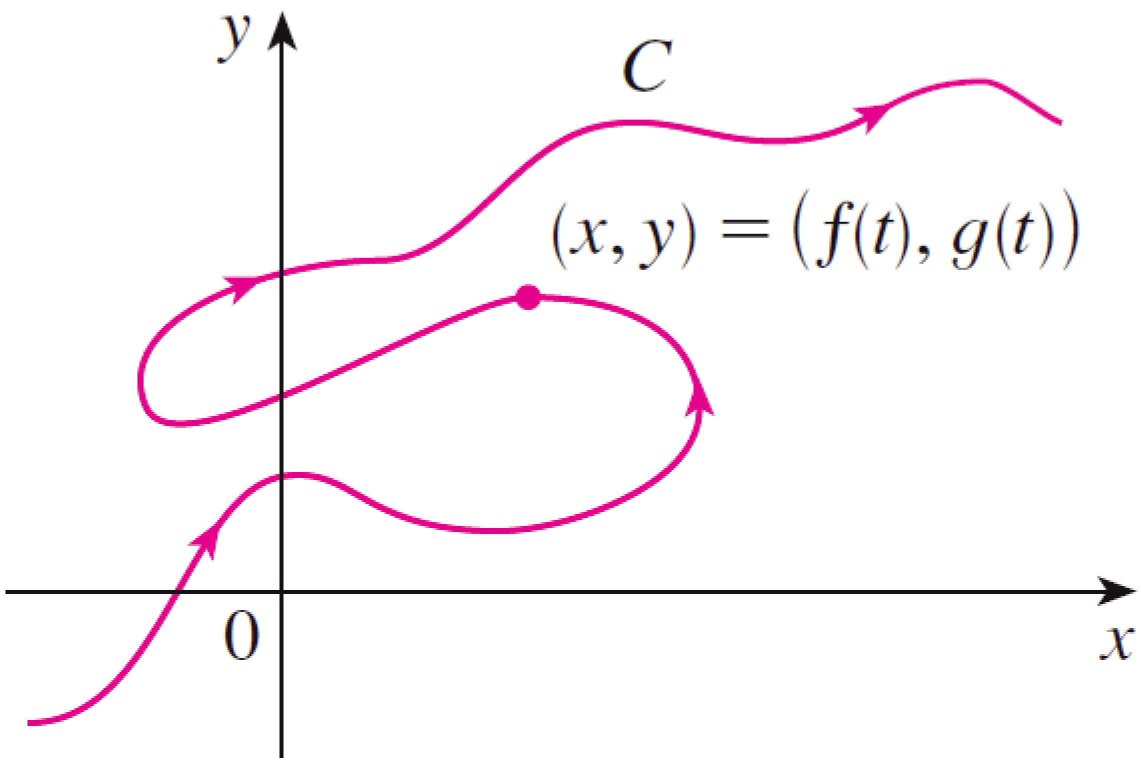
MAT150 - Summer 2023

Phd. Anabela Romina Turlione
Digipen, Bilbao

Parametric curves



In the previous graph,

- The curve C is not a function. It is imposible to decribe it as $y = f(x)$.
- But x and y are functions of the parameter t.

$$\left\{ \begin{array}{l} x = f(t) \\ y = g(t) \end{array} \right\} \tag{1}$$

- As t varies, the point $(x, y) = \big(f(t), g(t)\big)$ varies and traces out a curve C.

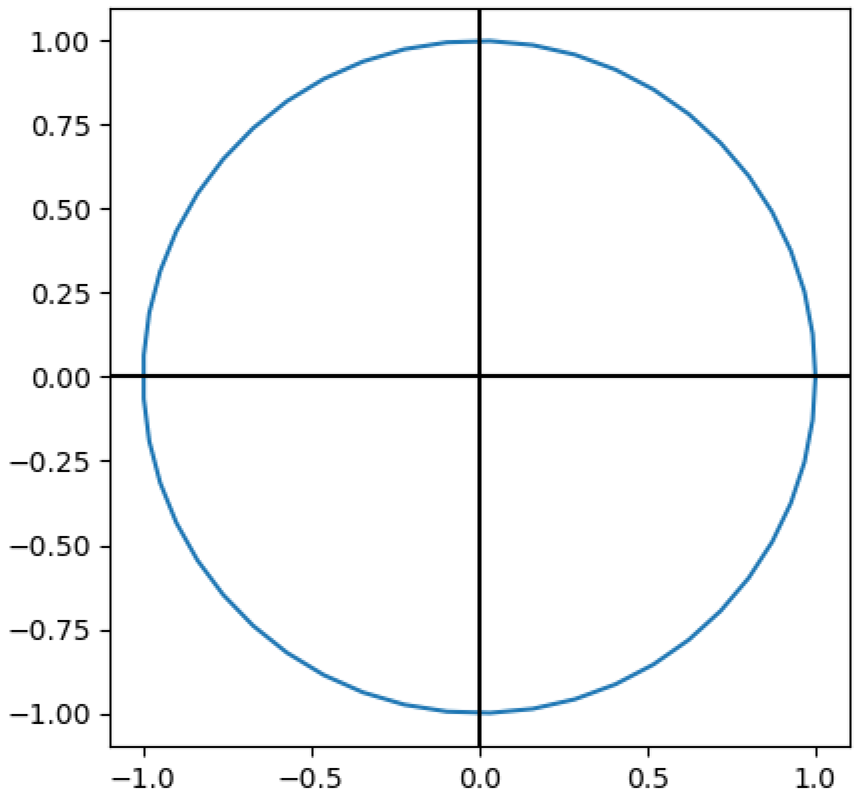
Equation for a Circle

Circle of unit radious:

$$\left\{ \begin{array}{l} x = \cos(t) \\ y = \sin(t) \end{array} \right., \ 0 \leq t \leq 2\pi \quad \Rightarrow x^2 + y^2 = 1 \tag{2}$$

As t increases from 0 to 2π, the point $(x, y) = (\cos t, \sin t)$ moves once around the circle in the counterclockwise direction starting from the point $(0, 1)$.

