(x-x_0)^2}{a^2} + \frac{(y-y_0)^2}{b^2} = 1 \tag{6}$$

Example:

$$\frac{x^2}{9} + y^2 = 1 \tag{7}$$

In [18]:
```
plot_z4(3,1)
```



Parabolic cylinder

$$(x - x_0)^2 = a(y - y_0) \tag{8}$$

Example

$$x^2 = y \tag{9}$$

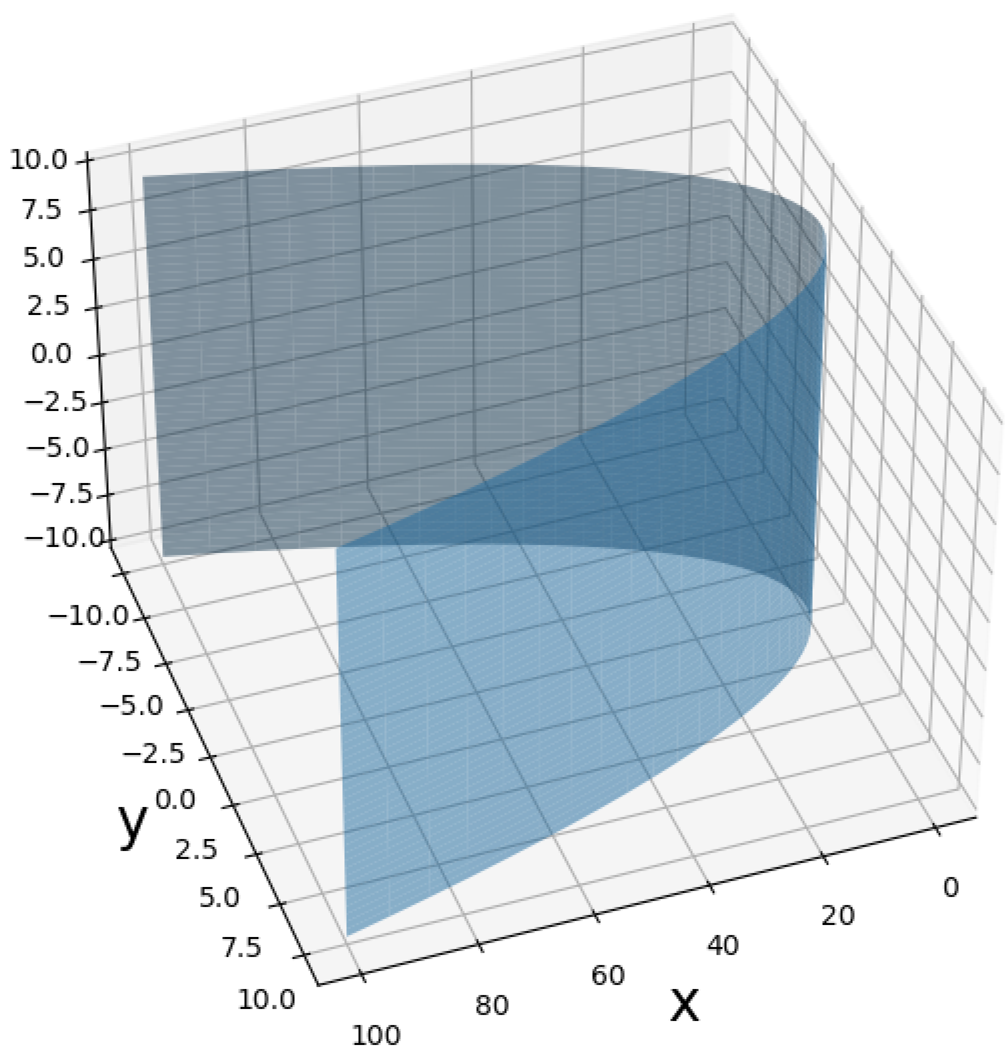
Parametric equation

$$\begin{cases} x = t^2 \\ y = t \quad ,t\ in\ [0, +\infty),\ u\ in\ [0, L] \\ z = u \end{cases} \tag{10}$$

In [19]:
```
import matplotlib.pyplot as plt
import numpy as np

def plot_z5():
    r = 1

    z = np.linspace(-10,10,50)
    t = np.linspace(-10,10,50)
```

```
    T, Z = np.meshgrid(t, z)

    X=T**2
    Y=T

    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')
    ax.plot_surface(X, Y, Z,alpha=0.5)

    plt.xlabel('x', fontsize=20)
    plt.ylabel('y', fontsize=20)

    plt.rcParams['figure.figsize'] = [7, 7]
#     ax.set_title('surface');
    ax.view_init(40, 70)
```

