18CS10069_Q3

October 23, 2021

[ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import random
from math import sqrt
from tabulate import tabulate
from mpl_toolkits import mplot3d
from matplotlib import cm
from mpl_toolkits.mplot3d import Axes3D
import math
```

[ ]:
```
# function to find the determinant of A
def det(A):
    if A.shape[0] != A.shape[1]:
        return None
    det = np.linalg.det(A)
    return det
```

[ ]:
```
def plot(x,y,z=None, N=100, dim=2):
    if dim == 3:
        plt.rcParams["figure.figsize"] = (5,5)           # set the figure size
        ax = plt.axes(projection='3d')  # plot the axes
        stride = 3

        x = x.reshape((N,N))
        y = y.reshape((N,N))
        z = z.reshape((N,N))

        # plot the surface in 3D
        ax.plot_surface(x , y , z , linewidth=0.0, cstride=stride,␣
 ↪rstride=stride)

        plt.grid()
        plt.show()

    elif dim == 2:
        plt.figure(figsize=(5,5))   # set the figure size
        ax = plt.axes()                 # plot the axes
```

```
        ax.plot(x, y)                        # plot the input points

        plt.grid()

        # limit the figure between xlim and ylim
        xlim = (math.floor(min(x)) - 1, math.ceil(max(x)) + 1)
        ylim = (math.floor(min(y)) - 1, math.ceil(max(y)) + 1)
        plt.xlim(xlim[0],xlim[1])
        plt.ylim(ylim[0],ylim[1])

        plt.show()
```

[ ]:
```
def parametric_form(dim=2, N=100):
    # obtain the parametric forms of a 2D circle or 3D sphere
    if dim == 2:
        # obtain the parametric form of points in 2D
        M = N * N
        theta = np.linspace(0, 2*np.pi, M)
        x = np.cos(theta) # cos(theta)
        y = np.sin(theta) # sin(theta)

        plot(x,y, dim=2)

        return x, y

    elif dim == 3:
        theta = np.linspace(0, 2 * np.pi, N)
        phi = np.linspace(0, np.pi, N)
        x = np.outer(np.cos(theta), np.sin(phi)) # cos(theta) . sin(phi)
        y = np.outer(np.sin(theta), np.sin(phi)) # sin(theta) . sin(phi)
        z = np.outer(np.ones(np.size(theta)), np.cos(phi)) # cos(phi)

        x = x.reshape((N*N,))
        y = y.reshape((N*N,))
        z = z.reshape((N*N,))

        plot(x, y, z, dim=3)
        return x, y, z
```

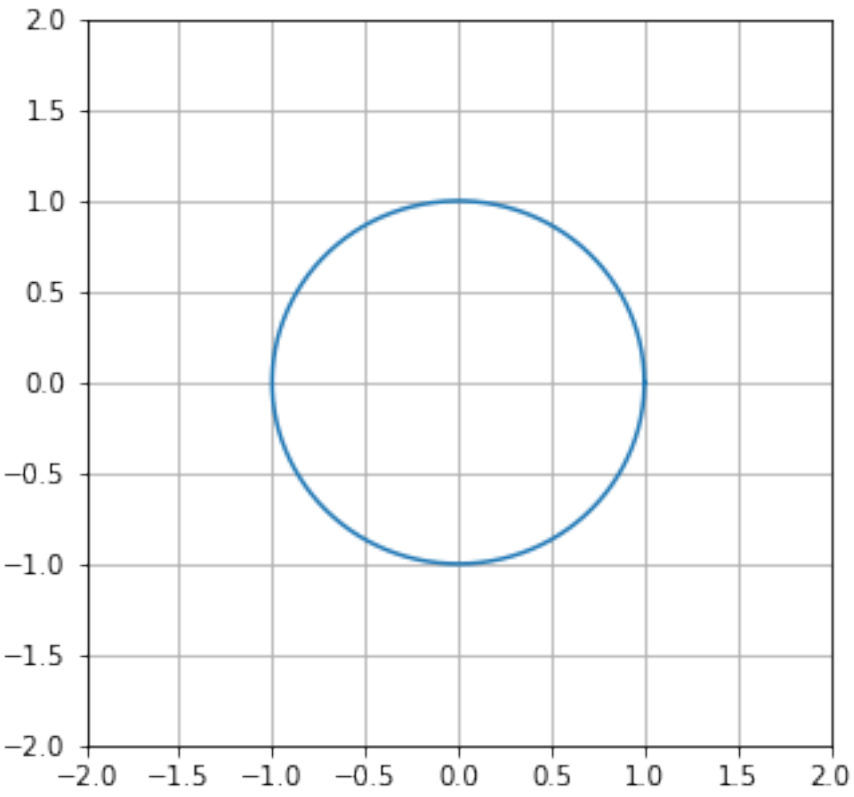
(a)

NOTE: The ellipsoid in 3D is very faintly visible

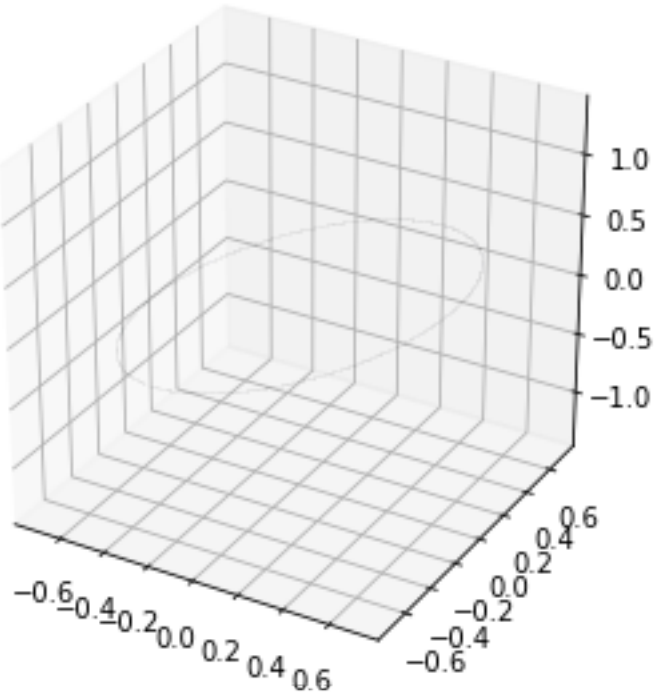
[ ]:
```
A = np.array([[-1/sqrt(2), 0], [0, -1/sqrt(2)], [-1, 1]])

print(f'Unit sphere in {A.shape[1]}D')
```

```
x, y = parametric_form(dim=A.shape[1])

print(f"\nEllipsoid in {A.shape[0]}D")
circle = np.array([x,y])
ellipse = np.dot(A, circle)
plot(ellipse[0], ellipse[1], ellipse[2], dim=A.shape[0])

determinant = det(A)
if determinant == None:
    print(f"\nDeterminant does not exists")
else:
    print(f"\nDeterminant of matrix A: {determinant:.4f}")
    if determinant != 0:
        print(f"\nThe matrix A is also invertible")
    else:
        print(f"\nThe matrix A is not invertible")

condition_num = np.linalg.cond(A)
print(f"\nCondition number of A: {condition_num:.4f}")
```

