

1. Задание Даны два вектора в трехмерном пространстве: (10,10,10) и (0,0,-10)
- A. Найдите их сумму. (на листочке)
- B. Напишите код на Python, реализующий расчет длины вектора, заданного его координатами. (в программе)

$$\vec{x}(10, 10, 10)$$

$$\vec{y}(0, 0, -10)$$

$$\vec{z} = \vec{x} + \vec{y} = (10, 10, 10) + (0, 0, -10) = (10, 10, 0)$$

```
In [10]: import math
A = (10, 10, 0)
summ = 0
for i in A:
    i = i**2
    summ += i
sqrt = math.sqrt(summ)
print(sqrt)
```

14.142135623730951

2. Задание (на листочке) Почему прямые не кажутся перпендикулярными? (см.ролик)

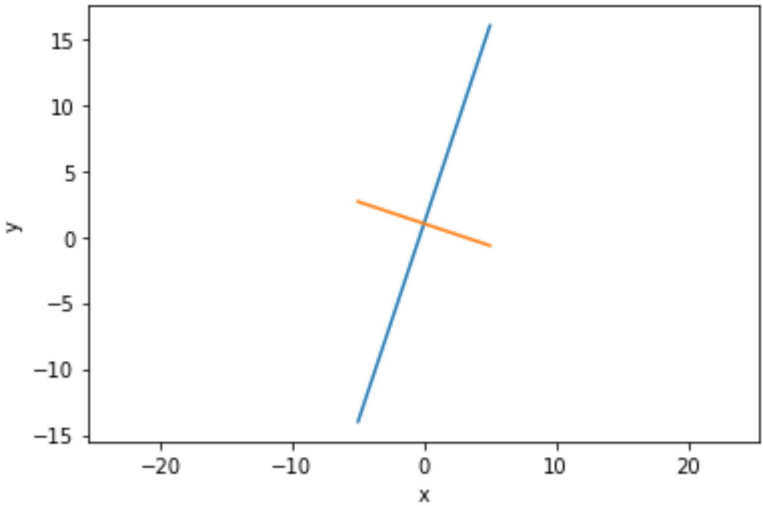
Ответ: в следствие разной размерности осей x, y создаётся, что-то подобное оптической иллюзии. При использовании команды plt.axis('equal'), всё становится на места.

```
In [158]: %matplotlib inline

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

```
In [12]: x = np.linspace(-5, 5, 21)
y1 = 3*x + 1
y2 = (-1/3)*x + 1
plt.axis('equal')
plt.plot(x, y1)
plt.plot(x, y2)
plt.xlabel('x')
plt.ylabel('y')
```

Out[12]: Text(0, 0.5, 'y')

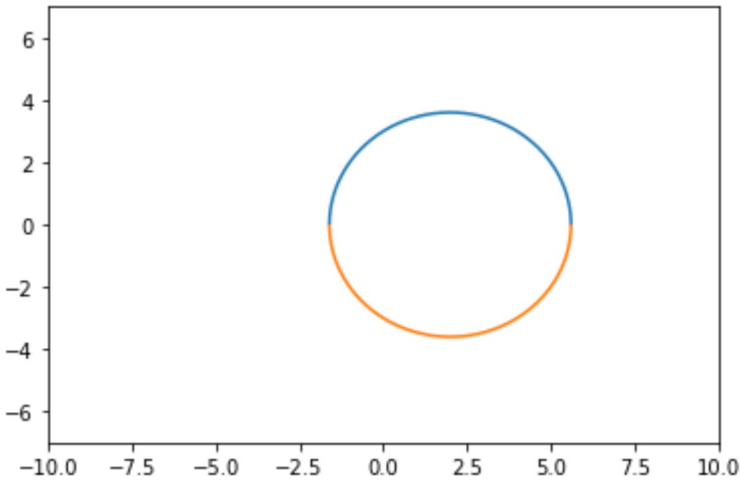


3. Задание (в программе) Напишите код на Python, реализующий построение графиков:
- A. окружности,
- B. эллипса,
- C. гиперболы.