In [20]:
```
plot_z5()
```



Sphere

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = r^2 \tag{11}$$

Example:

$$x^2 + y^2 + z^2 = 1 \tag{12}$$

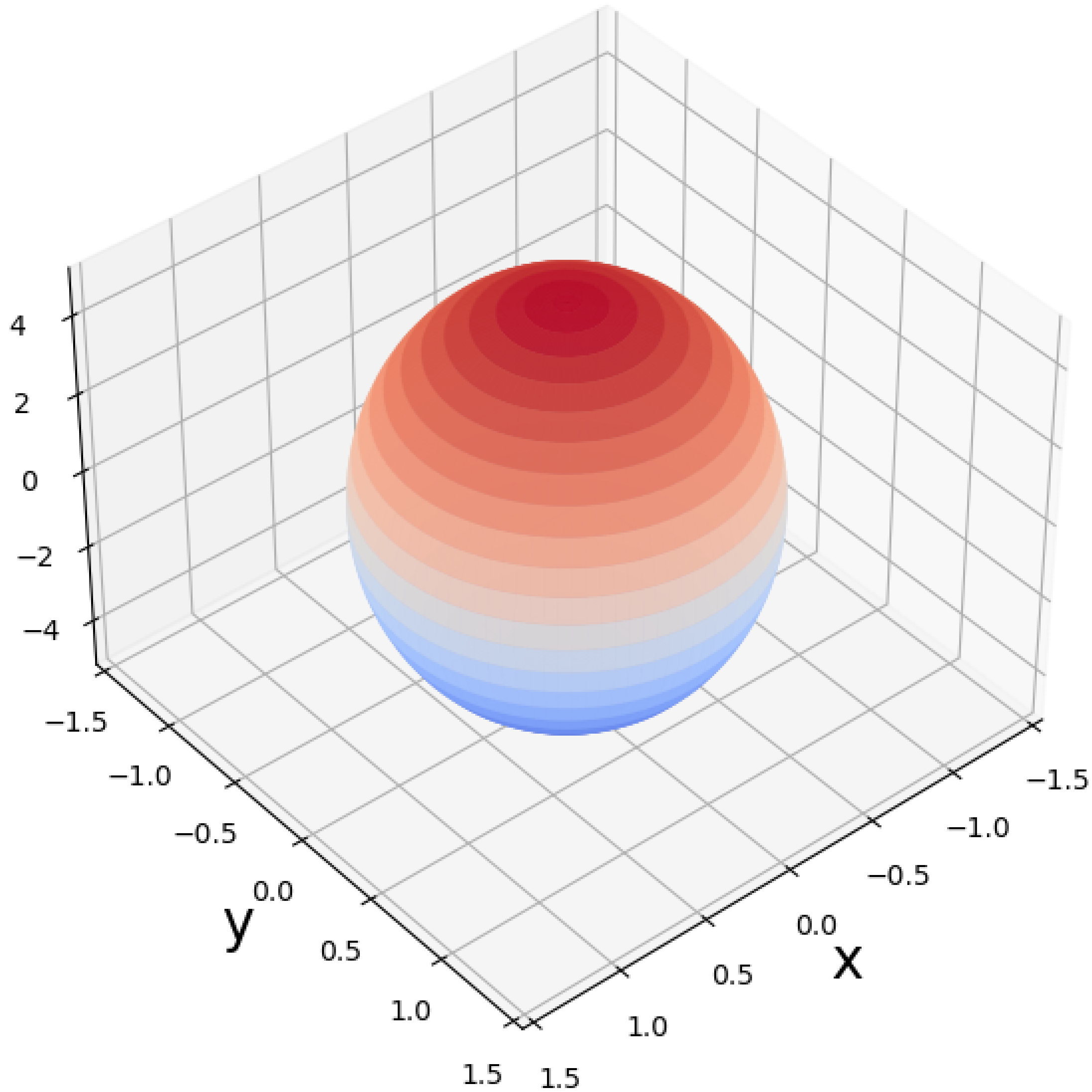
Parametric equation

We use spherical coordinates:

$$\begin{cases} x = r\sin(\theta)\cos(\phi) \\ y = r\sin(\theta)\sin(\phi) \\ z = r\cos(\theta) \end{cases}, \ \theta \in [0,\pi], \ \phi \, in \, [0,2\pi] \tag{13}$$

In [25]:
```
import matplotlib.pyplot as plt
import numpy as np
from matplotlib import cm

def plot_z6(a,b,c):


    phi = np.linspace(-2*np.pi,2*np.pi,50)
    theta = np.linspace(-np.pi,np.pi,50)


    THETA, PHI = np.meshgrid(theta, phi)

    X=a*np.sin(THETA)*np.cos(PHI)
    Y=b*np.sin(THETA)*np.sin(PHI)
    Z=c*np.cos(THETA)

    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')
    ax.plot_surface(X, Y, Z, cmap=cm.coolwarm, alpha=0.5)

    plt.xlabel('x', fontsize=20)
    plt.ylabel('y', fontsize=20)

    plt.rcParams['figure.figsize'] = [7, 7]
#     ax.set_title('surface');
    ax.view_init(40, 50)
    ax.set_xlim([-1.5,1.5])
    ax.set_ylim([-1.5,1.5])
```