```
Unit sphere in 2D
```



```
Ellipsoid in 3D
```



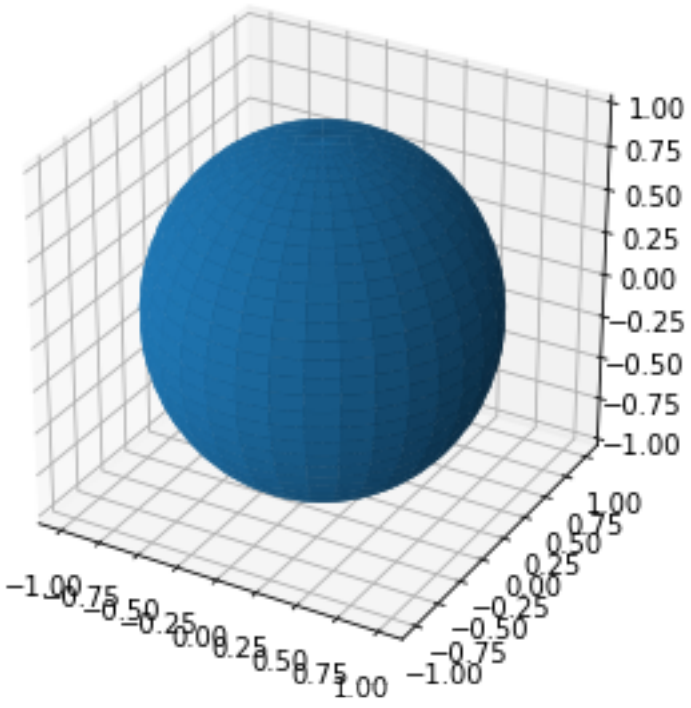

```
Determinant does not exists

Condition number of A: 2.2361
```

(b)

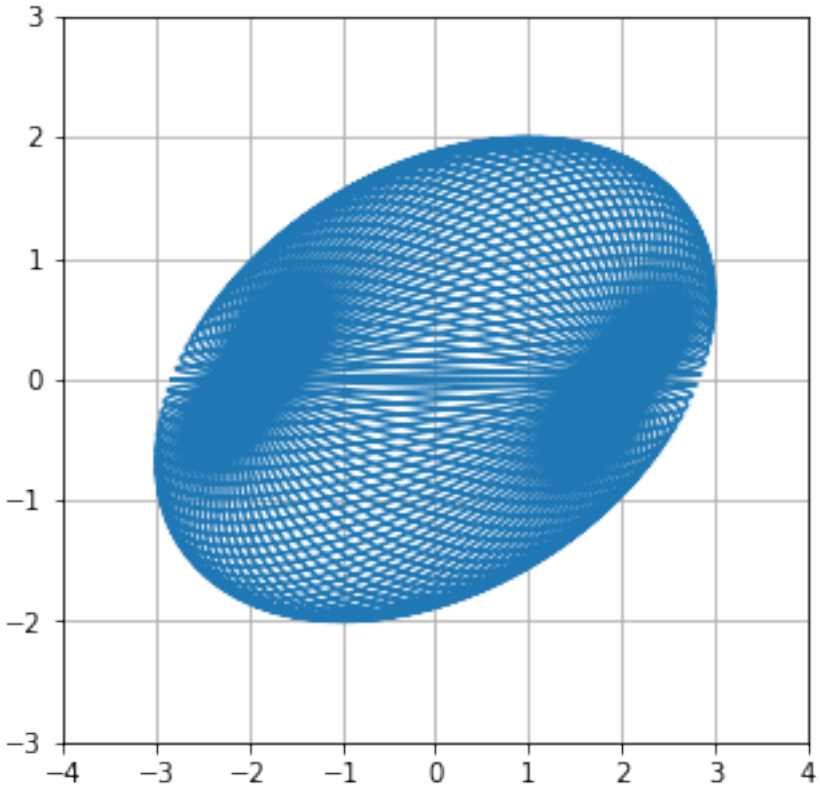
[ ]:
```
A = np.array([[-2, 1, 2],[0, 2, 0]])

print(f'Unit sphere in {A.shape[1]}D')
x, y, z = parametric_form(dim=A.shape[1])

print(f"\nEllipsoid in {A.shape[0]}D")
circle = np.array([x,y,z])
ellipse = np.dot(A, circle)
plot(ellipse[0], ellipse[1], dim=A.shape[0])

determinant = det(A)
```

```
if determinant == None:
    print(f"\nDeterminant does not exists")
else:
    print(f"\nDeterminant of matrix A: {determinant:.4f}")
    if determinant != 0:
        print(f"\nThe matrix A is also invertible")
    else:
        print(f"\nThe matrix A is not invertible")

condition_num = np.linalg.cond(A)
print(f"\nCondition number of A: {condition_num:.4f}")
```

Unit sphere in 3D



Ellipsoid in 2D



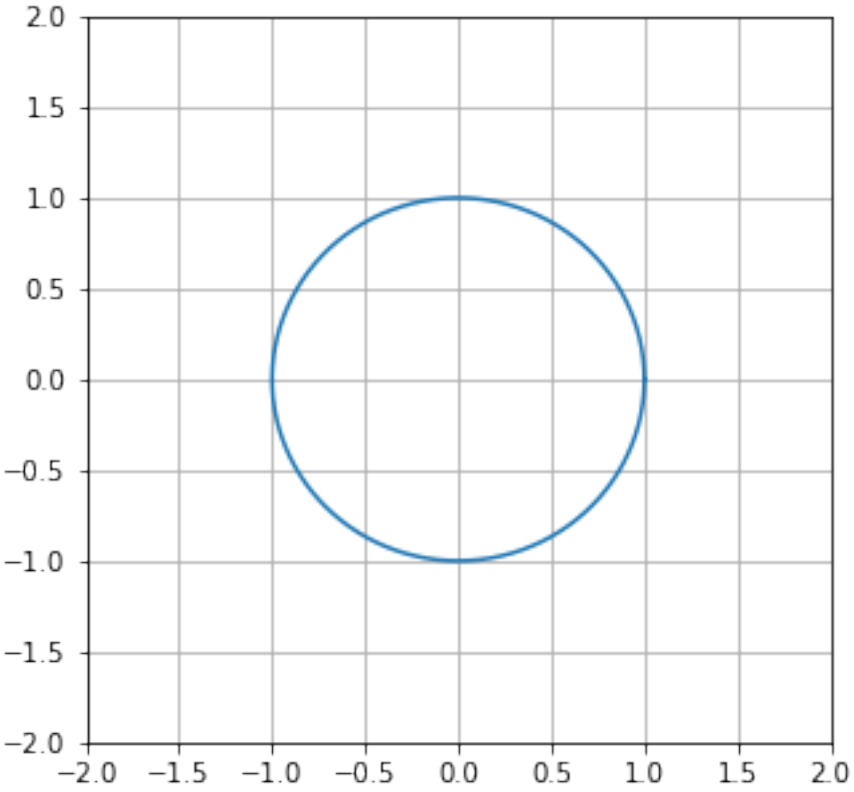

