

In [1]:

```
%matplotlib inline
import numpy as np
from math import sqrt
from array import *
import matplotlib.pyplot as plt
import math
import matplotlib.ticker as ticker
from pylab import *
from mpl_toolkits.mplot3d import Axes3D
```

Задания к уроку 2

1. Задание

Даны два вектора в трехмерном пространстве: (10,10,10) и (0,0,-10)

1) Найдите их сумму. (на листочке)

$$\begin{bmatrix} 10 \\ 10 \\ 10 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -10 \end{bmatrix} = \begin{bmatrix} 10 \\ 10 \\ 0 \end{bmatrix}$$

2) Напишите код на Python, реализующий расчет длины вектора, заданного его координатами. (в программе)

In [2]:

```
data = str(input("Введите координаты вектора через пробел ")).split()
print("Длина вектора, с введенными Вами координатами, равна: ", sqrt(sum([int(data[i])**2 for i in range(3)])))
```

Введите координаты вектора через пробел 4 5

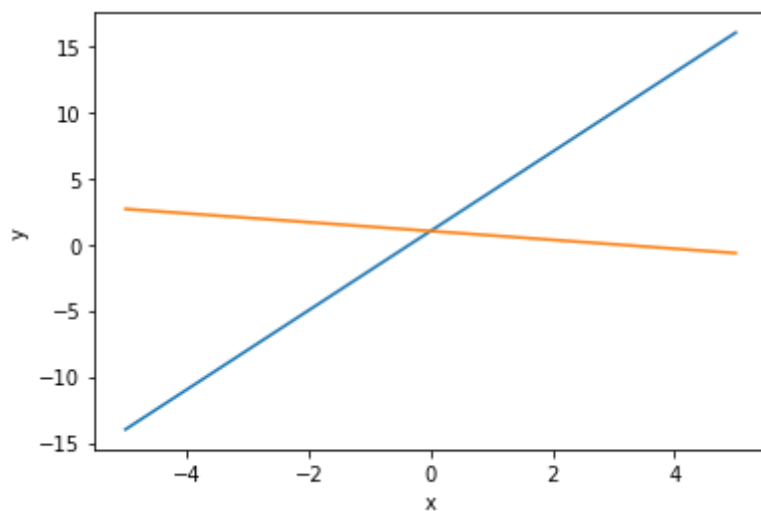
Длина вектора, с введенными Вами координатами, равна: 6.4031242374328485

2. Задание (на листочке)

Почему прямые не кажутся перпендикулярными? (см.ролик)

In [3]:

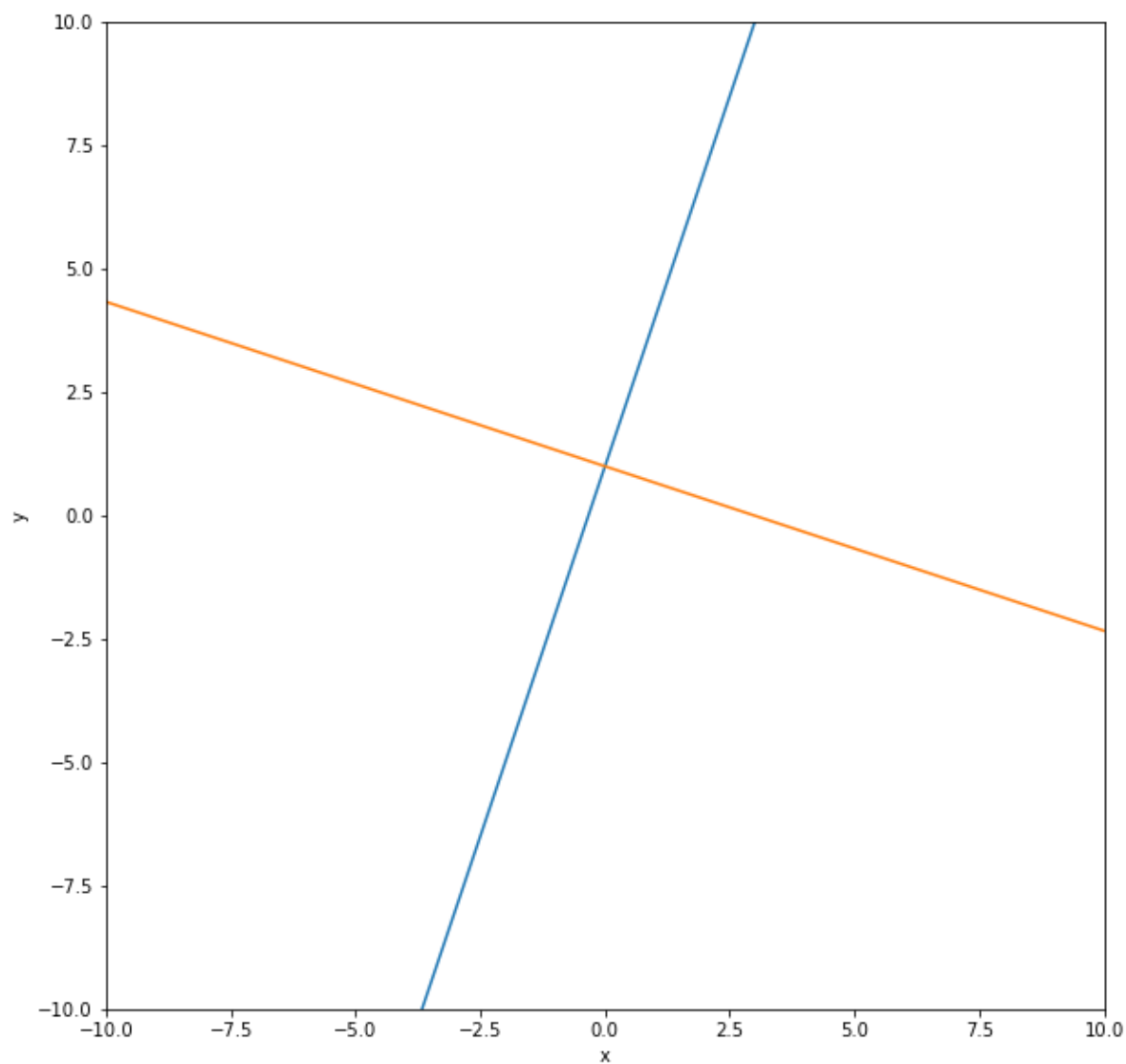
```
x = np.linspace(-5, 5, 21)
y = 3*x+1
y2 = (-1/3)*x+1
plt.plot(x,y)
plt.plot(x,y2)
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```



Это связано с тем, что цена деления и его размер на осях x и y отличаются. Проверим это. Сделаем цену деления и его размер на экране одинаковыми на осях x и y .

In [4]:

```
x = np.linspace(-10, 10, 21)
y = 3*x+1
y2 = (-1/3)*x+1
plt.subplots(figsize=(9.85, 10))
plt.plot(x,y)
plt.plot(x,y2)
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```



Видим, что прямые стали перпендикулярными.

3. Задание (в программе)

Напишите код на Python, реализующий построение графиков:

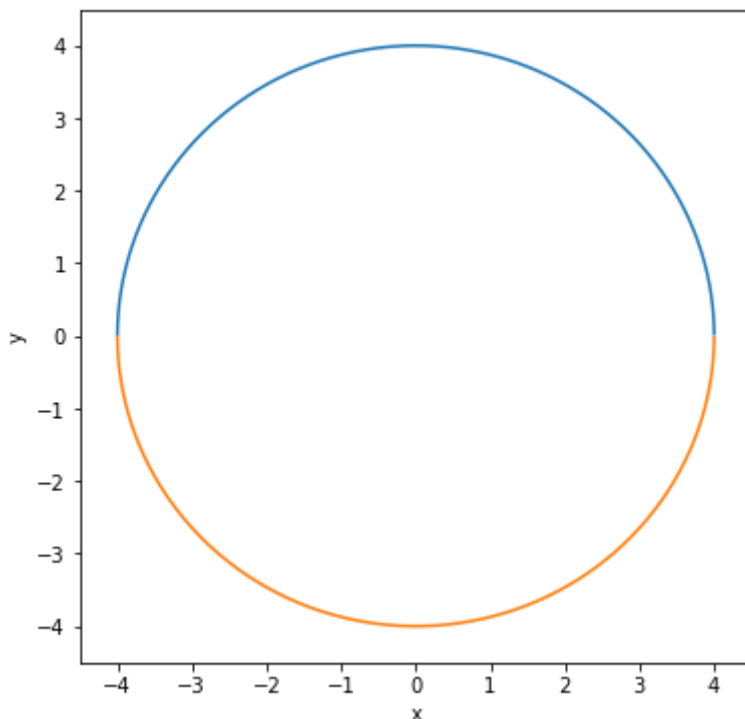
1. окружности

In [5]:

```
x = []
y1 = []
y2 = []
R = int(input("Введите радиус окружности, которую Вы хотите построить: "))
for i in range(-99999, 99999):
    x_ = i/100000*R
    x.append(x_)
    y1.append(sqrt(R**2-(x_**2)))
    y2.append(-sqrt(R**2-(x_**2)))

plt.subplots(figsize=(R+2, R+2))
plt.plot(x,y1)
plt.plot(x,y2)
plt.xlim(-R-0.5, R+0.5)
plt.ylim(-R-0.5, R+0.5)
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```

Введите радиус окружности, которую Вы хотите построить: 4



2. эллипса

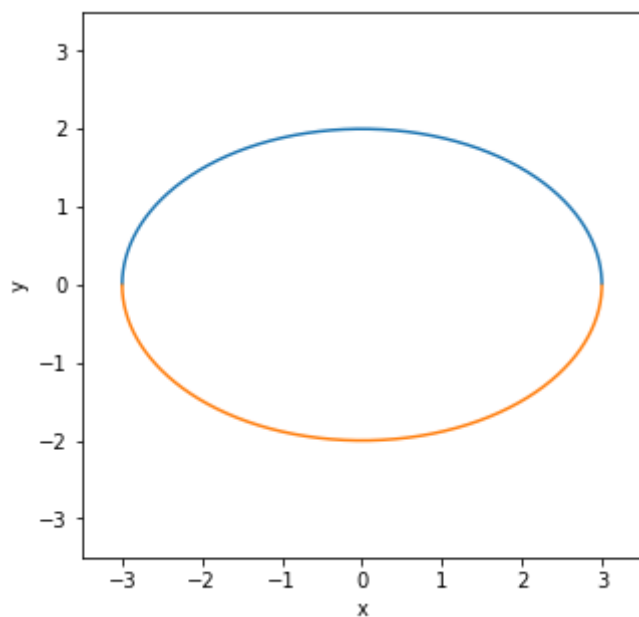
In [6]:

```
x = []
y1 = []
y2 = []
a = int(input("Введите большую полуось - a "))
b = int(input("Введите малую полуось - b "))
for i in range(-99999, 99999):
    x_ = i/100000*a
    x.append(x_)
    y1.append(b*sqrt(1-(x_**2/a**2)))
    y2.append(-b*sqrt(1-(x_**2/a**2)))

plt.subplots(figsize=(a+2, a+2))
plt.plot(x,y1)
plt.plot(x,y2)
plt.xlim(-a-0.5, a+0.5)
plt.ylim(-a-0.5, a+0.5)
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```

Введите большую полуось - a 3

Введите малую полуось - b 2



3. гиперболы

In [7]:

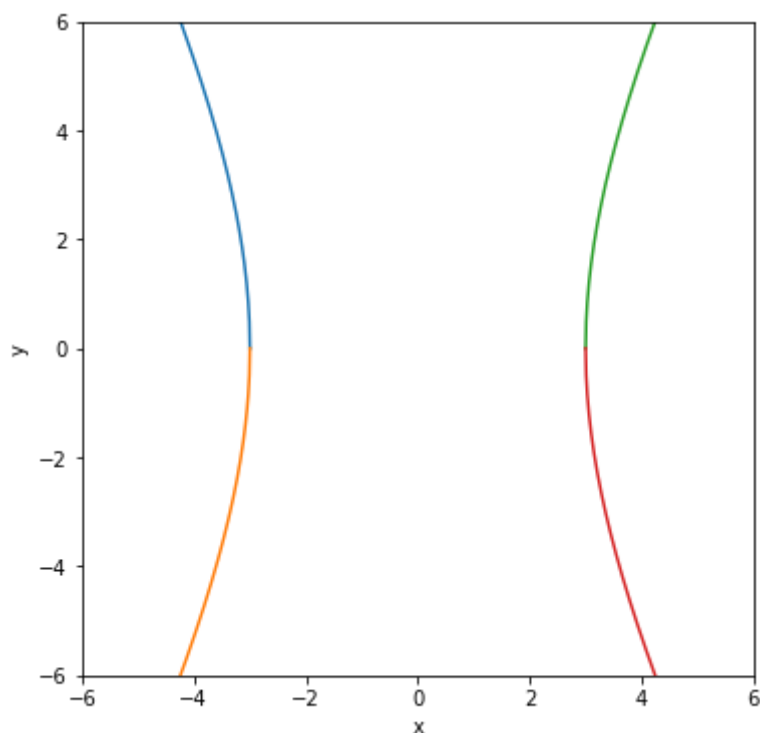
```
x = []
x1 = []
y1 = []
y2 = []
y3 = []
y4 = []
a = float(input("Введите большую полуось - a: "))
b = float(input("Введите прицельный параметр - b: "))
for i in np.arange(-a-5, -a+0.01, 0.01):
    x.append(i)
    y1.append(b*sqrt(i**2/a**2-1))
    y2.append(-b*sqrt(i**2/a**2-1))

for i in np.arange(a, a + 5, 0.01):
    x1.append(i)
    y3.append(b*sqrt(i**2/a**2-1))
    y4.append(-b*sqrt(i**2/a**2-1))

plt.subplots(figsize=(abs(a)+3, abs(a)+3))
plt.plot(x,y1)
plt.plot(x,y2)
plt.plot(x1,y3)
plt.plot(x1,y4)
plt.xlim(-a-3, a+a)
plt.ylim(-a-3, a+a)
plt.xlabel("x")
plt.ylabel("y")
plt.show()
```

Введите большую полуось - a: 3

Введите прицельный параметр - b: 6



4. Задание (на листочке)

1) Пусть задана плоскость:

$$Ax + By + Cz + D = 0$$

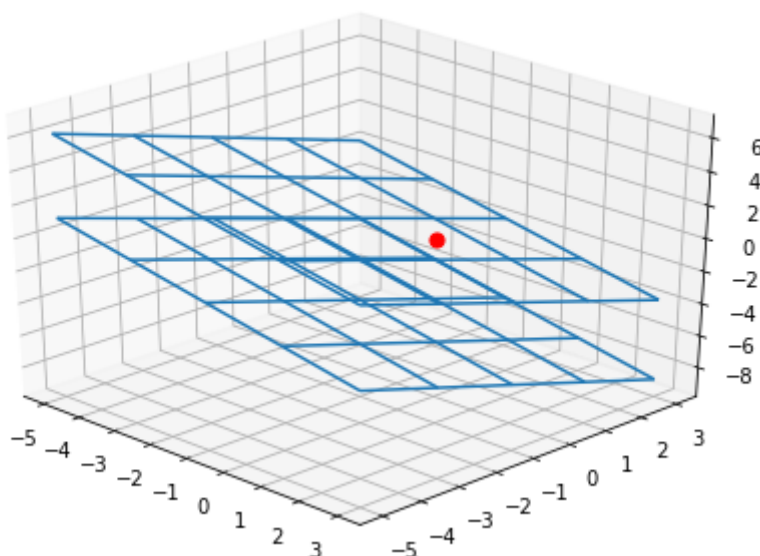
Напишите уравнение плоскости, параллельной данной и проходящей через начало координат.

Ответ: $Ax + By + Cz = 0$

Проверим ответ, построив две эти плоскости с коэффициентами $A=2$, $B=3$, $C=4$ и $D=20$

In [8]:

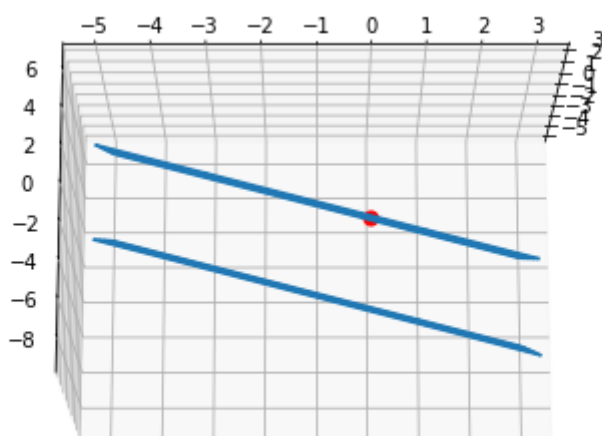
```
fig = figure()
ax = Axes3D(fig)
ax.view_init(30, -45)
X = np.arange(-5, 5, 2)
Y = np.arange(-5, 5, 2)
X, Y = np.meshgrid(X, Y)
Z = (-2*X - 3*Y - 20)/4
Z1 = (-2*X - 3*Y)/4
ax.plot_wireframe(X, Y, Z)
ax.plot_wireframe(X, Y, Z1)
ax.scatter(0, 0, 0, 'z', 50, 'red')
show()
```



Посмотрим те же плоскости под другим углом.