In [39]:
```
plot_z6(1,1,5)
```



Paraboloide

$$\left(\frac{x-x_0}{a}\right)^2+\left(\frac{y-y_0}{b}\right)^2=z-z_0 \tag{14}$$

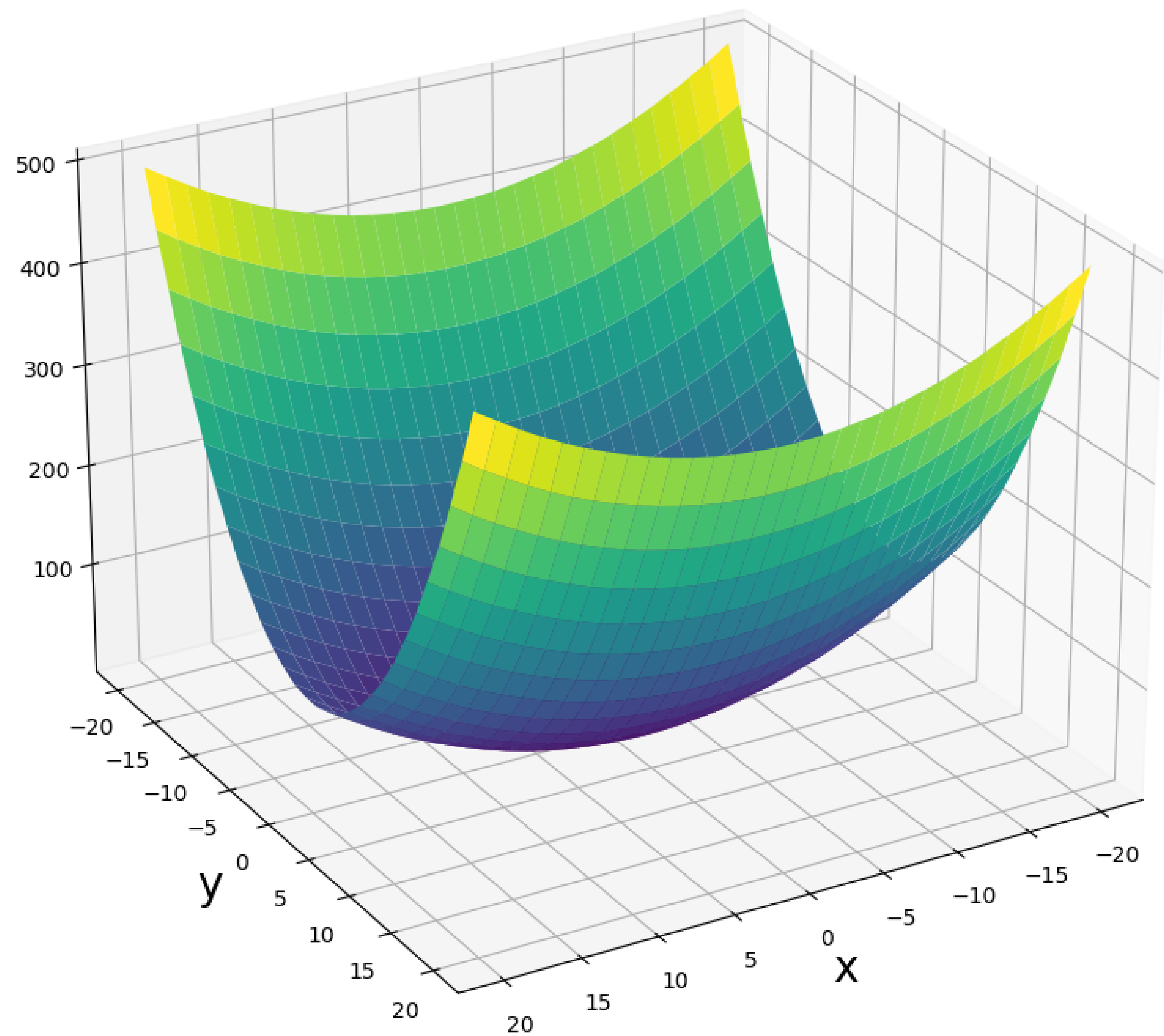
Parametric equation

$$\begin{cases} x=t \\ y=u \\ z=\left(\frac{t-t_0}{a}\right)^2+\left(\frac{u-u_0}{b}\right)^2 \end{cases} \tag{15}$$

In [25]:
```
import matplotlib.pyplot as plt
import numpy as np

def plot_z5(a,b,c=None):
    x = np.linspace(-20,20,30)
    y = np.linspace(-20,20,30)
    X, Y = np.meshgrid(x,y)
    Z = (X/a)**2+(Y/b)**2

    if c:
```