```
Determinant does not exists

Condition number of A: 1.7150
```

(c)

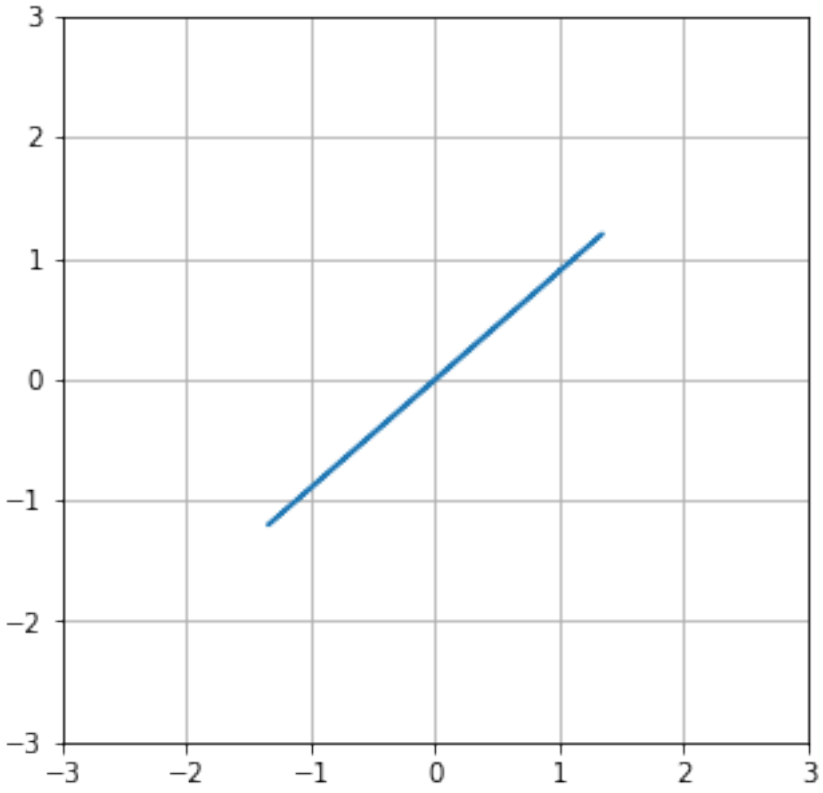
[ ]:
```
A = np.array([[1, 0.9],[0.9, 0.8]])

print(f'Unit sphere in {A.shape[1]}D')
x, y = parametric_form(dim=A.shape[1])

print(f"\nEllipsoid in {A.shape[0]}D")
circle = np.array([x,y])
ellipse = np.dot(A, circle)
plot(ellipse[0], ellipse[1], dim=A.shape[0])

determinant = det(A)
if determinant == None:
    print(f"\nDeterminant does not exists")
```

```
else:
    print(f"\nDeterminant of matrix A: {determinant:.4f}")
    if determinant != 0:
        print(f"\nThe matrix A is also invertible")
    else:
        print(f"\nThe matrix A is not invertible")

condition_num = np.linalg.cond(A)
print(f"\nCondition number of A: {condition_num:.4f}")
```

Unit sphere in 2D



Ellipsoid in 2D



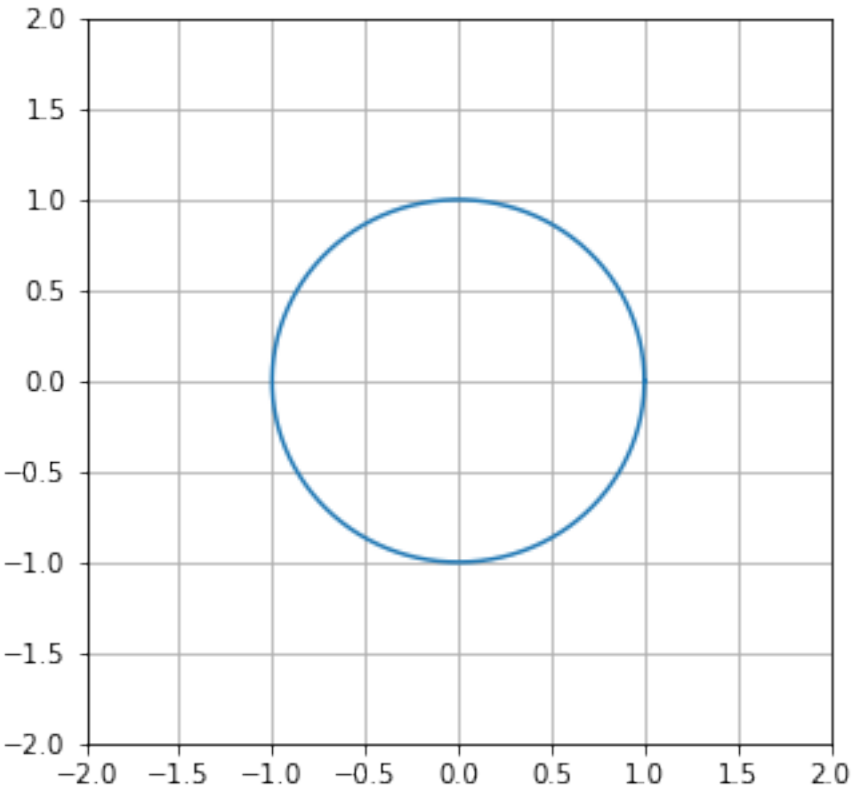

```
Determinant of matrix A: -0.0100

The matrix A is also invertible

Condition number of A: 325.9969
```

(d)

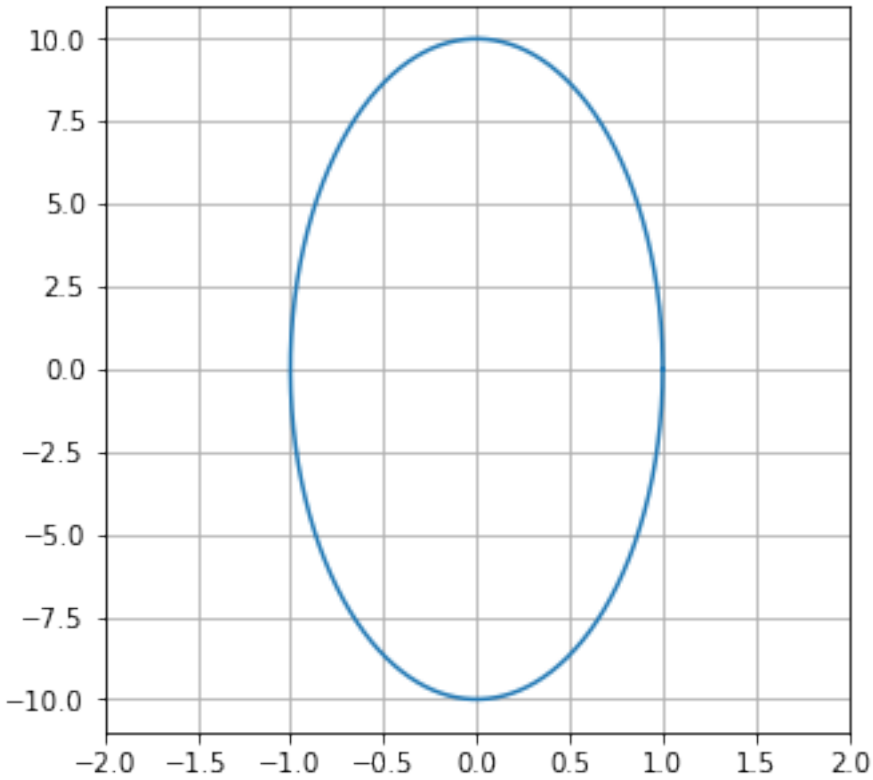
[ ]:
```
A = np.array([[1, 0],[0, -10]])

print(f'Unit sphere in {A.shape[1]}D')
x, y = parametric_form(dim=A.shape[1])

print(f"\nEllipsoid in {A.shape[0]}D")
circle = np.array([x,y])
ellipse = np.dot(A, circle)
plot(ellipse[0], ellipse[1], dim=A.shape[0])

determinant = det(A)
```

```
if determinant == None:
    print(f"\nDeterminant does not exists")
else:
    print(f"\nDeterminant of matrix A: {determinant:.4f}")
    if determinant != 0:
        print(f"\nThe matrix A is also invertible")
    else:
        print(f"\nThe matrix A is not invertible")

condition_num = np.linalg.cond(A)
print(f"\nCondition number of A: {condition_num:.4f}")
```

Unit sphere in 2D



Ellipsoid in 2D




