

Задание 1

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

- Найдите их сумму. (на листочке)
- Напишите код на Python, реализующий расчет длины вектора, заданного его координатами. (в программе)

$\text{len} = x_1(10,10,10) + x_2(0,0,-10) = (10,10,0)$

In [1]:

```
def lvec(x: list):  
    t = 0  
    for i in x:  
        t += i**2  
    return t**0.5
```

In [2]:

```
lvec([3, 4])
```

Out[2]:

5.0

Задание 2

Почему прямые не кажутся перпендикулярными?

In [3]:

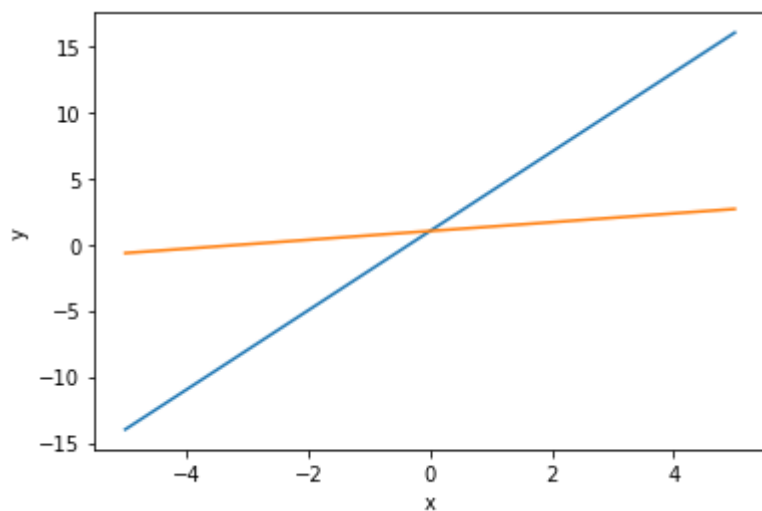
```
import numpy as np  
from matplotlib import pyplot as plt
```

In [4]:

```
from pylab import rcParams
```

In [5]:

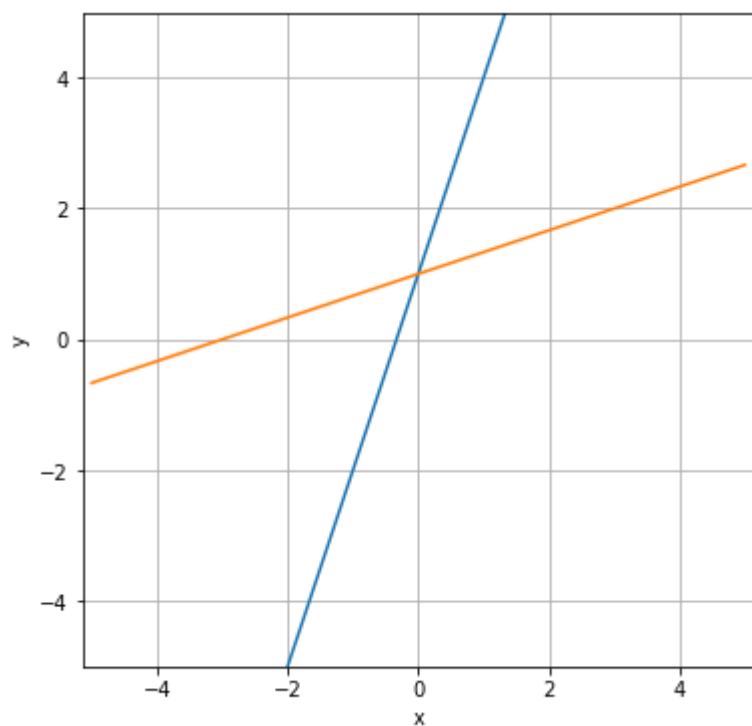
```
x = np.linspace(-5, 5, 21)
y1 = 3*x+1
y2 = (1/3)*x+1
plt.plot(x, y1)
plt.plot(x, y2)
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```



Ответ: гипотетически, из за разных масштабов осей

In [14]:

```
x = np.linspace(-5, 5, 21)
rcParams["figure.figsize"] = (6,6)
y1 = 3*x + 1
y2 = (1/3)*x + 1
plt.axis('equal')
plt.grid()
plt.xlim((-5, 5))
plt.ylim((-5, 5))
plt.plot(x, y1)
plt.plot(x, y2)
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```