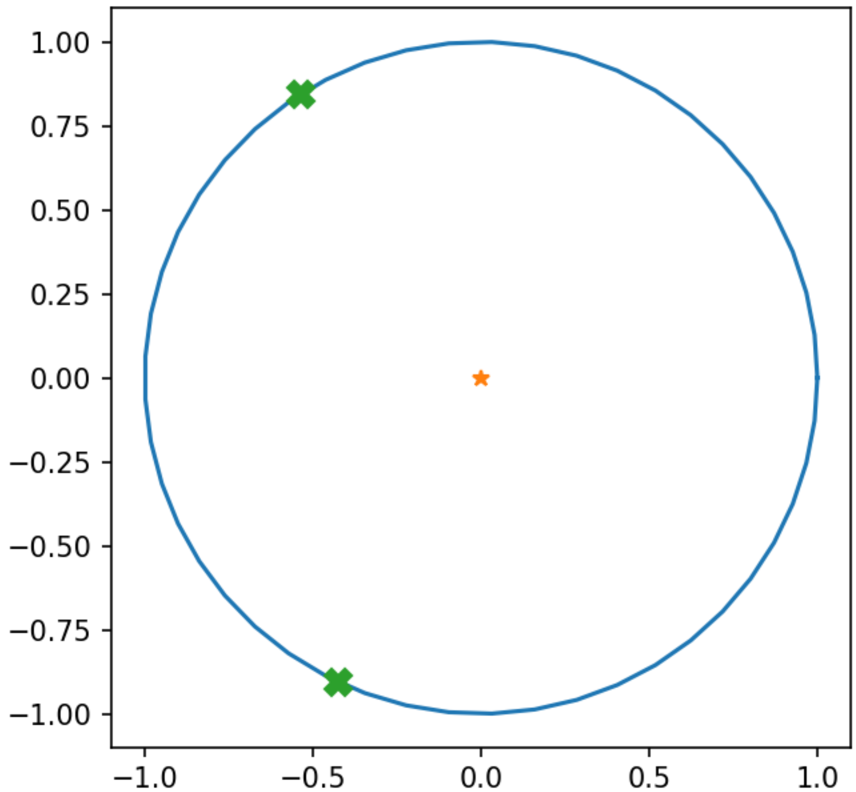
In [1]:
```python
import numpy as np
import matplotlib.pyplot as plt


t=np.linspace(0,2*np.pi)

plt.plot(np.cos(t),np.sin(t))
plt.axhline(y = 0, color = 'k', linestyle = '-')
plt.axvline(x = 0, color = 'k', linestyle = '-')

ax = plt.gca()
ax.set_aspect('equal')
```



Question: What happens if we change t by $2t$?

In [4]:
```python
# Enable interactive plot
%matplotlib notebook

import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation

fig, ax = plt.subplots()

t=np.linspace(0,2*np.pi)

ax.plot(np.cos(t),np.sin(t))
ax.plot(0,0,'*')
ax = plt.gca()
ax.set_aspect('equal')

line2, = ax.plot([],'X',markersize=10)     # A tuple unpacking to unpack the only plot
ax.set_xlim(-1.1, 1.1)
ax.set_ylim(-1.1, 1.1)

def animate(t):
    x = np.cos(2*np.pi * t/100)
    y = np.sin(2*np.pi * t/100)
```

```
    x1 = np.cos(2*np.pi * 2*t/100)
    y1 = np.sin(2*np.pi * 2*t/100)

    line2.set_data(([x,x1], [y,y1]))
    return line2

anim = FuncAnimation(fig, animate, frames=100, interval=20)
plt.show()
```



- Different sets of parametric equations can represent the same curve.
- Thus we distinguish between a **curve**, which is a set of points, and a **parametric curve**, in which the points are traced in a particular way.

General circle

Parametric equations for the circle with center (h, k) and radius r:

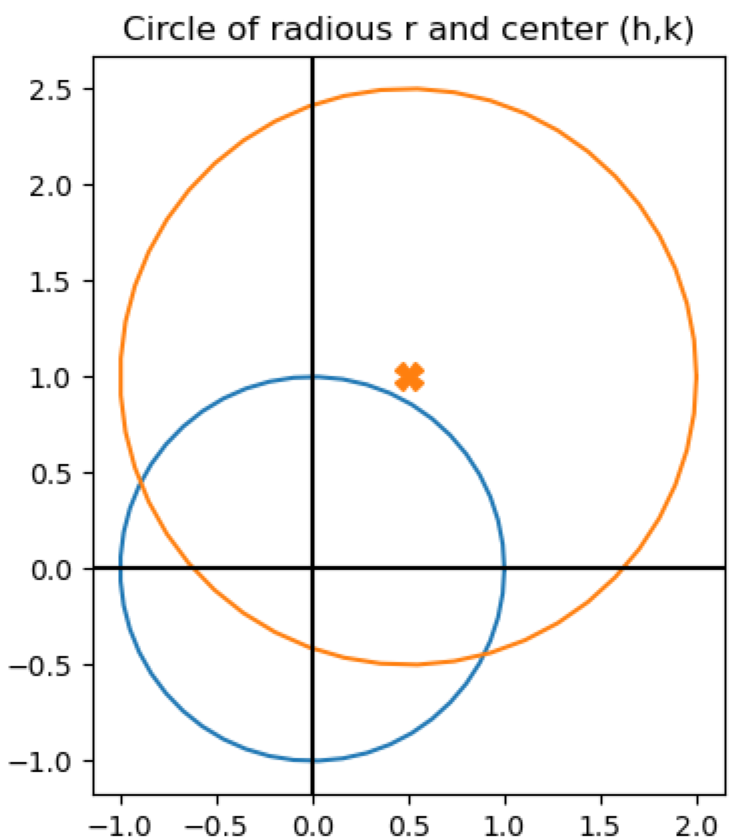
- Set of points (x, y) that satisfies: $\sqrt{(x-h)^2+(y-k)^2} = r \quad \Rightarrow \quad (x-h)^2+(y-k)^2 = r^2$
- A parametric representation may be given by:

$$\begin{cases} x-h = r\cos(\omega t) \\ y-k = r\sin(\omega t) \end{cases} \quad \Rightarrow \quad \begin{cases} x = r\cos(\omega t)+h \\ y = r\sin(\omega t)+k \end{cases}, \quad 0 \le t \le T \tag{3}$$

- $T = 2\pi/\omega$

In [8]:
```
t=np.linspace(0,2*np.pi)
```

```
r=1.5
k=1
h=0.5

plt.plot(np.cos(t),np.sin(t))
plt.plot(r*np.cos(t)+h,r*np.sin(t)+k)
plt.axhline(y = 0, color = 'k', linestyle = '-')
plt.axvline(x = 0, color = 'k', linestyle = '-')
plt.plot(h, k, marker="X", markersize=10, color="tab:orange")
plt.title('Circle of radious r and center (h,k)')
ax = plt.gca()
ax.set_aspect('equal')
```



What is the domain of a general circle?

- You can obtain the domain from the graph.
- Or from the study of its explicit functions.

$$\sqrt{(x-h)^2+(y-k)^2}=r \quad\Rightarrow\quad y=k\pm\sqrt{r^2-(x-h)^2} \tag{4}$$

$$\Rightarrow r^2-(x-h)^2>0 \Rightarrow |r|>|x-h| \tag{5}$$

Observation

Completing the square can be used as well in order to identify the radius or a circle and its center.

Ex: $x^2-2x+y^2=0$ is a circle.

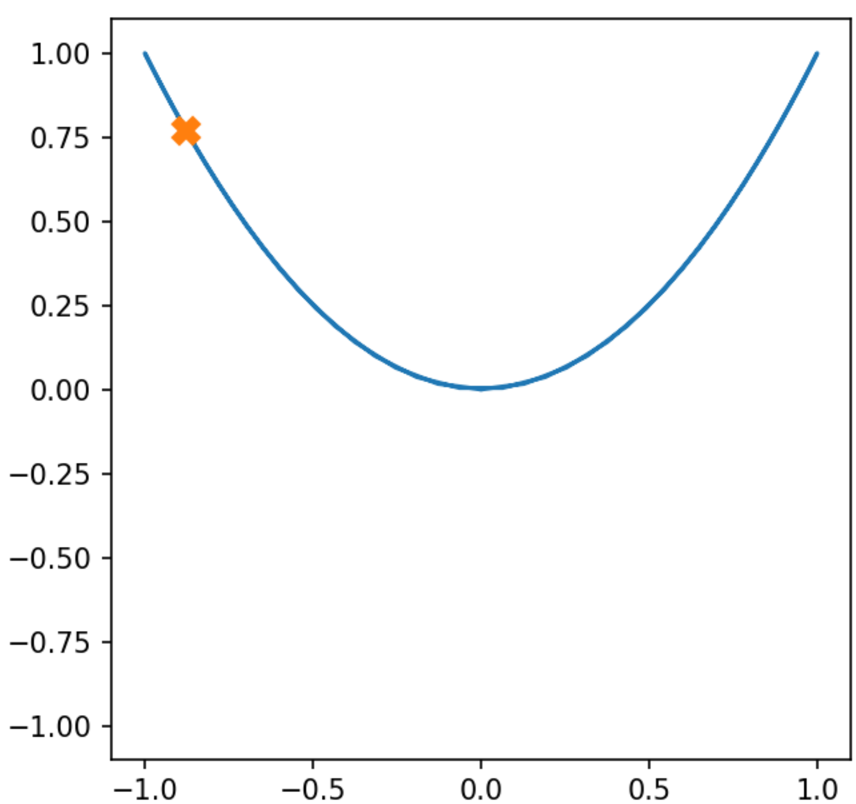
Example:

Which curve is:

$$\begin{cases} x = \sin(t) \\ y = \sin^2(t) \end{cases} \tag{6}$$

In [5]:

```
# Enable interactive plot
%matplotlib notebook

import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation

fig, ax = plt.subplots()

t=np.linspace(0,2*np.pi)

ax.plot(np.sin(t),np.sin(t)**2)
ax = plt.gca()
ax.set_aspect('equal')

line2, = ax.plot([],'X',markersize=10)      # A tuple unpacking to unpack the only plot
ax.set_xlim(-1.1, 1.1)
ax.set_ylim(-1.1, 1.1)

def animate(t):
    x = np.sin(2*np.pi *t/100)
    y = np.sin(2*np.pi *t/100)**2
    line2.set_data((x, y))
    return line2

anim = FuncAnimation(fig, animate, frames=100, interval=20)
plt.show()
```