```
Determinant of matrix A: -10.0000

The matrix A is also invertible

Condition number of A: 10.0000
```

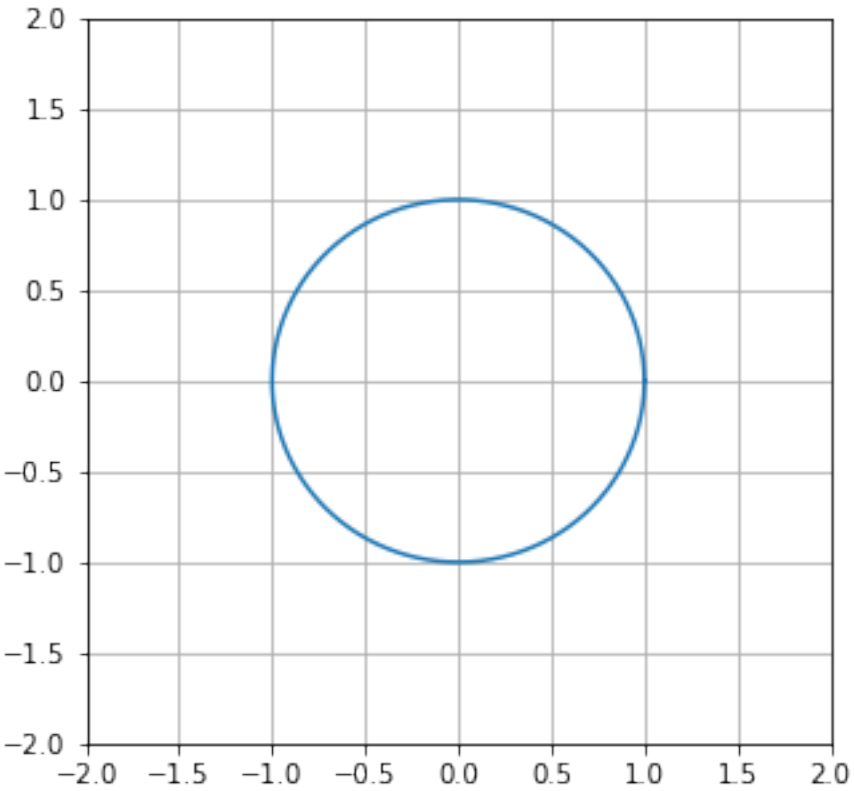
(e)

$\varepsilon = 10$

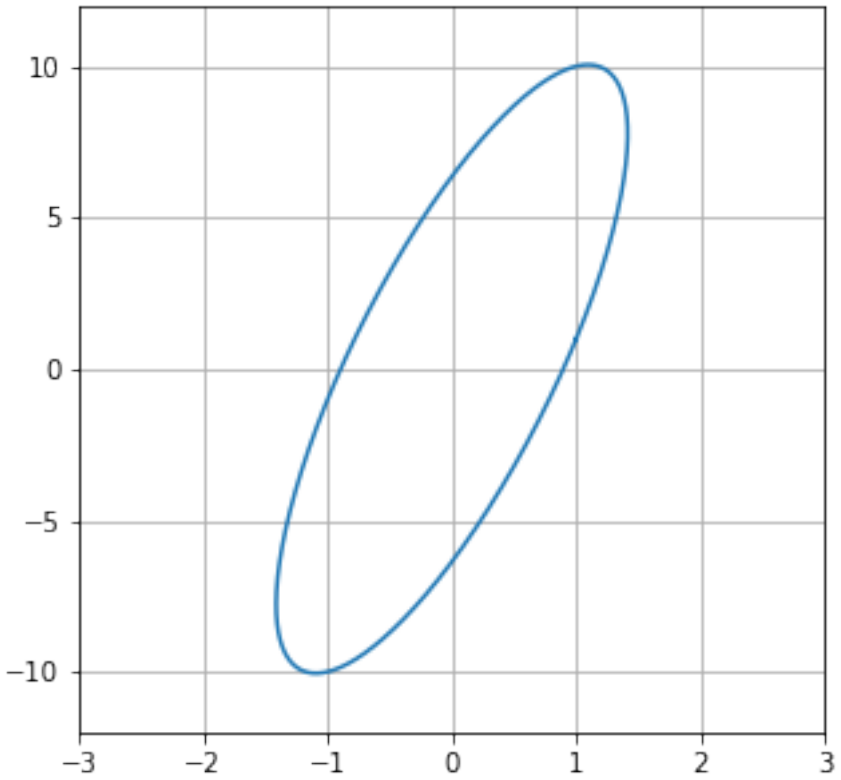
[ ]:
```
A = np.array([[1, 1],[1, 10]])

print(f'Unit sphere in {A.shape[1]}D')
x, y = parametric_form(dim=A.shape[1])

print(f"\nEllipsoid in {A.shape[0]}D")
circle = np.array([x,y])
ellipse = np.dot(A, circle)
plot(ellipse[0], ellipse[1], dim=A.shape[0])
```

```
determinant = det(A)
if determinant == None:
    print(f"\nDeterminant does not exists")
else:
    print(f"\nDeterminant of matrix A: {determinant:.4f}")
    if determinant != 0:
        print(f"\nThe matrix A is also invertible")
    else:
        print(f"\nThe matrix A is not invertible")

condition_num = np.linalg.cond(A)
print(f"\nCondition number of A: {condition_num:.4f}")
```

Unit sphere in 2D



Ellipsoid in 2D



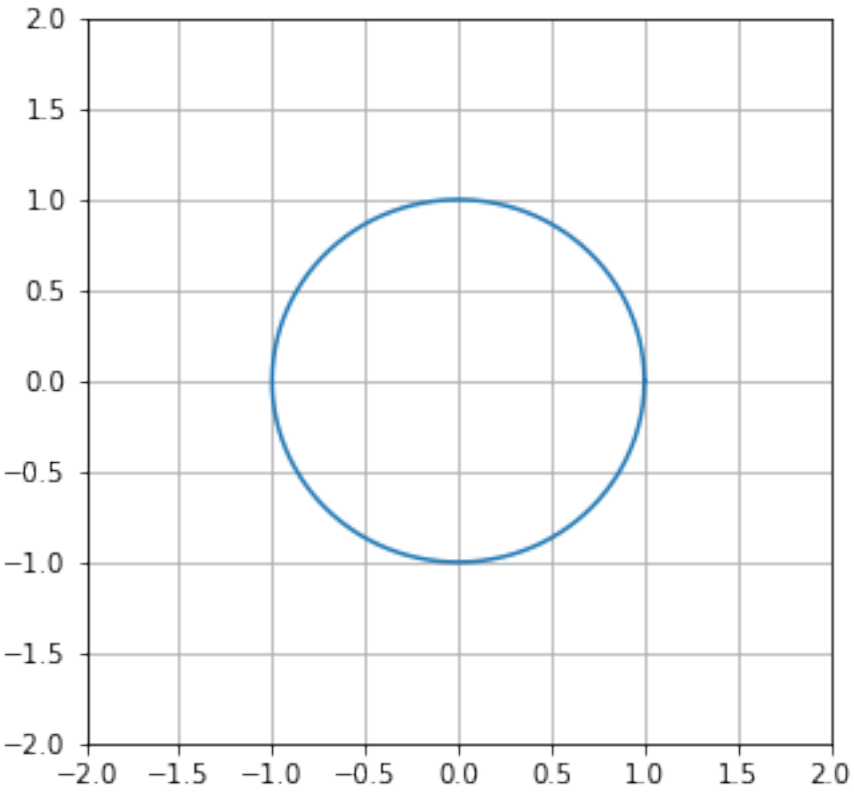