с осями теперь все ок, но прямые все равно не перпендикулярны $\nabla(\text{ツ})\nabla$

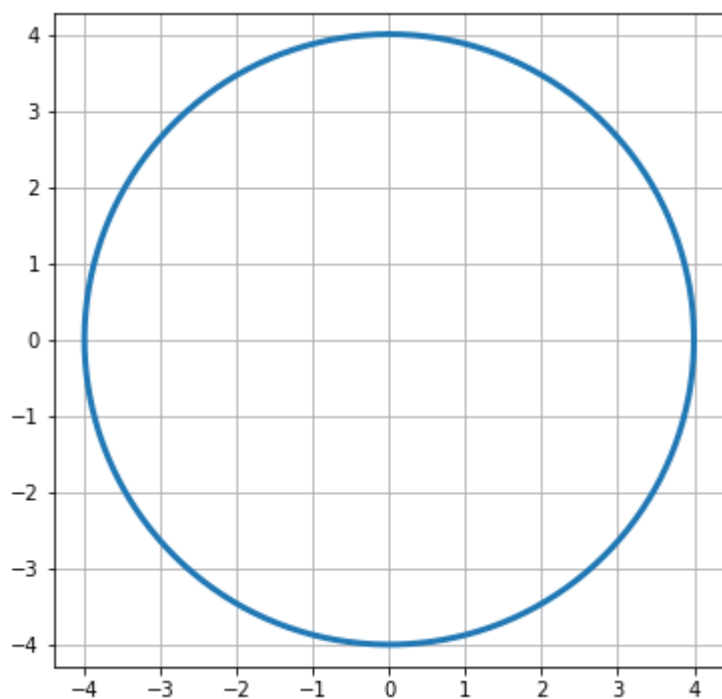
Задание 3

Напишите код на Python, реализующий построение графиков:

- окружности,
- эллипса,
- гиперболы.

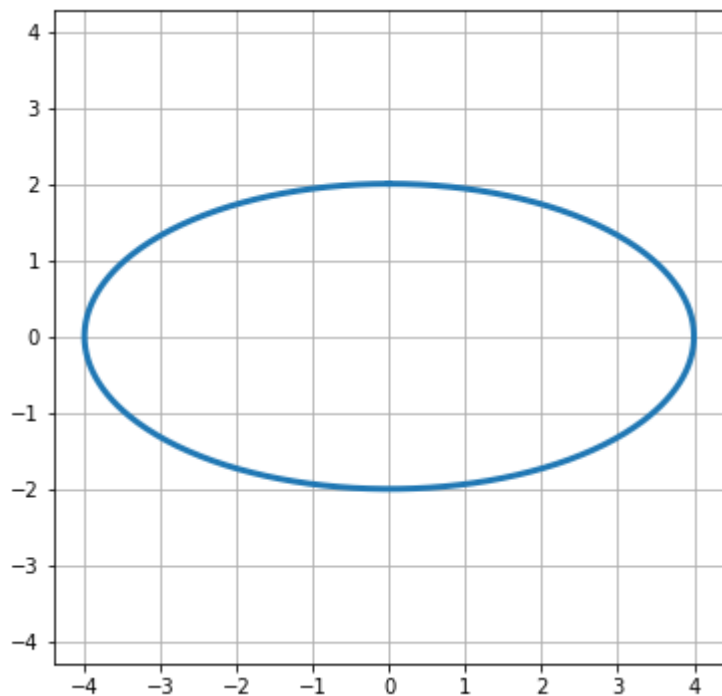
In [42]:

```
t = np.arange(0, 2*np.pi, 0.01)
r = 4
plt.grid()
plt.plot(r*np.sin(t), r*np.cos(t), lw=3)
plt.axis('equal')
plt.show()
```



In [43]:

```
t = np.arange(0, 2*np.pi, 0.01)
r = 4
plt.grid()
plt.plot(r*np.sin(t), 0.5*r*np.cos(t), lw=3)
plt.axis('equal')
plt.show()
```



In [37]:

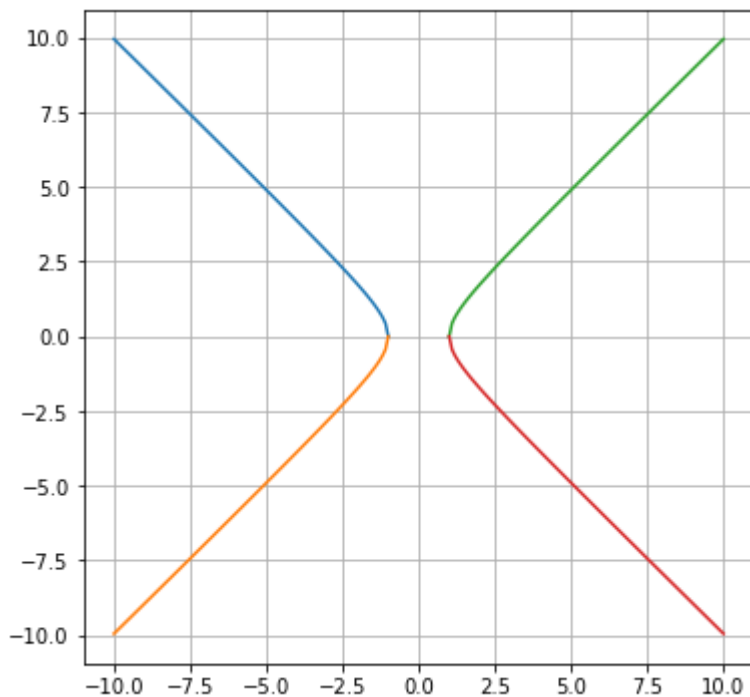
```
def hypY(x: int, a: int, b: int):
    return np.sqrt(np.abs(b**2*(x**2/a**2 - 1)))
```

In [41]:

```
x1 = np.linspace(-10, -1, 100)
x2 = np.linspace(1, 10, 100)
plt.grid()
plt.plot(x1, hypY(x1, -1, 1))
plt.plot(x1, -hypY(x1, -1, 1))
plt.plot(x2, hypY(x2, -1, 1))
plt.plot(x2, -hypY(x2, -1, 1))
```

Out[41]:

[<matplotlib.lines.Line2D at 0x17521945f60>]



Задание 4

1. Пусть задана плоскость: $Ax + By + Cz + D = 0$. Напишите уравнение плоскости, параллельной данной и проходящей через начало координат

Плоскость: $Ax + By + Cz = 0$ ($D = 0$)

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: подставить координаты прямой в уравнение плоскости

$$A_1x_1 + B_1y_1 + C_1z_1 + D_1 = A_1x_2 + B_1y_2 + C_1z_2 + D_1$$

Если равенство выполняется, значит прямая принадлежит плоскости

Задание 5

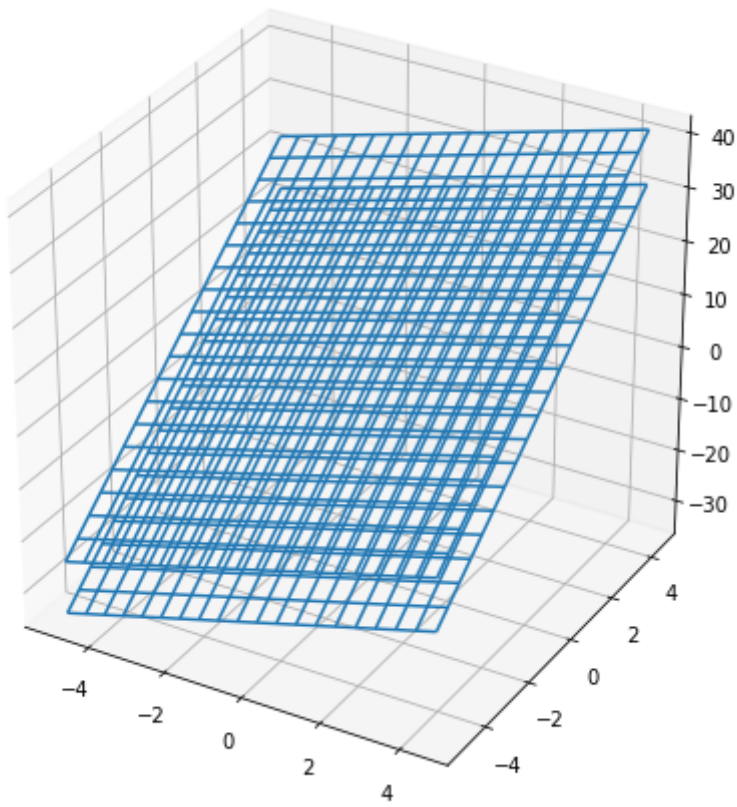
- Нарисуйте трехмерный график двух параллельных плоскостей.
- Нарисуйте трехмерный график двух любых поверхностей второго порядка.