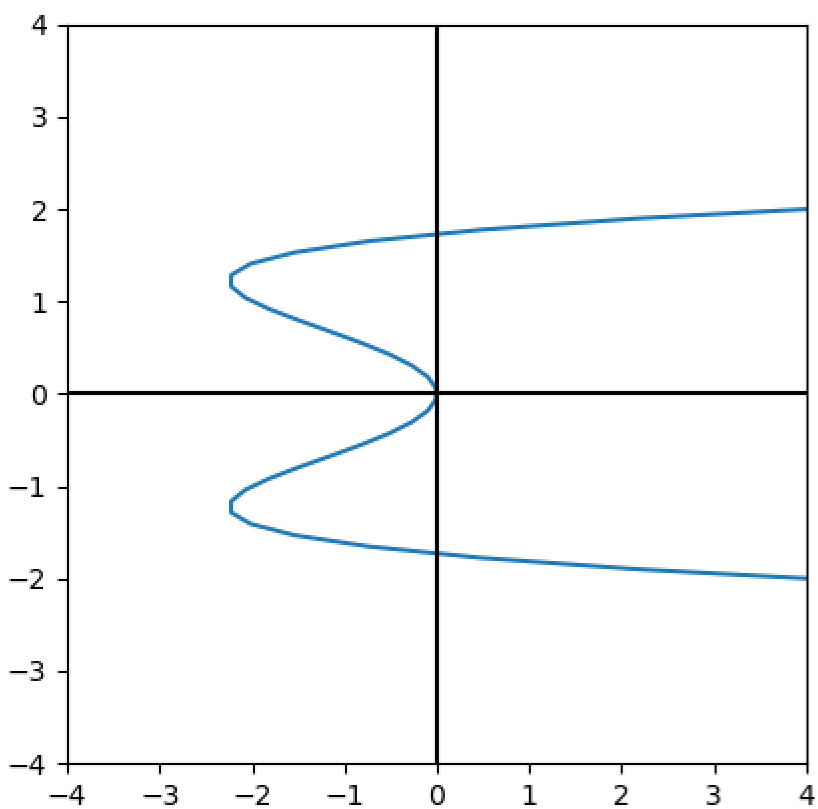
Example:

Graph the curve $x = y^4 - 3y^2$.

- You can solve the equation for y as four functions of x and graph them individually.
- You can transform the expression into a parametric equation:

$$\begin{cases} x = t^4 - 3t^2 \\ y = t \end{cases} \tag{7}$$

In [10]:
```
t=np.linspace(-3,3)

plt.plot(t**4-3*t**2,t)
plt.axhline(y = 0, color = 'k', linestyle = '-')
plt.axvline(x = 0, color = 'k', linestyle = '-')
plt.xlim([-4,4])
plt.ylim([-4,4])
ax = plt.gca()
ax.set_aspect('equal')
```



In General ...

- If we need to graph $x = g(y)$, we can use

$$\begin{cases} x = g(t) \\ y = t \end{cases} \tag{8}$$

- If we need to graph $y = f(x)$, we can use

$$\begin{cases} x = t \\ y = f(t) \end{cases} \tag{9}$$

More examples of parametric curves...

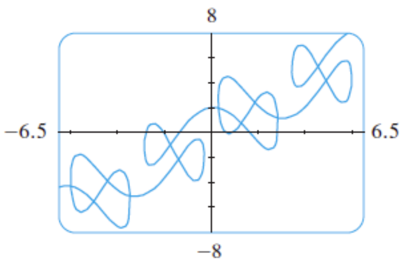


FIGURE 10
$x = t + 2\sin 2t$
$y = t + 2\cos 5t$

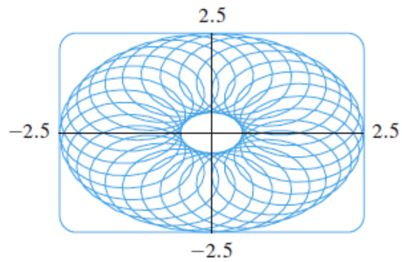


FIGURE 11
$x = 1.5\cos t - \cos 30t$
$y = 1.5\sin t - \sin 30t$

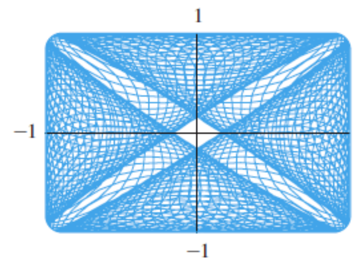
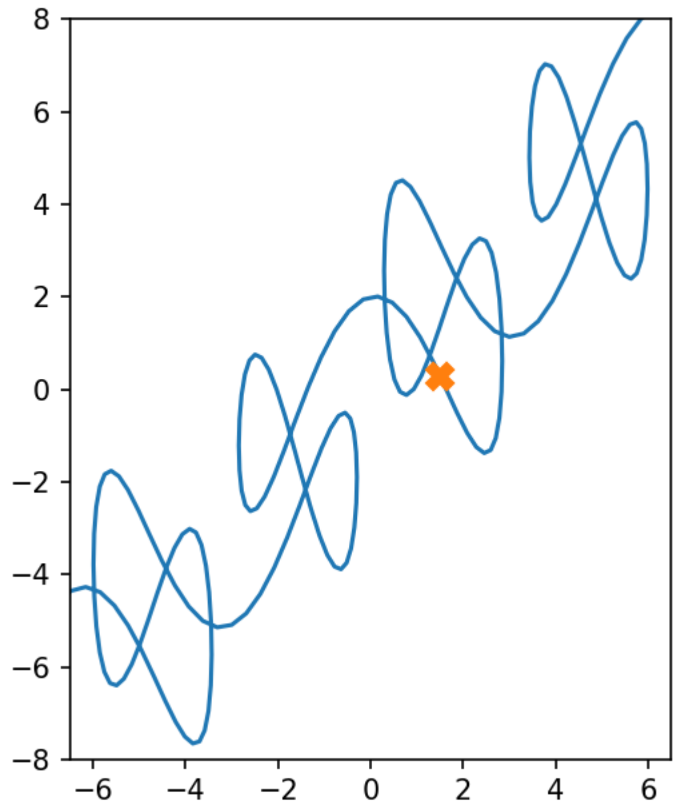