In [9]:

```
fig = figure()
ax = Axes3D(fig)
ax.view_init(-15, 0)
X = np.arange(-5, 5, 2)
Y = np.arange(-5, 5, 2)
X, Y = np.meshgrid(X, Y)
Z = (-2*X - 3*Y - 20)/4
Z1 = (-2*X - 3*Y)/4
ax.plot_wireframe(X, Y, Z)
ax.plot_wireframe(X, Y, Z1)
ax.scatter(0, 0, 0, 'z', 50, 'red')
show()
```



Видим, что вторая плоскость проходит через начало координат.

2) Пусть задана плоскость: $A_1x + B_1y + C_1z + D_1 = 0$

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

Как узнать, принадлежит прямая плоскости или нет?

Ответ:

Прямая проходит через две точки (x_1, y_1) и (x_2, y_2) . Если плоскость проходит через эти же 2 точки, то прямая принадлежит плоскости. Плоскость будет проходить через заданные точки только при условии выполнения системы уравнений:

$$\begin{cases} A_1(x-x_1)+B_1(y-y_1)+C_1(z-z_1)+D_1=0 \\ A_1(x_2-x_1)+B_1(y_2-y_1)+C_1(z_2-z_1)+D_1=0 \end{cases}$$

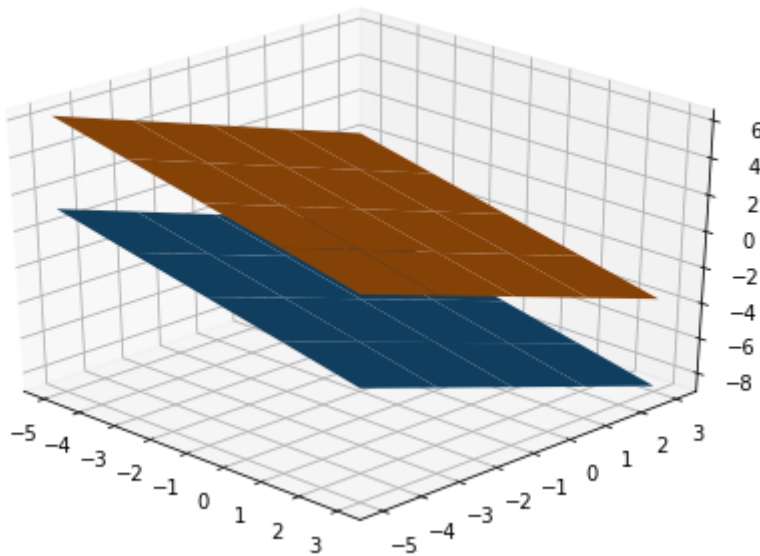
Таким образом, если данная система имеет решение, то прямая принадлежит плоскости

5. Задание (в программе)

1) Нарисуйте трехмерный график двух параллельных плоскостей.

In [10]:

```
fig = figure()
ax = Axes3D(fig)
ax.view_init(30, -45)
X = np.arange(-5, 5, 2)
Y = np.arange(-5, 5, 2)
X, Y = np.meshgrid(X, Y)
Z = (-2*X - 3*Y - 20)/4
Z1 = (-2*X - 3*Y)/4
ax.plot_surface(X, Y, Z)
ax.plot_surface(X, Y, Z1)
show()
```



2) Нарисуйте трехмерный график двух любых поверхностей второго порядка.

In [11]:

```
fig = figure()
ax = Axes3D(fig)
ax.view_init(30, -45)
X = np.arange(-100, 100, 10)
Y = np.arange(-50, 50, 10)
X, Y = np.meshgrid(X, Y)
Z = (-2*X**2 - 3*Y**2 - 20)/4
Z1 = (2*X**2 + 3*Y)/4
ax.plot_surface(X, Y, Z)
ax.plot_surface(X, Y, Z1)
show()
```