In [48]:

```
from pylab import *  
from mpl_toolkits.mplot3d import Axes3D
```

In [49]:

```
fig = figure()
ax = Axes3D(fig)
X = np.arange(-5, 5, 0.5)
Y = np.arange(-5, 5, 0.5)
X, Y = np.meshgrid(X, Y)
Z1 = 2*X + 5*Y
Z2 = 2*X + 5*Y + 10
ax.plot_wireframe(X, Y, Z1)
ax.plot_wireframe(X, Y, Z2)
show()
```

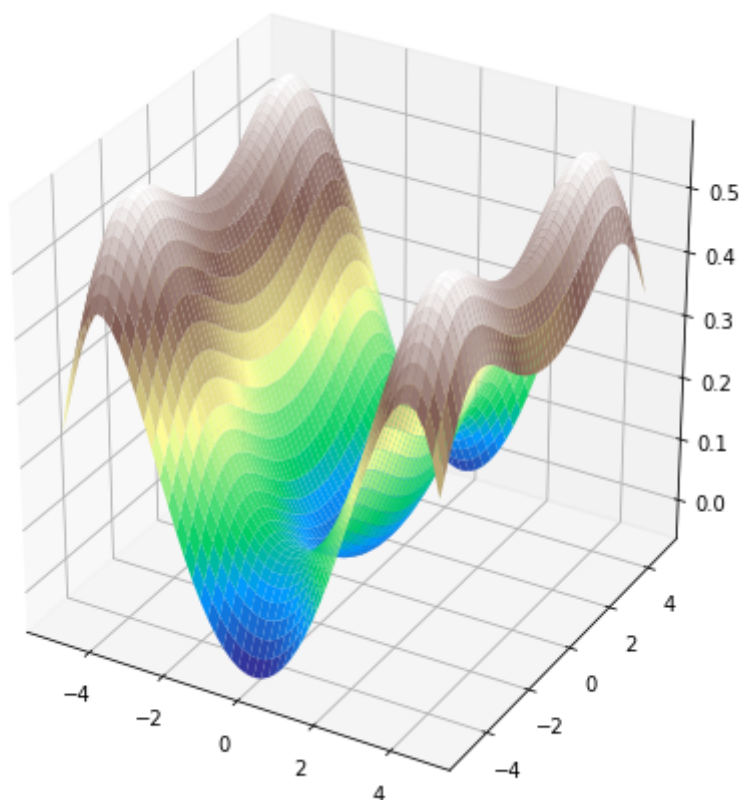



In [62]:

```
from matplotlib import cm
```

In [84]:

```
fig = figure()
ax = Axes3D(fig)
X = np.arange(-5, 5, 0.2)
Y = np.arange(-5, 5, 0.2)
X, Y = np.meshgrid(X, Y)
Z = 0.5*sin(0.1*X**2) + 0.09*sin(0.15*Y**2)
ax.plot_surface(X, Y, Z, cmap=cm.terrain)
show()
```

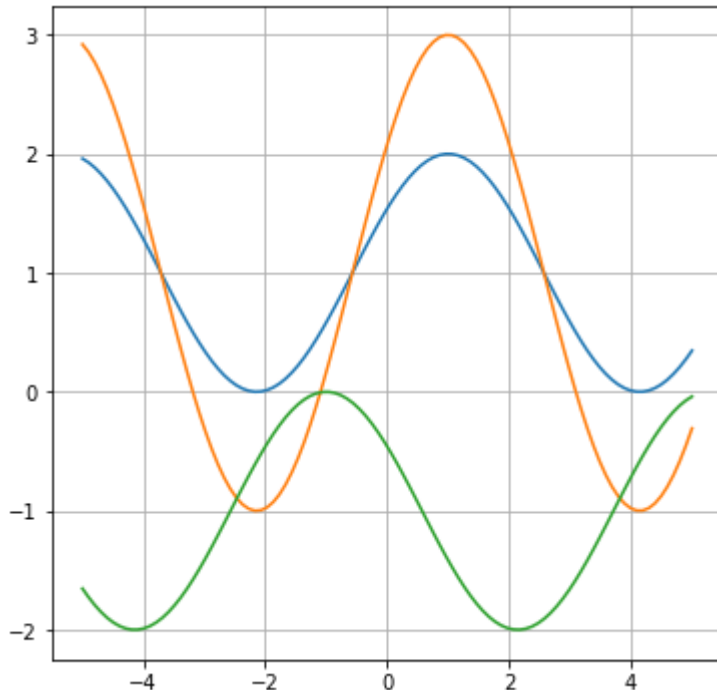


Задание 6

Нарисуйте график функции: $y(x) = k \cdot \cos(x - a) + b$ для некоторых (2-3 различных) значений параметров k , a , b