FIGURE 12
$x = \sin(t + \cos 100t)$
$y = \cos(t + \sin 100t)$

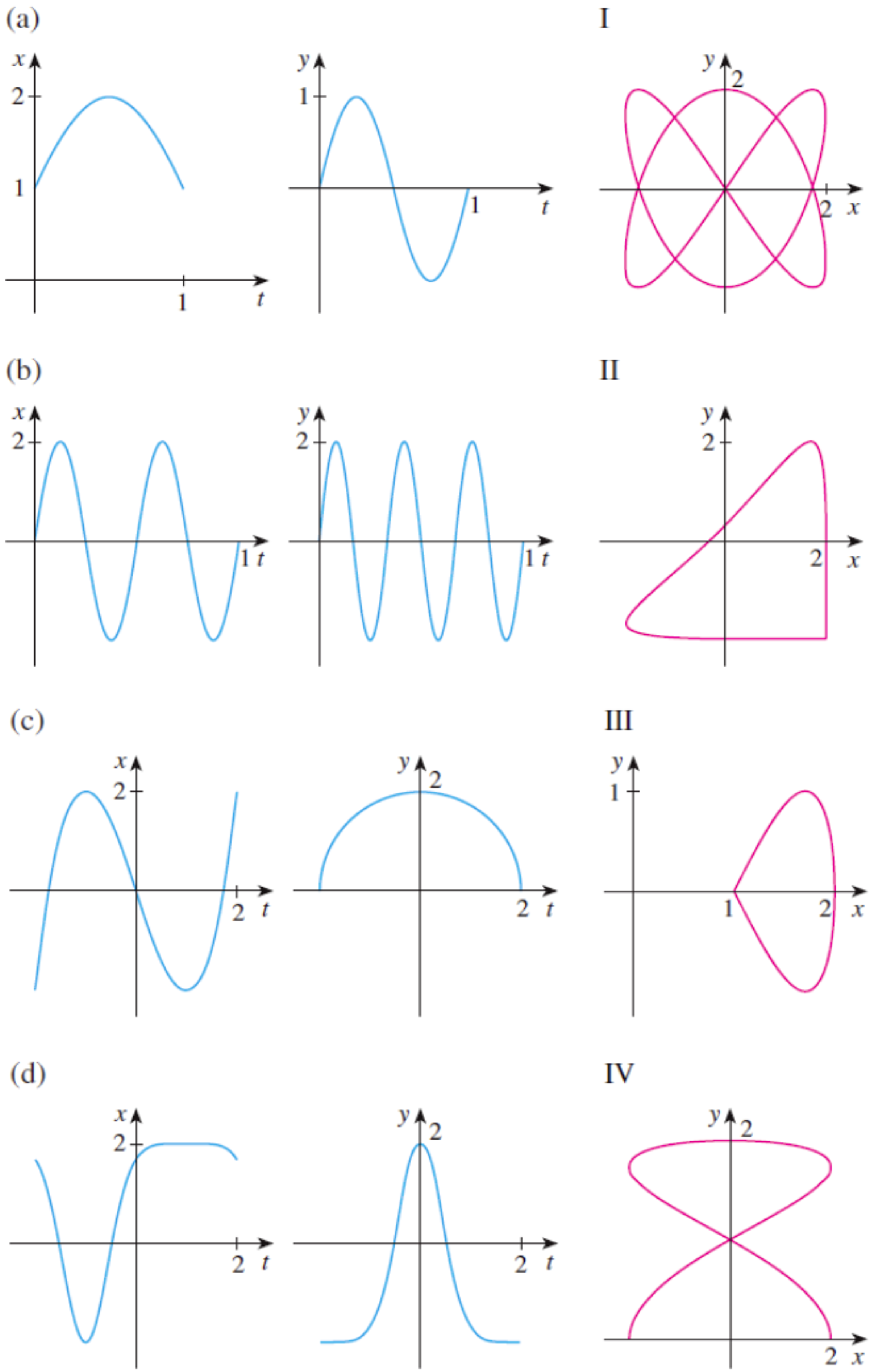
In [6]:

```
# Enable interactive plot
%matplotlib notebook

import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation

fig, ax = plt.subplots()

t=np.linspace(-10*np.pi,10*np.pi,1000)

ax.plot(t+2*np.sin(2*t),t+2*np.cos(5*t))
ax = plt.gca()
ax.set_aspect('equal')

line2, = ax.plot([],'X',markersize=10)      # A tuple unpacking to unpack the only plot
ax.set_xlim(-6.5,6.5)
ax.set_ylim(-8, 8)

def animate(t):
    t=-np.pi+t/50
    x = t+2*np.sin(2*t)
    y = t+2*np.cos(5*t)
    line2.set_data((x, y))
    return line2

anim = FuncAnimation(fig, animate, frames=1000, interval=20)
plt.show()
```



Exercise 1

Match the graphs of parametric equations $x = f(t)$ and $y = g(t)$, with the parametric curves labeled I-IV. Give reasons for your choice. Could you guess the direction of t? Add the arrow that represents t.