```
Determinant of matrix A: 9.0000

The matrix A is also invertible

Condition number of A: 11.3564
```

$\varepsilon = 5$

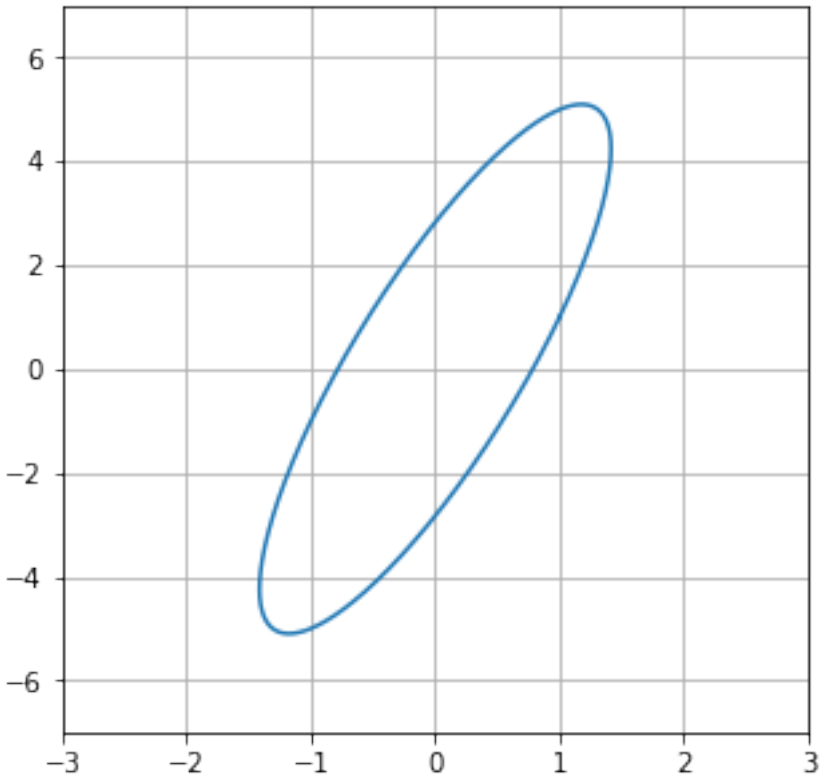
[ ]:
```
A = np.array([[1, 1],[1, 5]])

print(f'Unit sphere in {A.shape[1]}D')
x, y = parametric_form(dim=A.shape[1])

print(f"\nEllipsoid in {A.shape[0]}D")
circle = np.array([x,y])
ellipse = np.dot(A, circle)
plot(ellipse[0], ellipse[1], dim=A.shape[0])

determinant = det(A)
```

```
if determinant == None:
    print(f"\nDeterminant does not exists")
else:
    print(f"\nDeterminant of matrix A: {determinant:.4f}")
    if determinant != 0:
        print(f"\nThe matrix A is also invertible")
    else:
        print(f"\nThe matrix A is not invertible")

condition_num = np.linalg.cond(A)
print(f"\nCondition number of A: {condition_num:.4f}")
```

Unit sphere in 2D



Ellipsoid in 2D



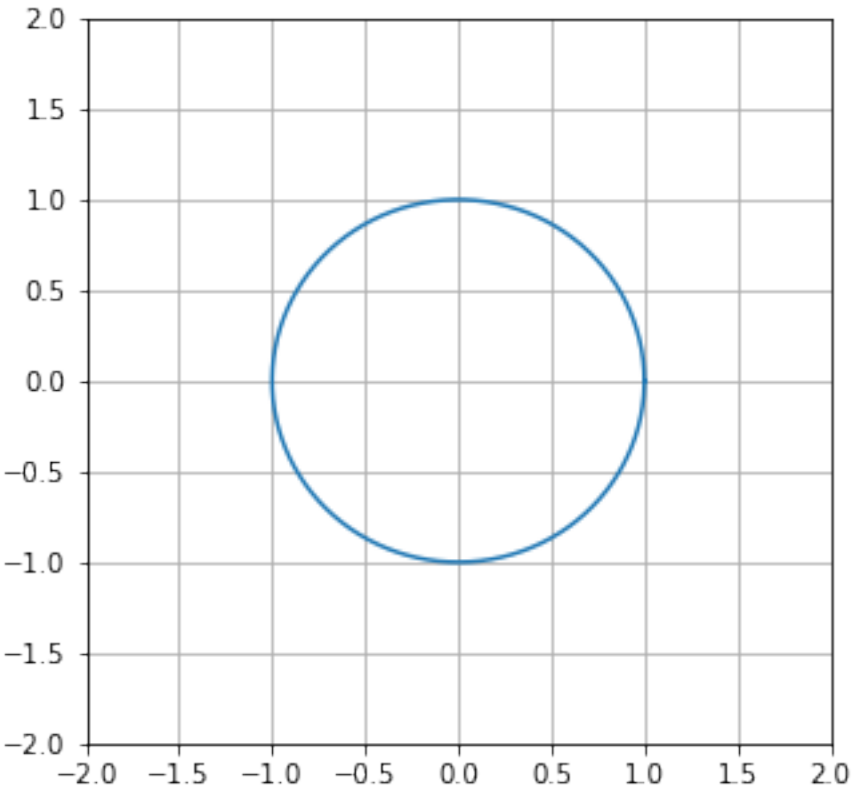

```
Determinant of matrix A: 4.0000

The matrix A is also invertible

Condition number of A: 6.8541
```

$\varepsilon = 1$

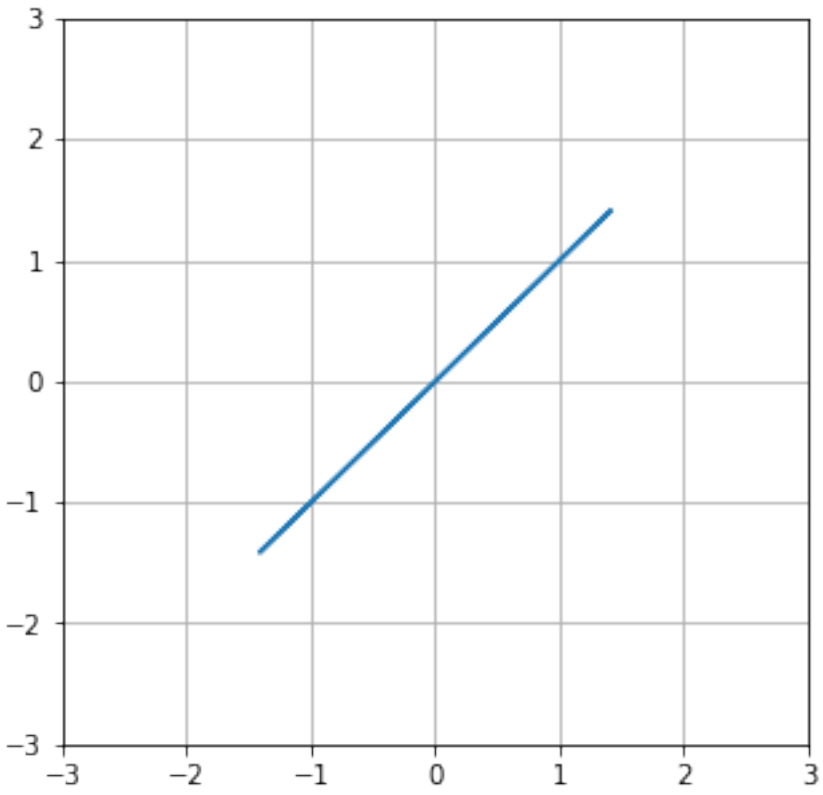
[ ]:
```
A = np.array([[1, 1],[1, 1]])