In [89]:

```
x = np.linspace(-5, 5, 100)
plt.grid()
plt.plot(x, np.cos(x - 1) + 1)
plt.plot(x, 2*np.cos(x - 1) + 1)
plt.plot(x, np.cos(x + 1) - 1)
plt.show()
```



Задание 7

Докажите, что при ортогональном преобразовании сохраняется расстояние между точками

□(ツ)□

Задание 8

- Напишите код, который будет переводить полярные координаты в декартовы.
- Напишите код, который будет рисовать график окружности в полярных координатах.
- Напишите код, который будет рисовать график отрезка прямой линии в полярных координатах.

In [90]:

```
def convPolar(r, a):
    return r*np.cos(a), r*np.sin(a)
```

In [92]:

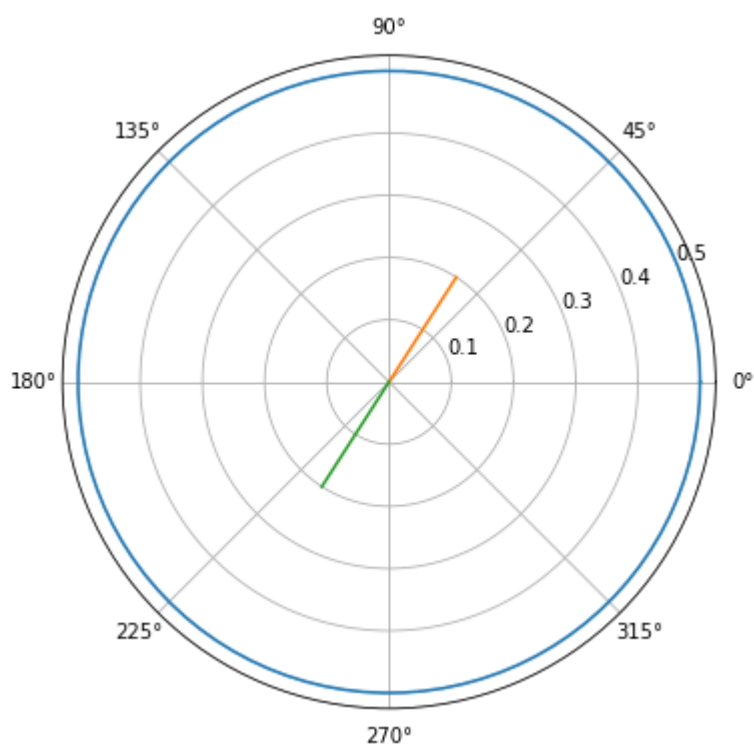
```
convPolar(1, np.pi/2)
```

Out[92]:

```
(6.123233995736766e-17, 1.0)
```

In [108]:

```
x = np.linspace(0, 2*np.pi, 100)
y = np.empty(100)
y.fill(0.5)
plt.polar(x, y)
r = np.array([0, 0.2])
a = np.array([1, 1])
b = np.array([1+np.pi, 1+np.pi])
plt.polar(a, r)
plt.polar(b, r)
plt.show()
```