```
In [94]: x = np.linspace(-10, 10, 40000)
y1 = np.sqrt((1 + x)*(5 - x)+4)
y2 = -np.sqrt((1 + x)*(5 - x)+4)
plt.ylim(-7, 7)
plt.xlim(-10, 10)
plt.plot(x, y1)
plt.plot(x, y2)

C:\Users\akins\AppData\Local\Temp\ipykernel_17672\3424762246.py:2: RuntimeWarning: invalid value encountered in sqrt
y1 = np.sqrt((1 + x)*(5 - x)+4)
C:\Users\akins\AppData\Local\Temp\ipykernel_17672\3424762246.py:3: RuntimeWarning: invalid value encountered in sqrt
y2 = -np.sqrt((1 + x)*(5 - x)+4)
```

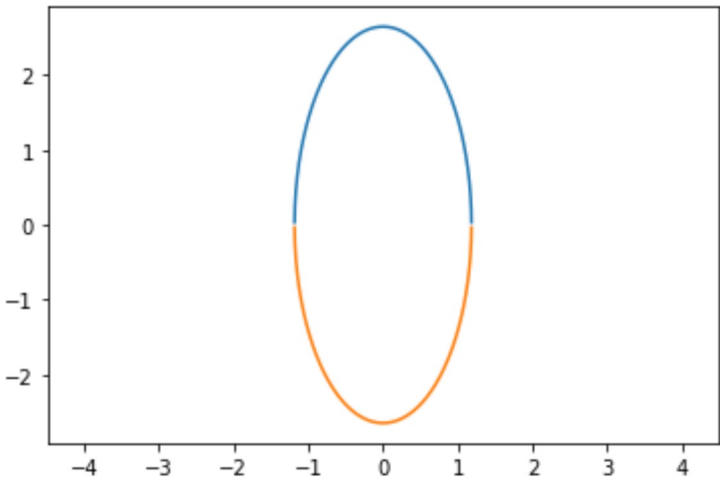
Out[94]: [<matplotlib.lines.Line2D at 0x1ad26c31bb0>]



```
In [104]: x = np.linspace(-10, 10, 10000)
y1 = np.sqrt((7 - (5*x**2)/1**2)/1**2)
y2 = -(np.sqrt((7 - (5*x**2)/1**2)/1**2))
plt.axis('equal')
plt.plot(x, y1)
plt.plot(x, y2)

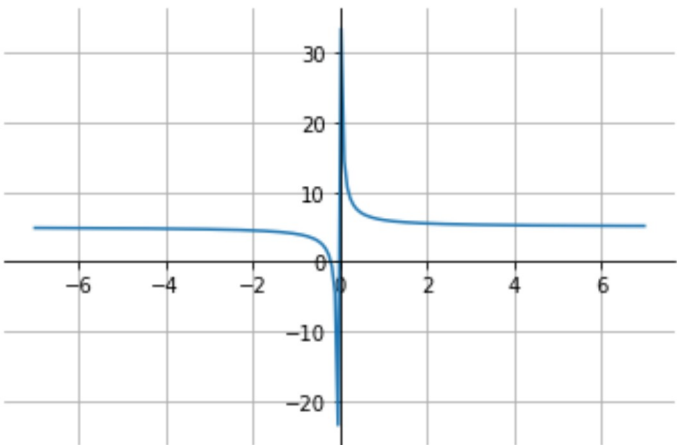
C:\Users\akins\AppData\Local\Temp\ipykernel_17672\1488526437.py:2: RuntimeWarning: invalid value encountered in sqrt
y1 = np.sqrt((7 - (5*x**2)/1**2)/1**2)
C:\Users\akins\AppData\Local\Temp\ipykernel_17672\1488526437.py:3: RuntimeWarning: invalid value encountered in sqrt
y2 = -(np.sqrt((7 - (5*x**2)/1**2)/1**2))
```

Out[104]: [<matplotlib.lines.Line2D at 0x1ad27fafc40>]