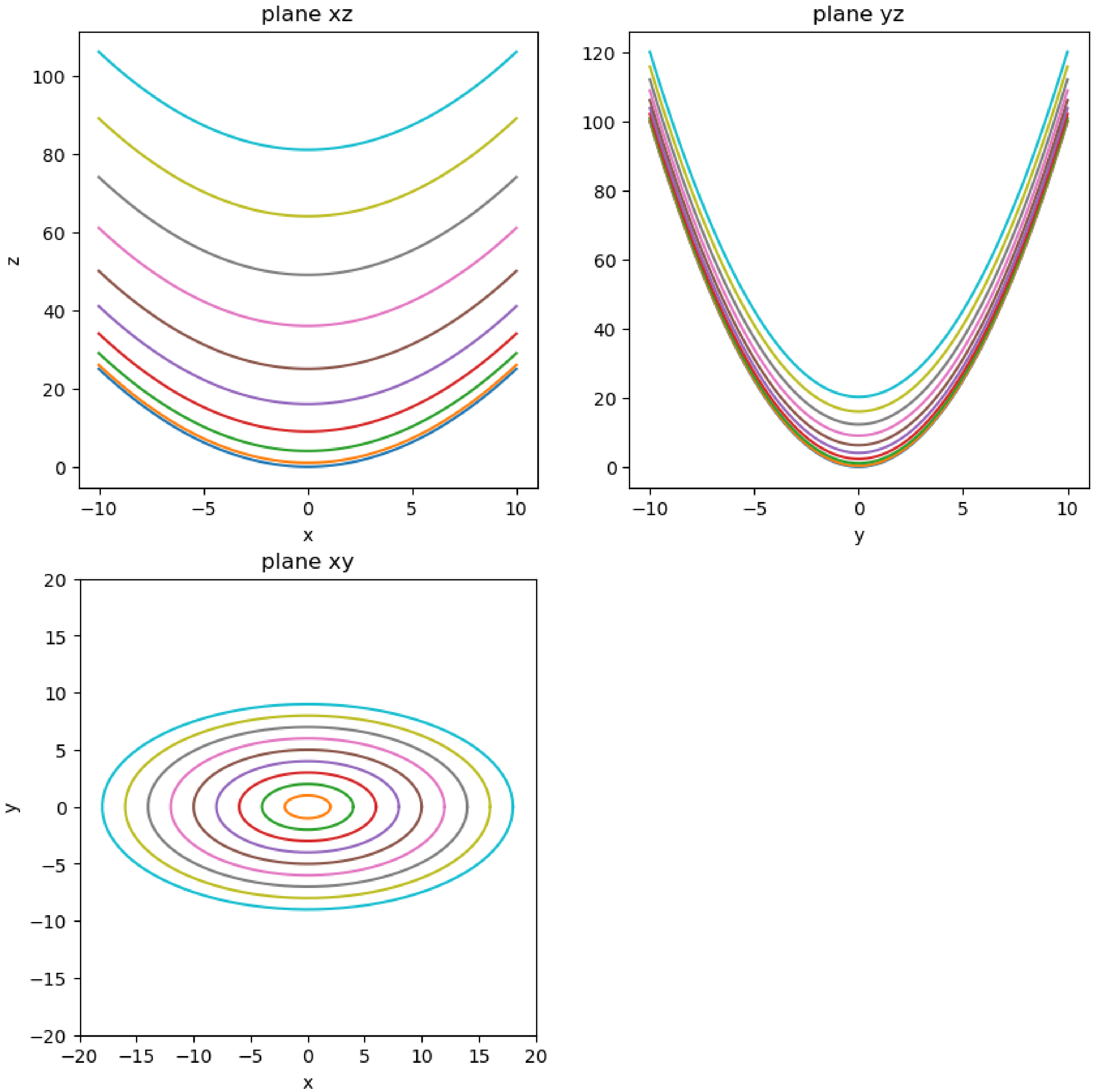
```
        fig,ax=plt.subplots(1,1)
        ax.contour(Z)
    else:

        ax = plt.axes(projection='3d')
        ax.plot_surface(X, Y, Z, rstride=1, cstride=1,
                        cmap='viridis', edgecolor='none')
        plt.xlabel('x', fontsize=20)
        plt.ylabel('y', fontsize=20)

        plt.rcParams['figure.figsize'] = [10, 10]
        ax.view_init(25, 60)

#################################################################
def plot_z5_projections(a,b):
    fig = plt.figure(figsize=plt.figaspect(1.))

    # First subplot
    ax = fig.add_subplot(2, 2, 1)
    ax.set_title('plane xz')
    x = np.linspace(-10,10,100)
    for y in range(0,10):
        ax.plot(x,(x/a)**2+(y/b)**2)
        ax.set_xlabel('x', fontsize=10)
        ax.set_ylabel('z', fontsize=10)

    # Third subplot
    ax = fig.add_subplot(2, 2, 2)
    ax.set_title('plane yz')

    y = np.linspace(-10,10,100)
    for x in range(0,10):
        ax.plot(y,(x/a)**2+(y/b)**2)
        ax.set_xlabel('y', fontsize=10)

    # Fourth subplot
    ax = fig.add_subplot(2, 2, 3)
    ax.set_title('plane xy')
    t = np.linspace(0,2*np.pi,100)
    for z in range(0,10):
        x=z*a*np.cos(t)
        y=z*b*np.sin(t)
        ax.plot(x,y)
        ax.set_xlabel('x', fontsize=10)
        ax.set_ylabel('y', fontsize=10)
        ax.set_xlim([-20,20])
        ax.set_ylim([-20,20])
        ax.set_aspect('equal')

#     # Fith subplot
#     ax = fig.add_subplot(2, 3, 4, projection='3d')
#     x = np.linspace(-10,10,30)
#     y = np.linspace(-10,10,30)
#     X, Y = np.meshgrid(x,y)
#     Z = (X/a)**2+(Y/b)**2

#     ax.plot_surface(X, Y, Z, rstride=1, cstride=1,
#                            linewidth=0, antialiased=False)
#     plt.xlabel('x', fontsize=10)
#     plt.ylabel('y', fontsize=10)

#     plt.rcParams['figure.figsize'] = [10, 10]
#     ax.view_init(25, 60)

    plt.show()
```