Задание 9

1. Решите систему уравнений

$$\begin{cases} y = x^2 - 1 \\ e^x + x(1 - y) = 1 \end{cases}$$

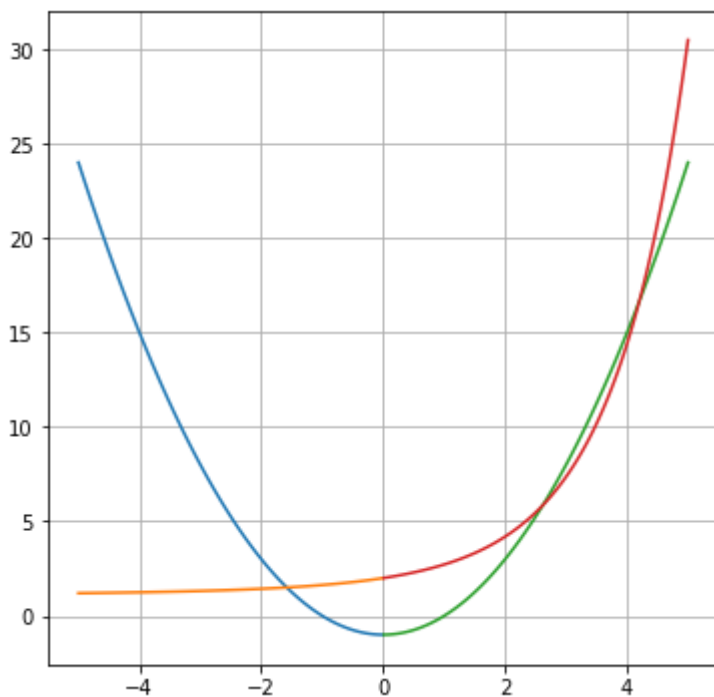
$$\begin{cases} y = x^2 - 1 \\ y = 1 - \frac{1 - e^x}{x} \end{cases}$$

In [131]:

```
def fy1(x):  
    return x**2 - 1  
  
def fy2(x):  
    return 1 - (1 - np.exp(x))/x
```

In [133]:

```
xx1 = np.linspace(-5, -0.01, 100)  
xx2 = np.linspace(0.01, 5, 100)  
plt.plot(xx1, fy1(xx1))  
plt.plot(xx1, fy2(xx1))  
plt.plot(xx2, fy1(xx2))  
plt.plot(xx2, fy2(xx2))  
plt.grid()  
plt.show()
```



In [117]:

```
from scipy.optimize import fsolve
```

In [134]:

```
def equations(p):  
    x, y = p  
    return (y - x**2 + 1, y-1+(1-np.exp(x))/x)
```

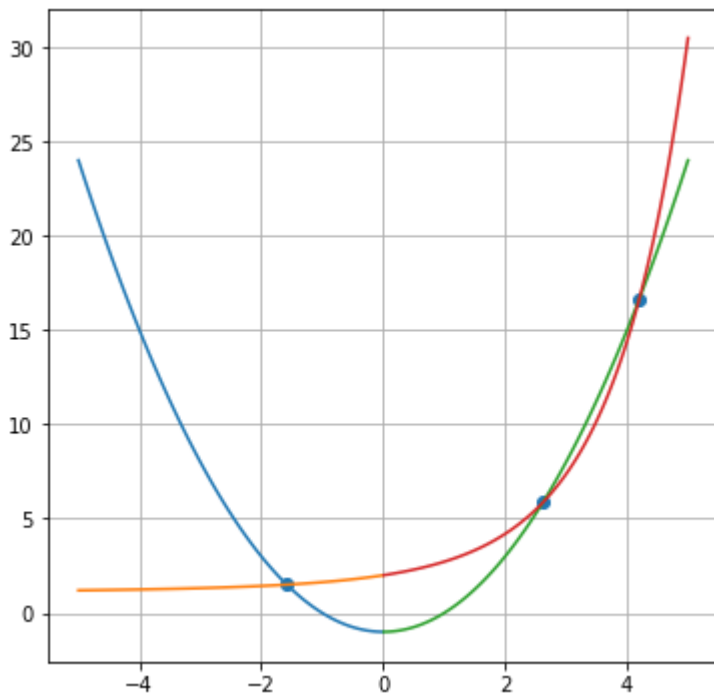
In [135]:

```
x1, y1 = fsolve(equations, (-4, 5))
x2, y2 = fsolve(equations, (3, 3))
x3, y3 = fsolve(equations, (5, 5))
print(x1, y1)
print(x2, y2)
print(x3, y3)
```

```
-1.5818353528959466 1.50220308367128
2.618145573085482 5.854686241867206
4.200105841166561 16.640889076993545
```

In [137]:

```
xx1 = np.linspace(-5, -0.01, 100)
xx2 = np.linspace(0.01, 5, 100)
plt.plot(xx1, fy1(xx1))
plt.plot(xx1, fy2(xx1))
plt.plot(xx2, fy1(xx2))
plt.plot(xx2, fy2(xx2))
plt.scatter([x1, x2, x3], [y1, y2, y3])
plt.grid()
plt.show()
```



1. Решите систему уравнений и неравенств

$$\begin{cases} y = x^2 - 1 \\ e^x + x(1 - y) > 1 \end{cases}$$

$$\begin{cases} y = x^2 - 1 \\ y > 1 - \frac{1 - e^x}{x} \end{cases}$$

¬(¬)¬(¬)¬(¬)¬