print(f'Unit sphere in {A.shape[1]}D')
x, y = parametric_form(dim=A.shape[1])

print(f"\nEllipsoid in {A.shape[0]}D")
circle = np.array([x,y])
ellipse = np.dot(A, circle)
plot(ellipse[0], ellipse[1], dim=A.shape[0])

determinant = det(A)
```

```
if determinant == None:
    print(f"\nDeterminant does not exists")
else:
    print(f"\nDeterminant of matrix A: {determinant:.4f}")
    if determinant != 0:
        print(f"\nThe matrix A is also invertible")
    else:
        print(f"\nThe matrix A is not invertible")

condition_num = np.linalg.cond(A)
print(f"\nCondition number of A: {condition_num:.4f}")
```

Unit sphere in 2D



Ellipsoid in 2D



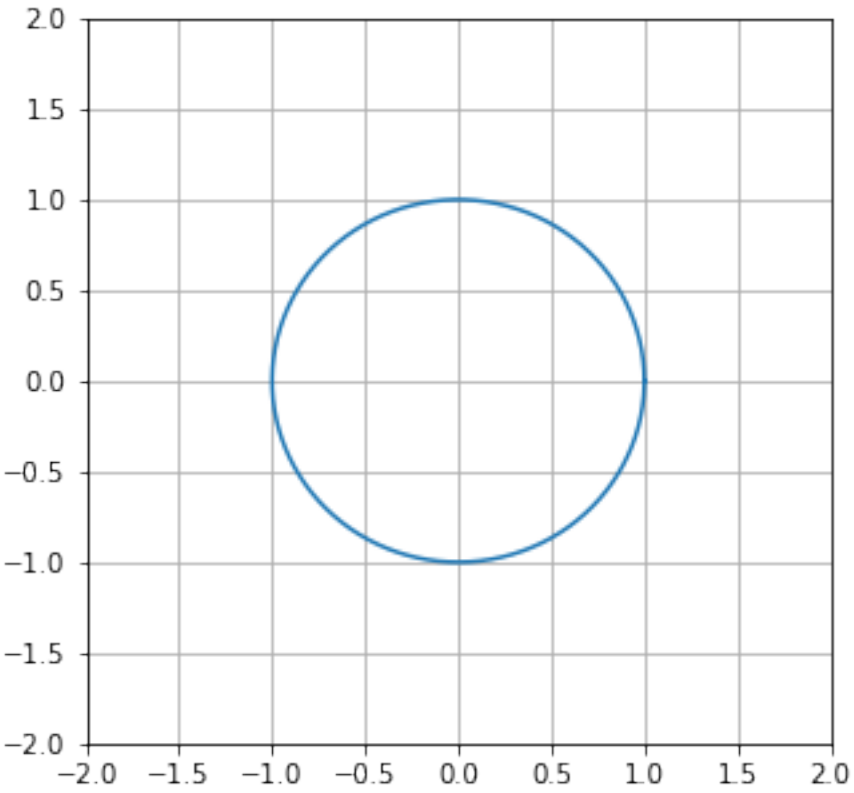

```
Determinant of matrix A: 0.0000

The matrix A is not invertible

Condition number of A: 59617770476389832.0000
```

$\varepsilon = 0.1$

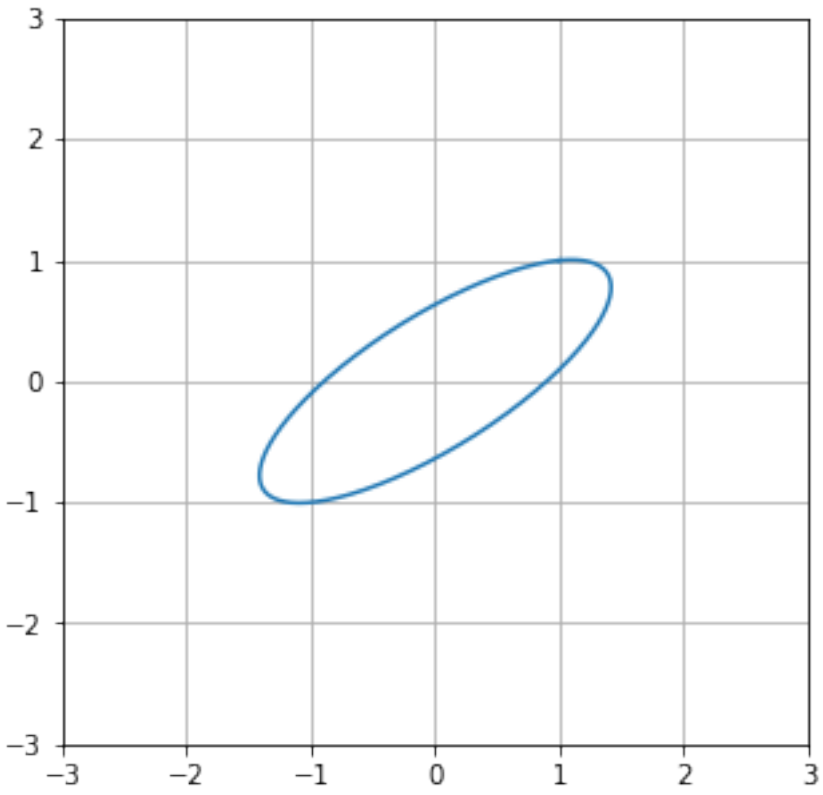
[ ]:
```
A = np.array([[1, 1],[1, 0.1]])

print(f'Unit sphere in {A.shape[1]}D')
x, y = parametric_form(dim=A.shape[1])

print(f"\nEllipsoid in {A.shape[0]}D")
circle = np.array([x,y])
ellipse = np.dot(A, circle)
plot(ellipse[0], ellipse[1], dim=A.shape[0])

determinant = det(A)
```

```
if determinant == None:
    print(f"\nDeterminant does not exists")
else:
    print(f"\nDeterminant of matrix A: {determinant:.4f}")
    if determinant != 0:
        print(f"\nThe matrix A is also invertible")
    else:
        print(f"\nThe matrix A is not invertible")

condition_num = np.linalg.cond(A)
print(f"\nCondition number of A: {condition_num:.4f}")
```

Unit sphere in 2D



Ellipsoid in 2D



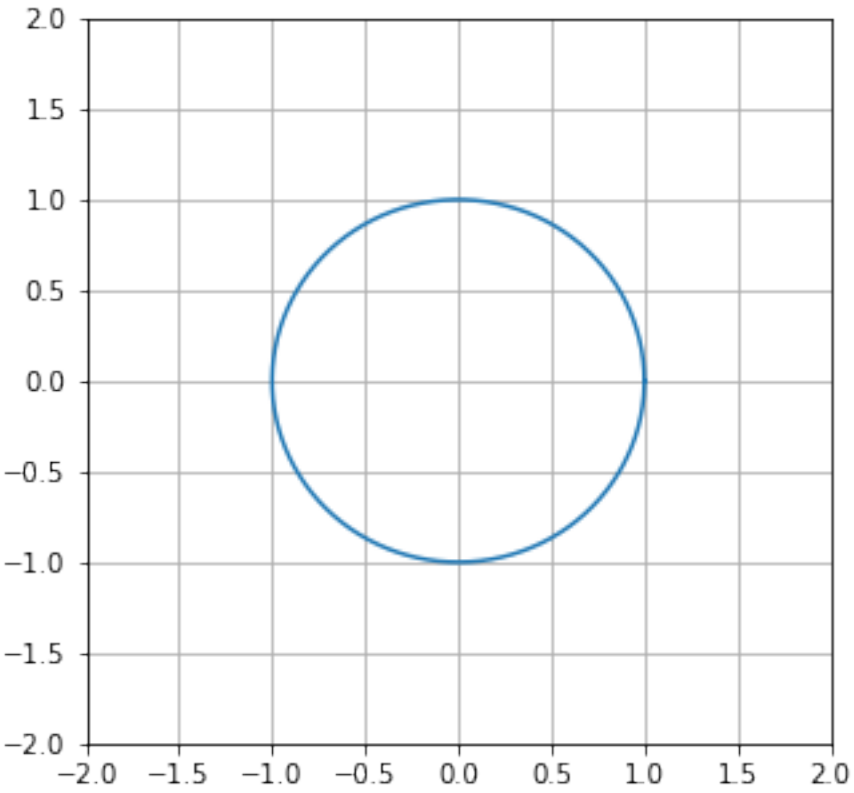