In [26]:
```
a,b=2,1
x0,y0,z0=0,0,0
plot_z5(a,b)
```



Projections

In [27]:
```
plot_z5_projections(a,b)
```



A cone

$$\left(\frac{x-x_0}{a}\right)^2+\left(\frac{y-y_0}{b}\right)^2=\left(\frac{z-z_0}{c}\right)^2 \tag{16}$$

Parametric equation

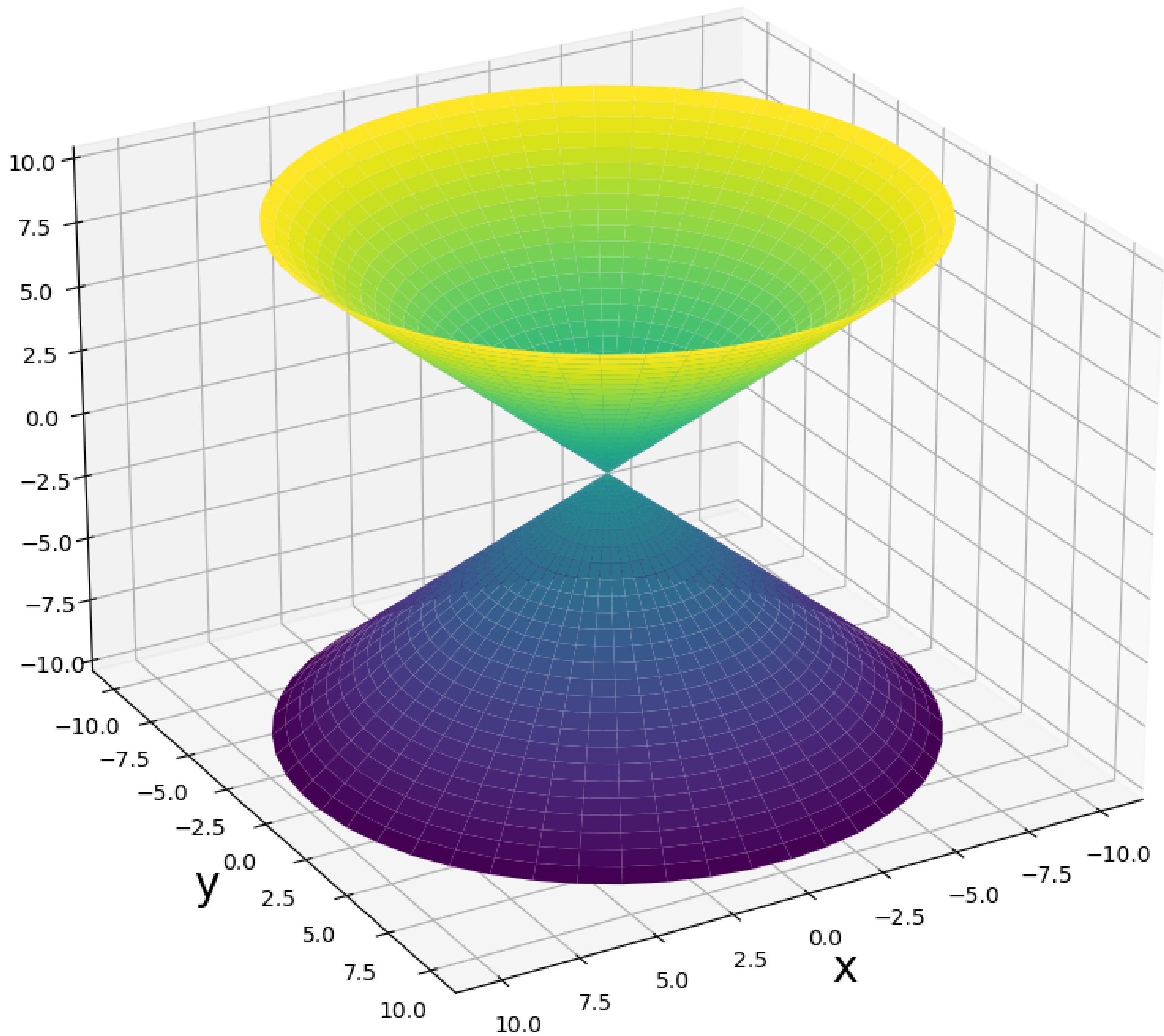
$$\begin{cases} x = at\cos(\theta)/c + x_0 \\ y = at\sin(\theta)/c + y_0 \\ z = t + z_0 \end{cases}, \ \theta \in [0, 2\pi], \ t \in [-L1, L2] \tag{17}$$