```
In [153]: x = np.linspace(-7, 7, 200)
y = 1/x + 5
ax = plt.gca()
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.xaxis.set_ticks_position('bottom')
ax.spines['bottom'].set_position(('data',0))
ax.yaxis.set_ticks_position('left')
ax.spines['left'].set_position(('data',0))
plt.grid(True)
plt.plot(x, y)
```

Out[153]: [<matplotlib.lines.Line2D at 0x1ad2c2b0a90>]



4. Задание (на листочке)

А. Пусть задана плоскость:

$$A \cdot x + B \cdot y + C \cdot z + D = 0$$

Напишите уравнение плоскости, параллельной данной и проходящей через начало координат.

Ответ:

$$A \cdot x + B \cdot y + C \cdot z = 0$$

В. Пусть задана плоскость:

$$A_1 \cdot x + B_1 \cdot y + C_1 \cdot z + D_1 = 0$$

и прямая:

$$\frac{x - x_1}{x - x_2} = \frac{y - y_1}{y - y_2} = \frac{z - z_1}{z - z_2}$$

Как узнать, принадлежит прямая плоскости или нет?

Ответ: Возьмём две точки:

$$Q(x - x_1, y - y_1, z - z_1), R(x_2 - x_1, y_2 - y_1, z_2 - z_1)$$

и подставим их в уравнение:

$$A_1 \cdot (x - x_1) + B_1 \cdot (y - y_1) + C_1 \cdot (z - z_1) + D_1 = 0$$

$$A_1 \cdot (x_2 - x_1) + B_1 \cdot (y_2 - y_1) + C_1 \cdot (z_2 - z_1) + D_1 = 0$$

если оба тождества верны, то прямая принадлежит плоскости, если верно одно - то прямая пересекает плоскость, если оба не верны - прямая параллельна плоскости.

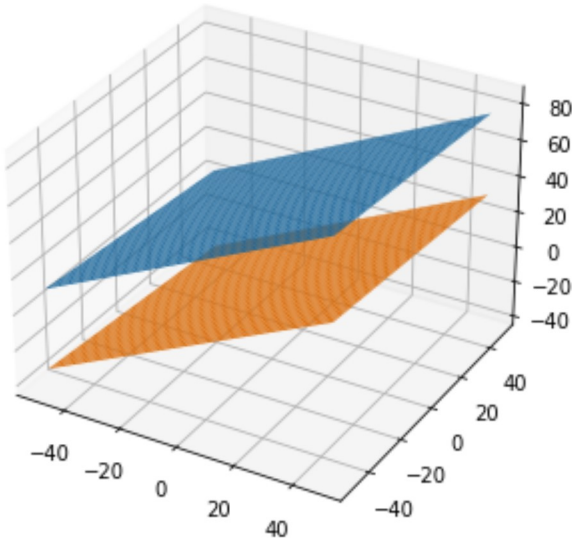
5. Задание (в программе)
- А. Нарисуйте трехмерный график двух параллельных плоскостей.
- В. Нарисуйте трехмерный график двух любых поверхностей второго порядка

In [186]: 