```
Determinant of matrix A: -0.9000

The matrix A is also invertible

Condition number of A: 3.0125
```

$\varepsilon = 0.01$

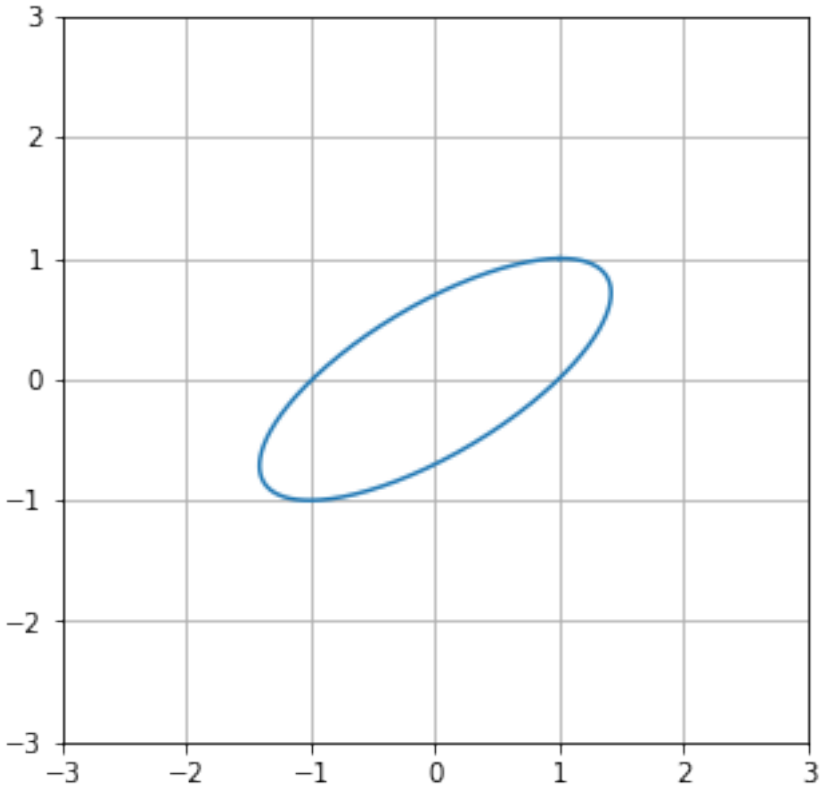
[ ]:
```
A = np.array([[1, 1],[1, 0.01]])

print(f'Unit sphere in {A.shape[1]}D')
x, y = parametric_form(dim=A.shape[1])

print(f"\nEllipsoid in {A.shape[0]}D")
circle = np.array([x,y])
ellipse = np.dot(A, circle)
plot(ellipse[0], ellipse[1], dim=A.shape[0])

determinant = det(A)
```

```
if determinant == None:
    print(f"\nDeterminant does not exists")
else:
    print(f"\nDeterminant of matrix A: {determinant:.4f}")
    if determinant != 0:
        print(f"\nThe matrix A is also invertible")
    else:
        print(f"\nThe matrix A is not invertible")

condition_num = np.linalg.cond(A)
print(f"\nCondition number of A: {condition_num:.4f}")
```

Unit sphere in 2D



Ellipsoid in 2D




```
Determinant of matrix A: -0.9900

The matrix A is also invertible

Condition number of A: 2.6536
```

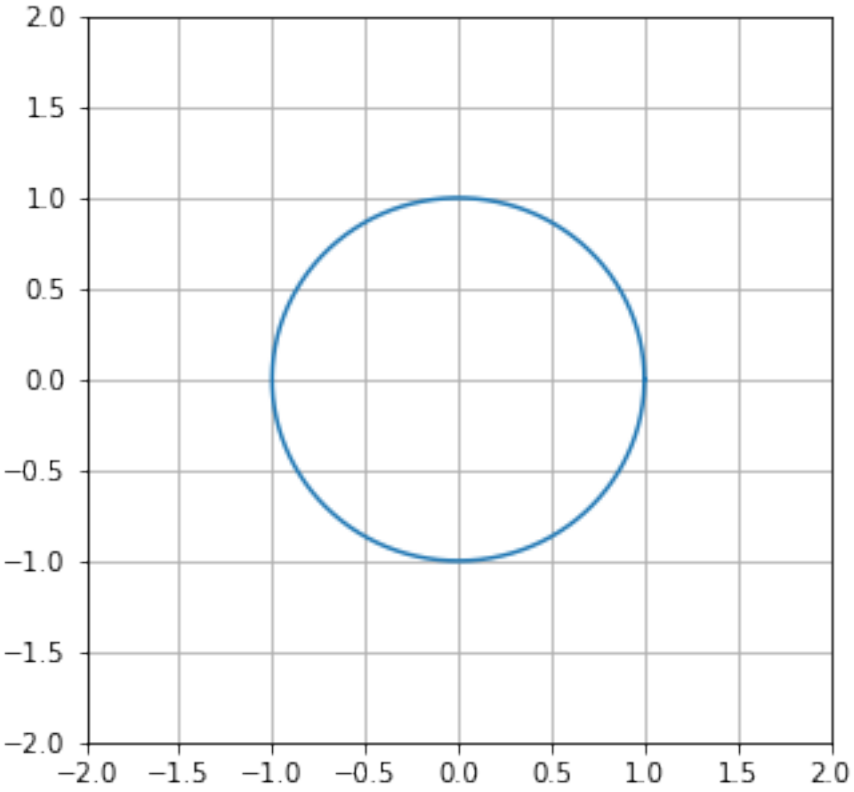
$\varepsilon = 0.0001$

--

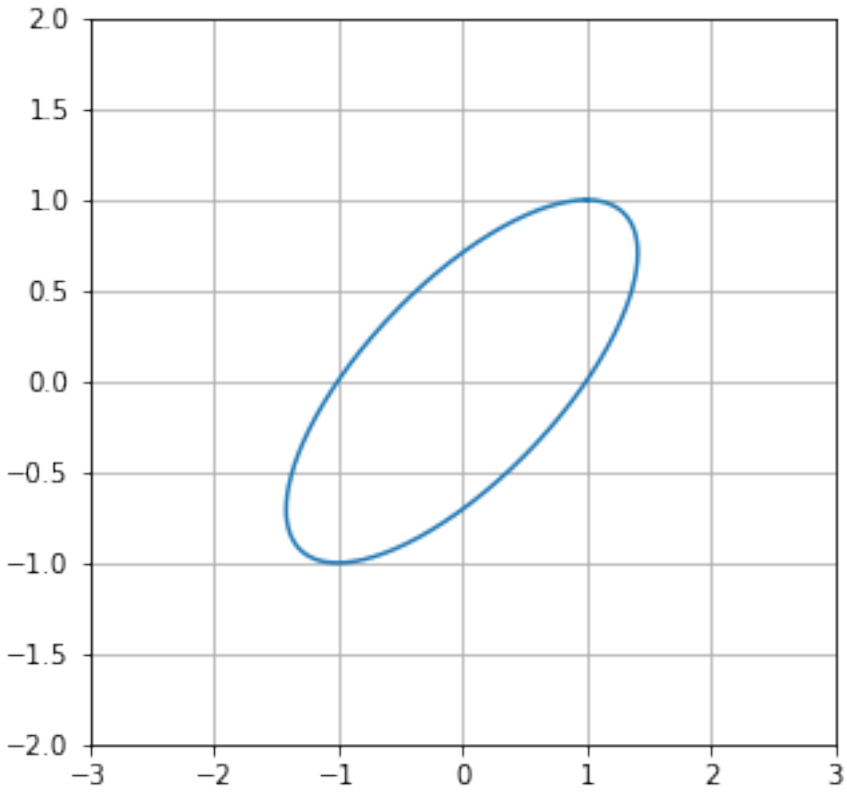
[ ]:
```
A = np.array([[1, 1],[1, 1e-4]])