Exercise 2

Plot the following curves without a graphing calculator:

$$\begin{cases} y = t^2 - 2t \\ x = t + 1 \end{cases} \tag{10}$$

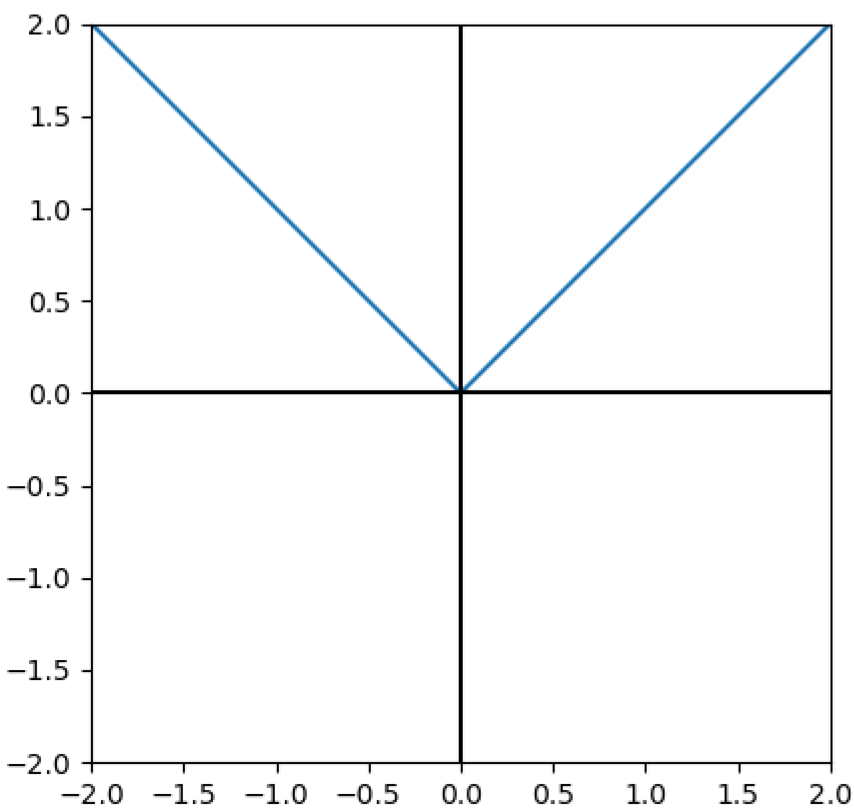
Piecewise Functions

The absolute value function

$$f(x) = |x| = \sqrt{x^2} = \begin{cases} -x, & x < 0 \\ x, & x \geq 0 \end{cases} \tag{11}$$

In [12]:

```
x1=np.linspace(0,3)
x2=np.linspace(-3,0)

plt.plot(x1,x1)
plt.plot(x2,-x2, color='tab:blue')

plt.axhline(y = 0, color = 'k', linestyle = '-')
plt.axvline(x = 0, color = 'k', linestyle = '-')
plt.xlim([-2,2])
plt.ylim([-2,2])
ax = plt.gca()
ax.set_aspect('equal')
```



Exercise 3:

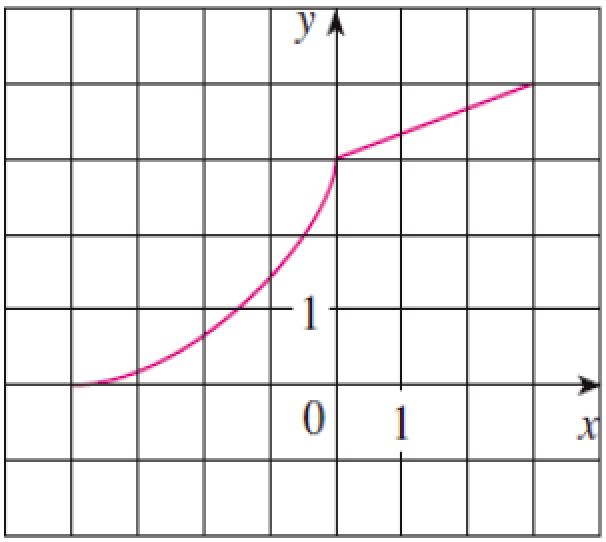
Write $y = |2x - 3|$ as a piecewise function and sketch it.

Exercise 4:

Solve the inequality: $|x-1|+|x-3|\geq 5$

Further exercises of the section

1. The graph of f is given. Draw the graphs of the following functions (Midterm spring 2015).



a. $y=f(2x)$

b. $y=f^{-1}(x)$

c. $y=\frac{1}{2}f(x)-1$

1. Sketch the following functions:

a. $y=\sin x+x$

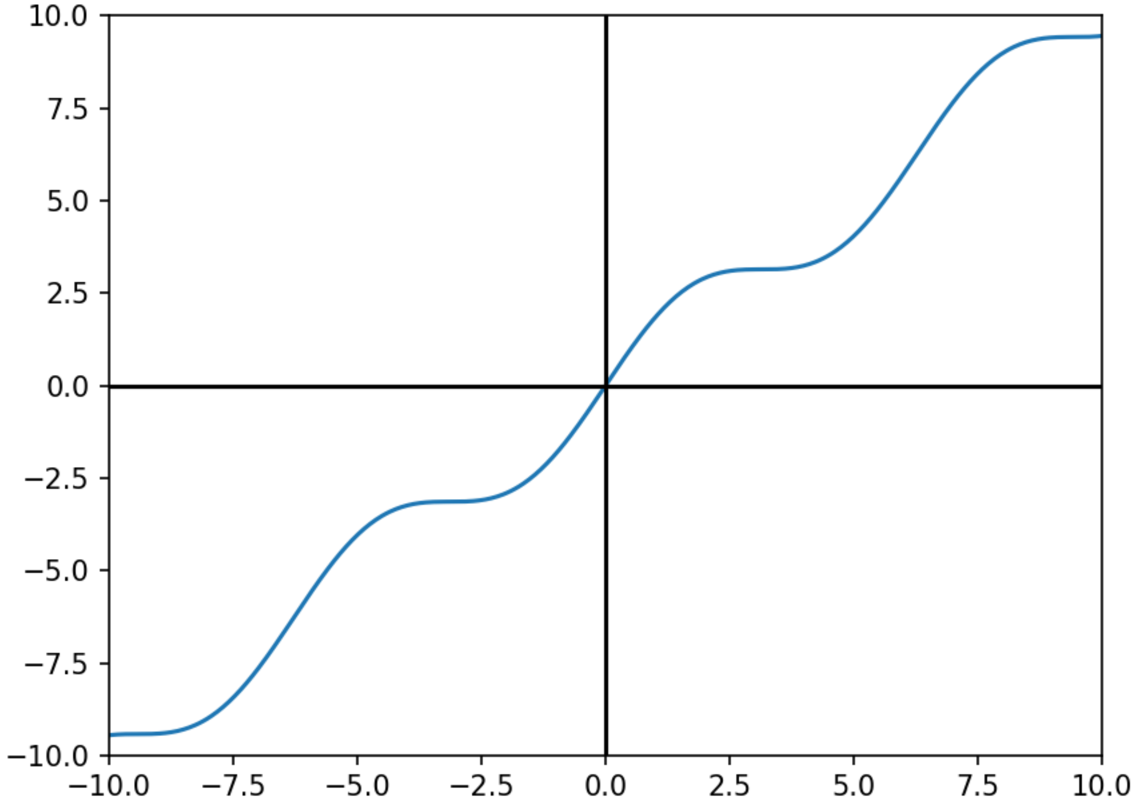
b. $y=\sin x+(1/10)\sin(10x)$

c. $y=x\sin x$

d. $y=e^{-x}\sin x$

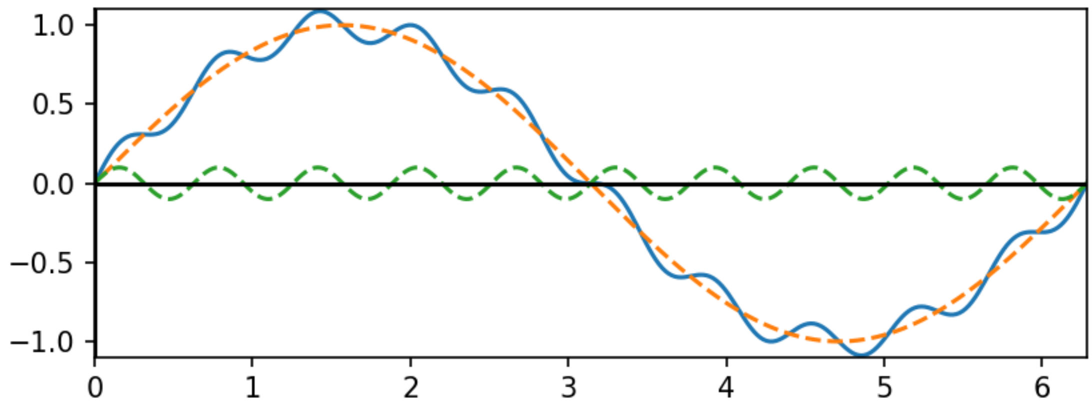
In [29]:

```
x=np.linspace(-20,20,1000)


plt.plot(x,np.sin(x)+x)
# plt.plot(x,x+1,'--',color='tab:blue')
# plt.plot(x,x-1,'--',color='tab:blue')
# plt.plot(x,np.sin(x),'--')

plt.axhline(y = 0, color = 'k', linestyle = '-')
plt.axvline(x = 0, color = 'k', linestyle = '-')
plt.xlim([-10,10])
plt.ylim([-10,10])
ax = plt.gca()
```