In [28]:
```
def plot_z6(a,b,c,x0,y0,z0):
    z = np.linspace(-10,10,50)
    theta = np.linspace(0,2*np.pi,50)
```

```
    THETA, Z = np.meshgrid(theta, z)

    X=a*(Z-z0)*np.cos(THETA)/c+x0
    Y=b*(Z-z0)*np.sin(THETA)/c+y0

    ax = plt.axes(projection='3d')
    ax.plot_surface(X, Y, Z, rstride=1, cstride=1,
                    cmap='viridis', edgecolor='none')
    plt.xlabel('x', fontsize=20)
    plt.ylabel('y', fontsize=20)

    plt.rcParams['figure.figsize'] = [10, 10]
    ax.view_init(25, 60)

#################################################################
def plot_z6_projections(a,b,c,x0,y0,z0):
    x=np.linspace(-10,10,1000)
    c=1

    fig = plt.figure(figsize=plt.figaspect(1.))

    # First subplot
    ax = fig.add_subplot(2, 2, 1)
    ax.set_title('plane xz')
    x = np.linspace(-10,10,100)
    for y in range(0,10):
        ax.plot(x,z0+c*np.sqrt((x-x0)**2/a**2+(y-y0)**2/b**2))
        ax.plot(x,z0-c*np.sqrt((x-x0)**2/a**2+(y-y0)**2/b**2))
        ax.set_xlabel('x', fontsize=10)
        ax.set_ylabel('z', fontsize=10)

    # Second subplot
    ax = fig.add_subplot(2, 2, 2)
    ax.set_title('plane yz')
    y = np.linspace(-10,10,100)
    for x in range(0,10):
        ax.plot(y,z0+c*np.sqrt((x-x0)**2/a**2+(y-y0)**2/b**2))
        ax.plot(y,z0-c*np.sqrt((x-x0)**2/a**2+(y-y0)**2/b**2))
        ax.set_xlabel('x', fontsize=10)
        ax.set_ylabel('z', fontsize=10)

    # Third subplot
    ax = fig.add_subplot(2, 2, 3)
    ax.set_title('plane xy')
    t = np.linspace(0,2*np.pi,100)
    for z in range(0,10):
        x=(z-z0)*a*np.cos(t)/c
        y=(z-z0)*b*np.sin(t)/c
        ax.plot(x,y)
        ax.set_xlabel('x', fontsize=10)
        ax.set_ylabel('y', fontsize=10)
#          ax.set_xlim([-20,20])
#          ax.set_ylim([-20,20])
        ax.set_aspect('equal')
```

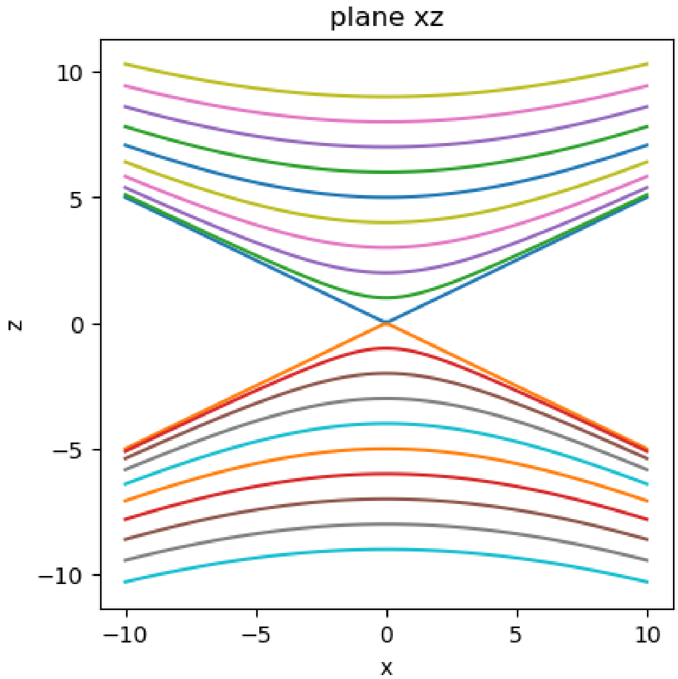
In [29]:

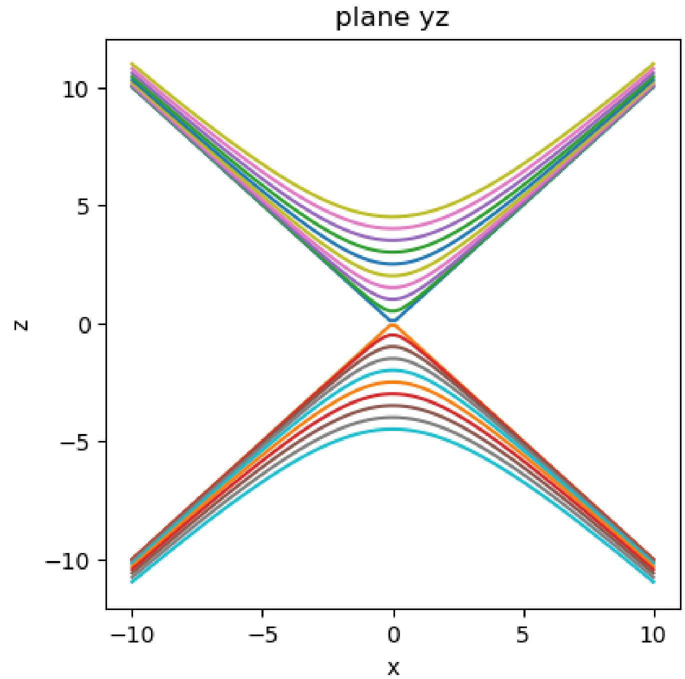
```
x0,y0,z0=0,0,0
a,b,c=1,1,1

plot_z6(a,b,c,x0,y0,z0)
```