print(f'Unit sphere in {A.shape[1]}D')
x, y = parametric_form(dim=A.shape[1])

print(f"\nEllipsoid in {A.shape[0]}D")
circle = np.array([x,y])
ellipse = np.dot(A, circle)
plot(ellipse[0], ellipse[1], dim=A.shape[0])

determinant = det(A)
if determinant == None:
    print(f"\nDeterminant does not exists")
```

```
else:
    print(f"\nDeterminant of matrix A: {determinant:.4f}")
    if determinant != 0:
        print(f"\nThe matrix A is also invertible")
    else:
        print(f"\nThe matrix A is not invertible")

condition_num = np.linalg.cond(A)
print(f"\nCondition number of A: {condition_num:.4f}")
```

Unit sphere in 2D



Ellipsoid in 2D




```
Determinant of matrix A: -0.9999

The matrix A is also invertible

Condition number of A: 2.6184
```

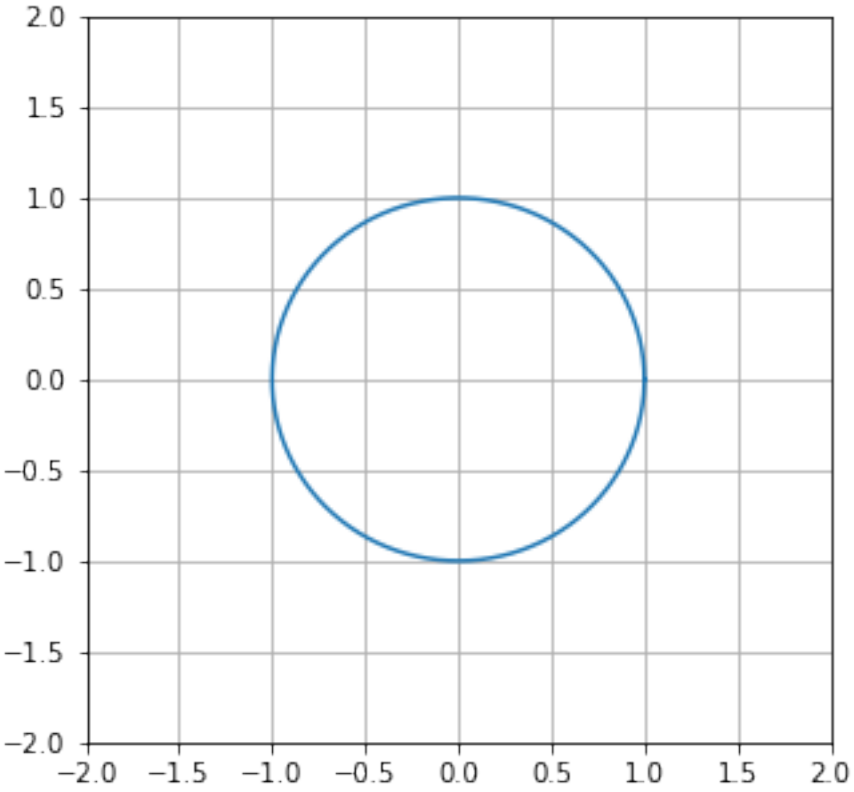
$\varepsilon = 0$

--

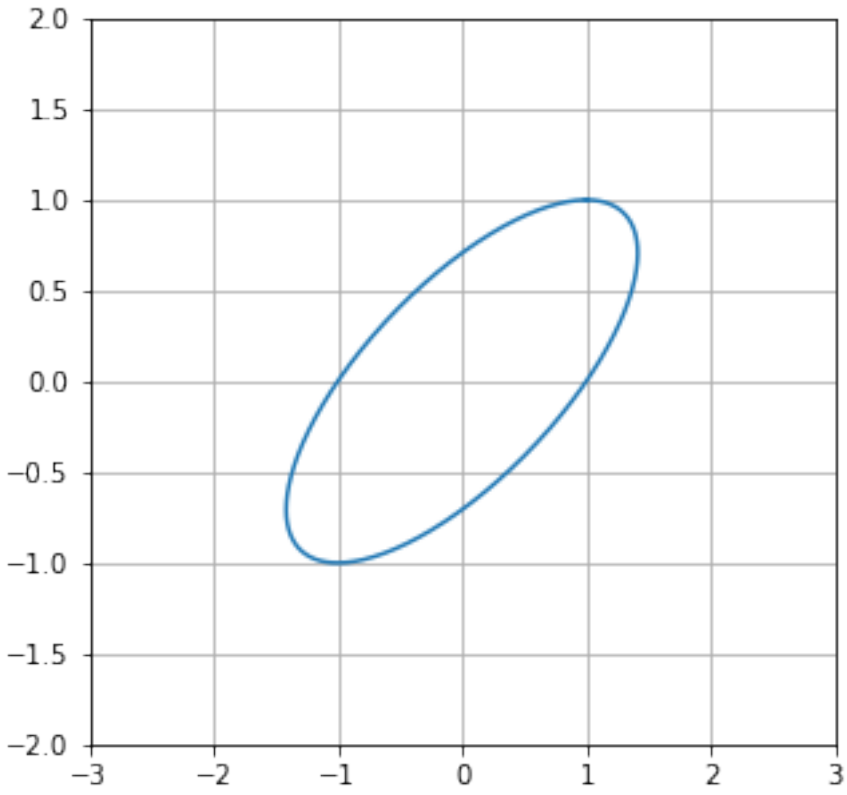
[ ]:
```
A = np.array([[1, 1],[1, 0]])

print(f'Unit sphere in {A.shape[1]}D')
x, y = parametric_form(dim=A.shape[1])

print(f"\nEllipsoid in {A.shape[0]}D")
circle = np.array([x,y])
ellipse = np.dot(A, circle)
plot(ellipse[0], ellipse[1], dim=A.shape[0])

determinant = det(A)
if determinant == None:
    print(f"\nDeterminant does not exists")
```

```
else:
    print(f"\nDeterminant of matrix A: {determinant:.4f}")
    if determinant != 0:
        print(f"\nThe matrix A is also invertible")
    else:
        print(f"\nThe matrix A is not invertible")

condition_num = np.linalg.cond(A)
print(f"\nCondition number of A: {condition_num:.4f}")
```

Unit sphere in 2D



Ellipsoid in 2D




```
Determinant of matrix A: -1.0000

The matrix A is also invertible

Condition number of A: 2.6180
```

- There is as such no relationship between determinant and condition number.
- But, we can observe one relationship between determinant and condition number here, i.e., as the determinants come closer, the condition numbers also come closer (We specifically see in the case of various values of ε). Simply put, we see that as determinant converge to a particular value, their condition numbers also seem to converge.