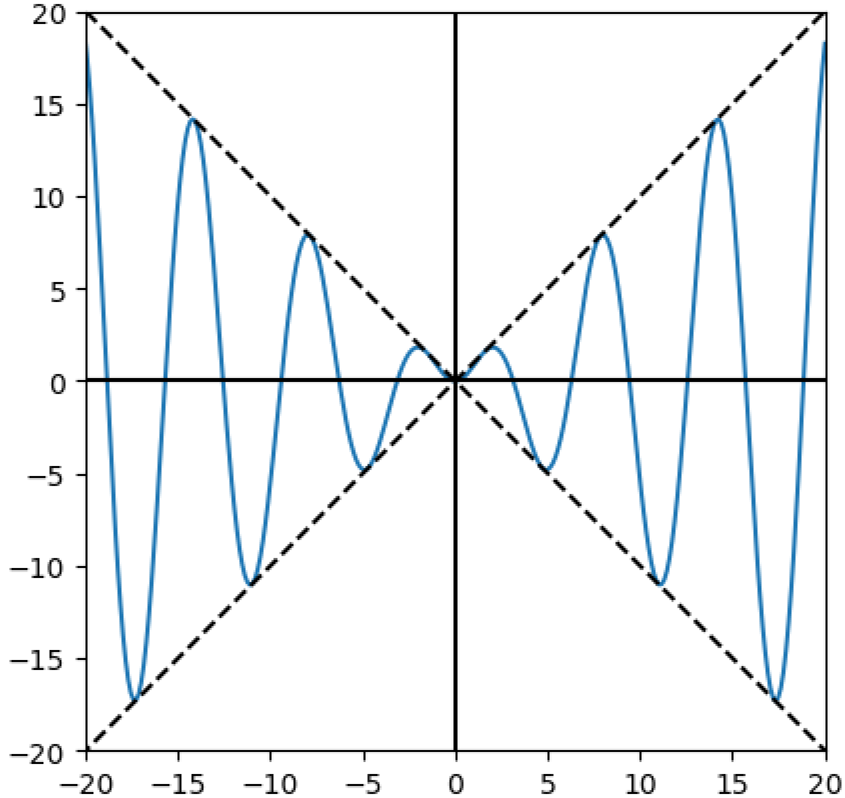
In [33]:

```
x=np.linspace(-10,10,1000)


plt.plot(x,np.sin(x)+np.sin(10*x)/10)
plt.plot(x,np.sin(x),'--')
plt.plot(x,np.sin(10*x)/10,'--')

plt.axhline(y = 0, color = 'k', linestyle = '-')
plt.axvline(x = 0, color = 'k', linestyle = '-')
plt.xlim([0,2*np.pi])
plt.ylim([-1.1,1.1])
ax = plt.gca()
ax.set_aspect('equal')
```

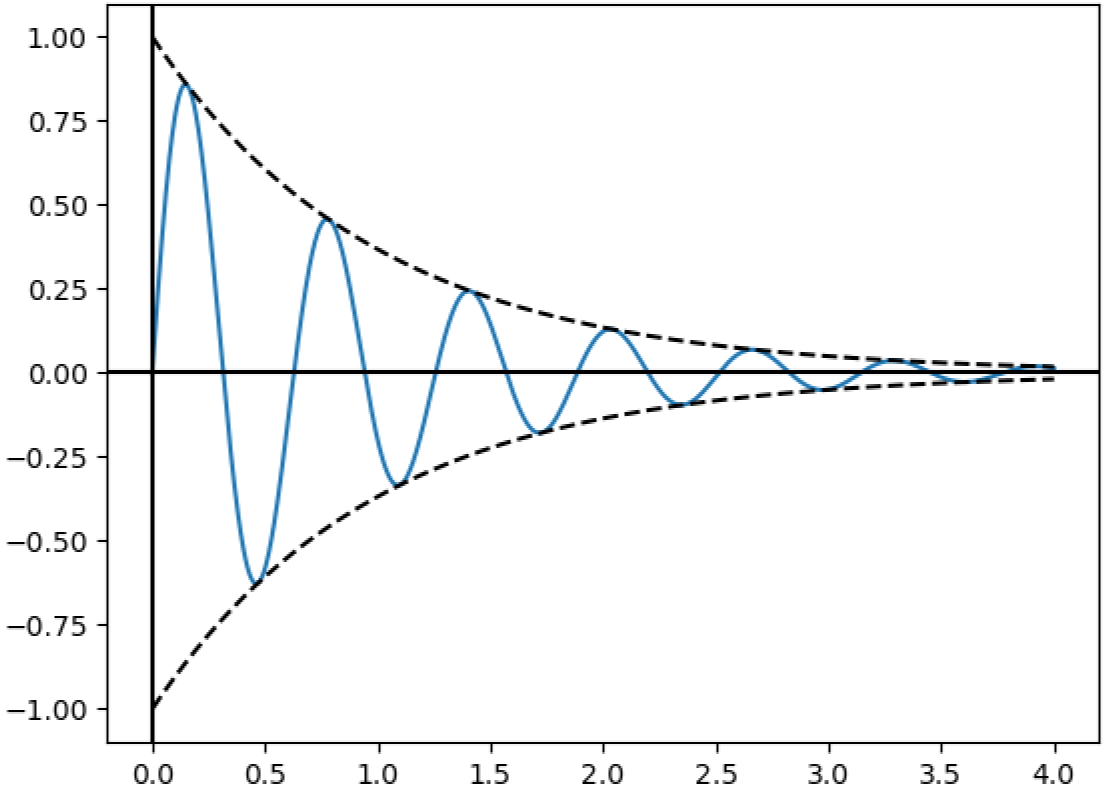


In [15]:

```
x=np.linspace(-120,20,1000)


plt.plot(x,np.sin(x)*x)
plt.plot(x,x,'--',color='k')
plt.plot(x,-x,'--',color='k')

plt.axhline(y = 0, color = 'k', linestyle = '-')
plt.axvline(x = 0, color = 'k', linestyle = '-')
plt.xlim([-20,20])
plt.ylim([-20,20])
ax = plt.gca()
ax.set_aspect('equal')
```



In [50]:
```
x=np.linspace(0,4,1000)


plt.plot(x,np.e**(-x)*np.sin(10*x))
plt.plot(x,np.e**(-x),'--',color='k')
plt.plot(x,-np.e**(-x),'--',color='k')

plt.axhline(y = 0, color = 'k', linestyle = '-')
plt.axvline(x = 0, color = 'k', linestyle = '-')
# plt.xlim([-20,120])
# plt.ylim([-20,120])
ax = plt.gca()
# ax.set_aspect('equal')
```


1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0