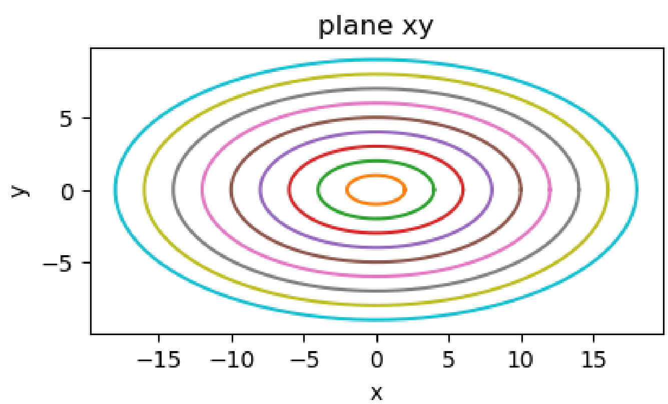


In [30]:

```
x0,y0,z0=0,0,0
a,b,c=2,1,1

plot_z6_projections(a,b,c,x0,y0,z0)
```







Domain and range in $\mathfrak{R}^3$

- Domain is the **region** in the xy-plane for which the function is defined.
- If the domain covers ALL the region it will be: $Domain = \mathfrak{R}^2$ or $Domain = (-\infty, \infty) \times (-\infty, \infty)$
- If the Domain is defined for $x \in [a, b]$ and $y \in [c, d]$, it is expressed as: $Domain = [a, b] \times [c, d]$

Exercise 3

Find and sketch the domain of:

1. $z = \sqrt{1-x^2} - \sqrt{1-y^2}$

1. $z = \sqrt{x+y+1}$

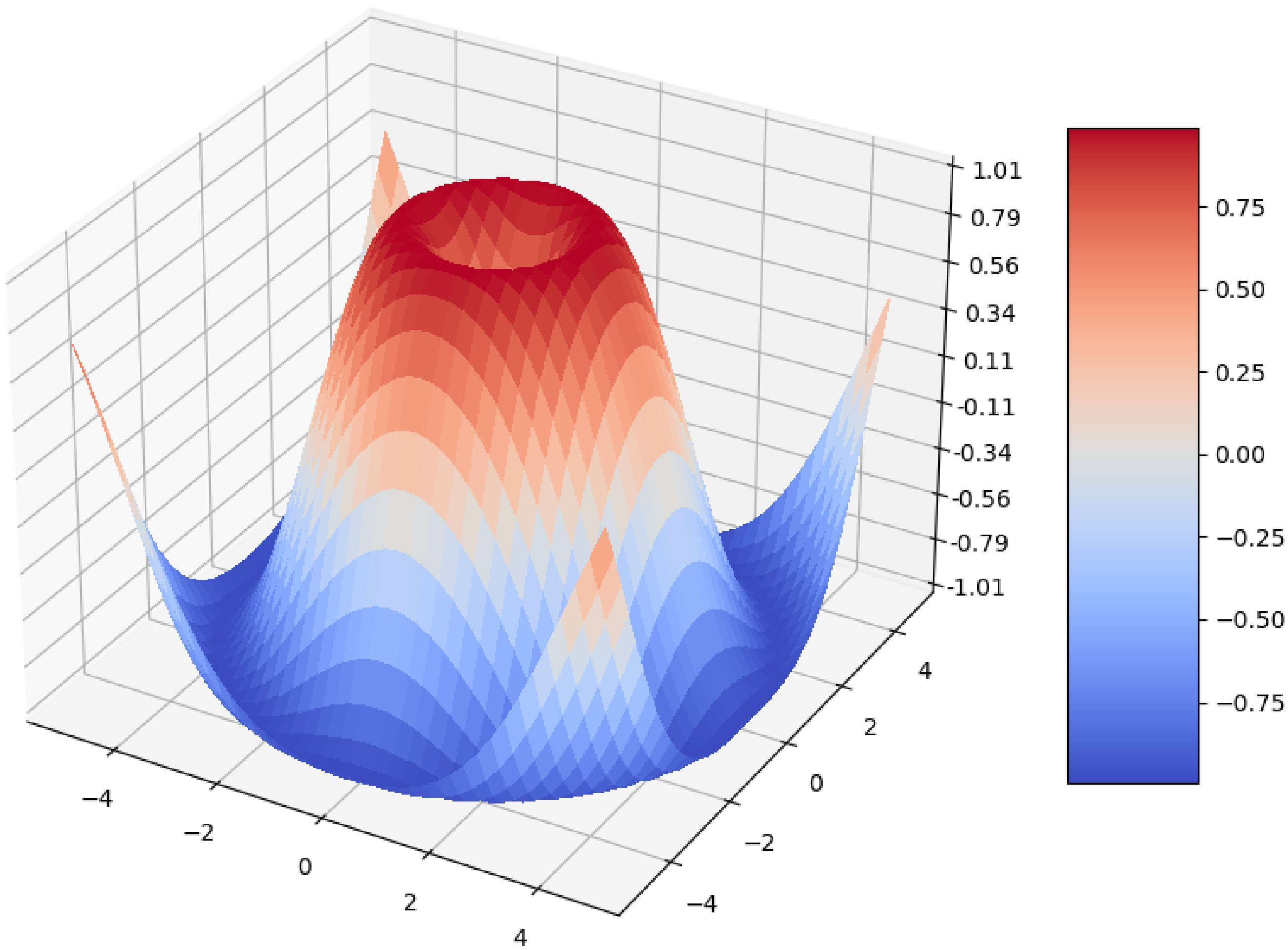
1. $z = \frac{\sqrt{2-x^2-y}}{x-1}$

1. $z = x\ln(x-y^2)$

1. $z = \frac{\sec(\pi y)}{\ln(x-y)}$

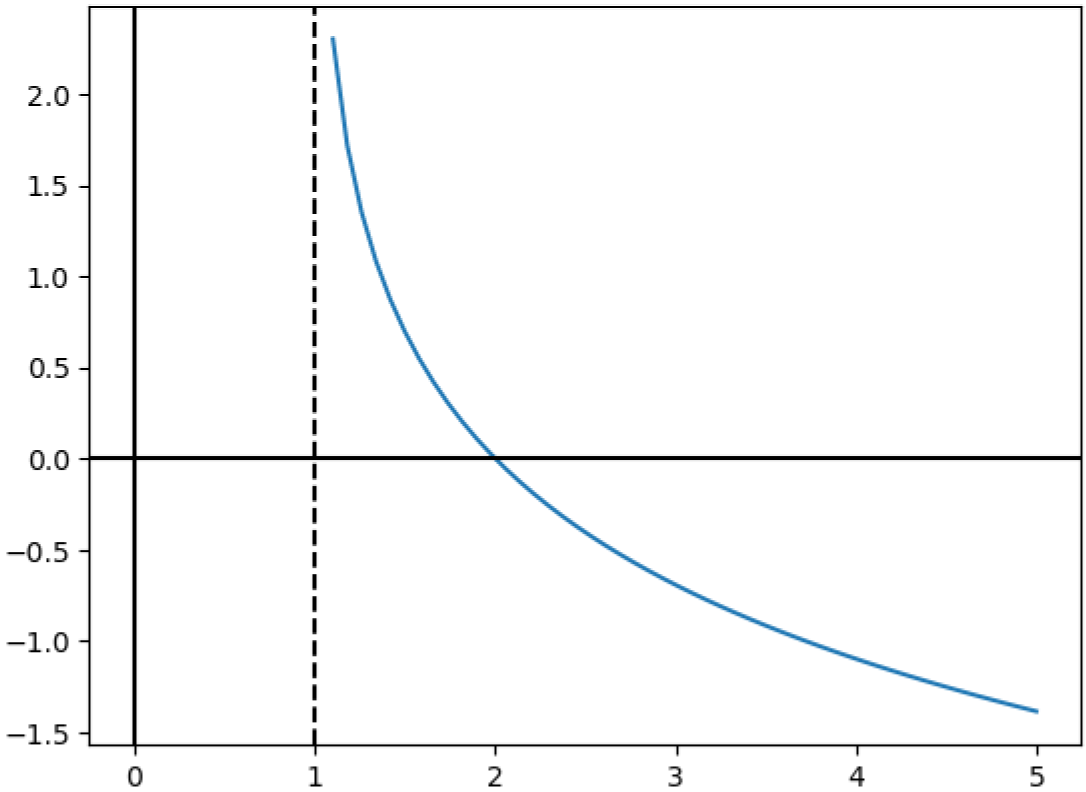
1. $x^2 + (y-1)^2 + z^2 = 9$

In [31]:
```
import matplotlib.pyplot as plt
from matplotlib import cm
from matplotlib.ticker import LinearLocator
import numpy as np

fig, ax = plt.subplots(subplot_kw={"projection": "3d"})

# Make data.
X = np.arange(-5, 5, 0.25)
Y = np.arange(-5, 5, 0.25)
X, Y = np.meshgrid(X, Y)
R = np.sqrt(X**2 + Y**2)
Z = np.sin(R)

# Plot the surface.
surf = ax.plot_surface(X, Y, Z, cmap=cm.coolwarm,
                       linewidth=0, antialiased=False)

# Customize the z axis.
ax.set_zlim(-1.01, 1.01)
ax.zaxis.set_major_locator(LinearLocator(10))
# A StrMethodFormatter is used automatically
ax.zaxis.set_major_formatter('{x:.02f}')

# Add a color bar which maps values to colors.
fig.colorbar(surf, shrink=0.5, aspect=5)

plt.show()
```



In [4]:
```
import matplotlib.pyplot as plt
import numpy as np

x=np.linspace(1.1,5)
plt.plot(x,-np.log(x-1))
plt.axhline(y = 0, color = 'k', linestyle = '-')
plt.axvline(x = 0, color = 'k', linestyle = '-')
plt.axvline(x = 1, color = 'k', linestyle = '--')
```

Out[4]:
```
<matplotlib.lines.Line2D at 0x19b146dd4e0>
```



In [8]:
```
x=np.linspace(0,2*np.pi/3)
plt.plot(x,2*np.cos(3*x+np.pi))
plt.axhline(y = 0, color = 'k', linestyle = '-')
plt.axvline(x = 0, color = 'k', linestyle = '-')
plt.axvline(x = np.pi/3, color = 'k', linestyle = '--')
```

Out[8]: <matplotlib.lines.Line2D at 0x19b148ea410>