```
from pylab import *
from mpl_toolkits.mplot3d import Axes3D
fig = figure()
ax = Axes3D(fig)
X = np.arange(-50, 50, 1)
Y = np.arange(-50, 50, 1)
X, Y = np.meshgrid(X, Y)
Z1 = (-2*Y + 15*X)/20+45
Z2 = (-2*Y + 15*X)/20
ax.plot_surface(X, Y, Z1)
ax.plot_surface(X, Y, Z2)
show()
```

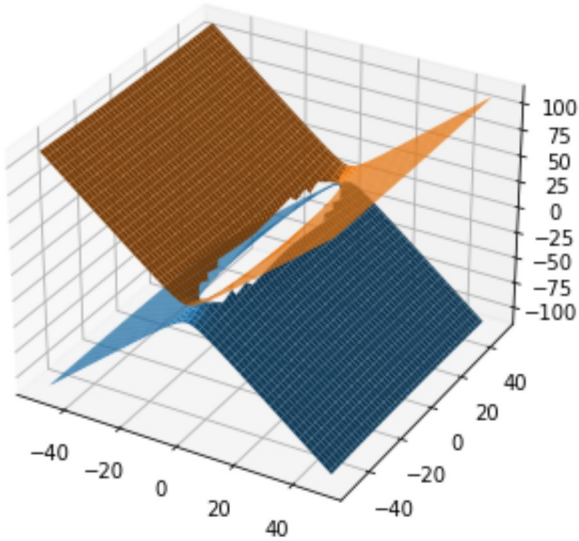
C:\Users\akins\AppData\Local\Temp\ipykernel\_17672\1078781180.py:4: MatplotlibDeprecationWarning: Axes3D(fig) adding it self to the figure is deprecated since 3.4. Pass the keyword argument auto\_add\_to\_figure=False and use fig.add\_axes(ax) to suppress this warning. The default value of auto\_add\_to\_figure will change to False in mpl3.5 and True values will no longer work in 3.6. This is consistent with other Axes classes.

ax = Axes3D(fig)



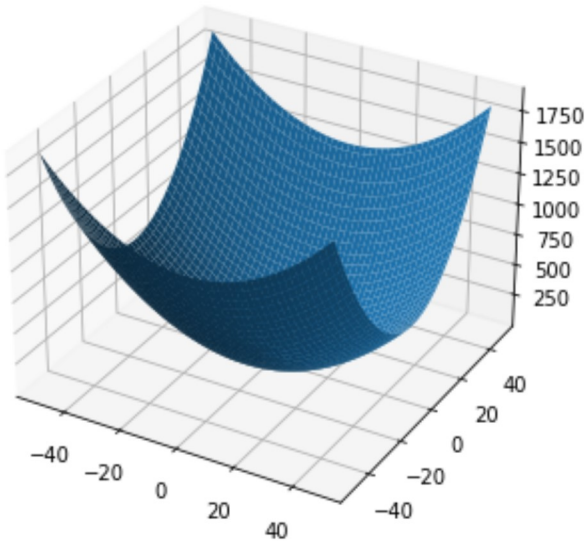
```
In [193]: from pylab import *
from mpl_toolkits.mplot3d import Axes3D
fig = figure()
ax = Axes3D(fig)
X = np.arange(-50, 50)
Y = np.arange(-50, 50)
X, Y = np.meshgrid(X, Y)
Z2 = np.sqrt(((5*X**2/5**2) + (7*Y**2/42**2) - 7)*5**2)
Z1 = -(np.sqrt(((5*X**2/5**2) + (7*Y**2/42**2) - 7)*5**2))
ax.plot_surface(X, Y, Z1)
ax.plot_surface(X, Y, Z2)
show()

C:\Users\akins\AppData\Local\Temp\ipykernel_17672\1673284201.py:4: MatplotlibDeprecationWarning: Axes3D(fig) adding it
self to the figure is deprecated since 3.4. Pass the keyword argument auto_add_to_figure=False and use fig.add_axes(a
x) to suppress this warning. The default value of auto_add_to_figure will change to False in mpl3.5 and True values wi
ll no longer work in 3.6. This is consistent with other Axes classes.
    ax = Axes3D(fig)
C:\Users\akins\AppData\Local\Temp\ipykernel_17672\1673284201.py:8: RuntimeWarning: invalid value encountered in sqrt
    Z2 = np.sqrt(((5*X**2/5**2) + (7*Y**2/42**2) - 7)*5**2)
C:\Users\akins\AppData\Local\Temp\ipykernel_17672\1673284201.py:9: RuntimeWarning: invalid value encountered in sqrt
    Z1 = -(np.sqrt(((5*X**2/5**2) + (7*Y**2/42**2) - 7)*5**2))
```



```
In [196]: from pylab import *
from mpl_toolkits.mplot3d import Axes3D
fig = figure()
ax = Axes3D(fig)
X = np.arange(-50, 50, 1)
Y = np.arange(-50, 50, 1)
X, Y = np.meshgrid(X, Y)
Z = (4*X**2/4**2) + (2*Y**2/2**2) + 20
ax.plot_surface(X, Y, Z)
show()

C:\Users\akins\AppData\Local\Temp\ipykernel_17672\865703166.py:4: MatplotlibDeprecationWarning: Axes3D(fig) adding its
elf to the figure is deprecated since 3.4. Pass the keyword argument auto_add_to_figure=False and use fig.add_axes(ax)
to suppress this warning. The default value of auto_add_to_figure will change to False in mpl3.5 and True values will
no longer work in 3.6. This is consistent with other Axes classes.
    ax = Axes3D(fig)
```



```
In [ ]: 
```