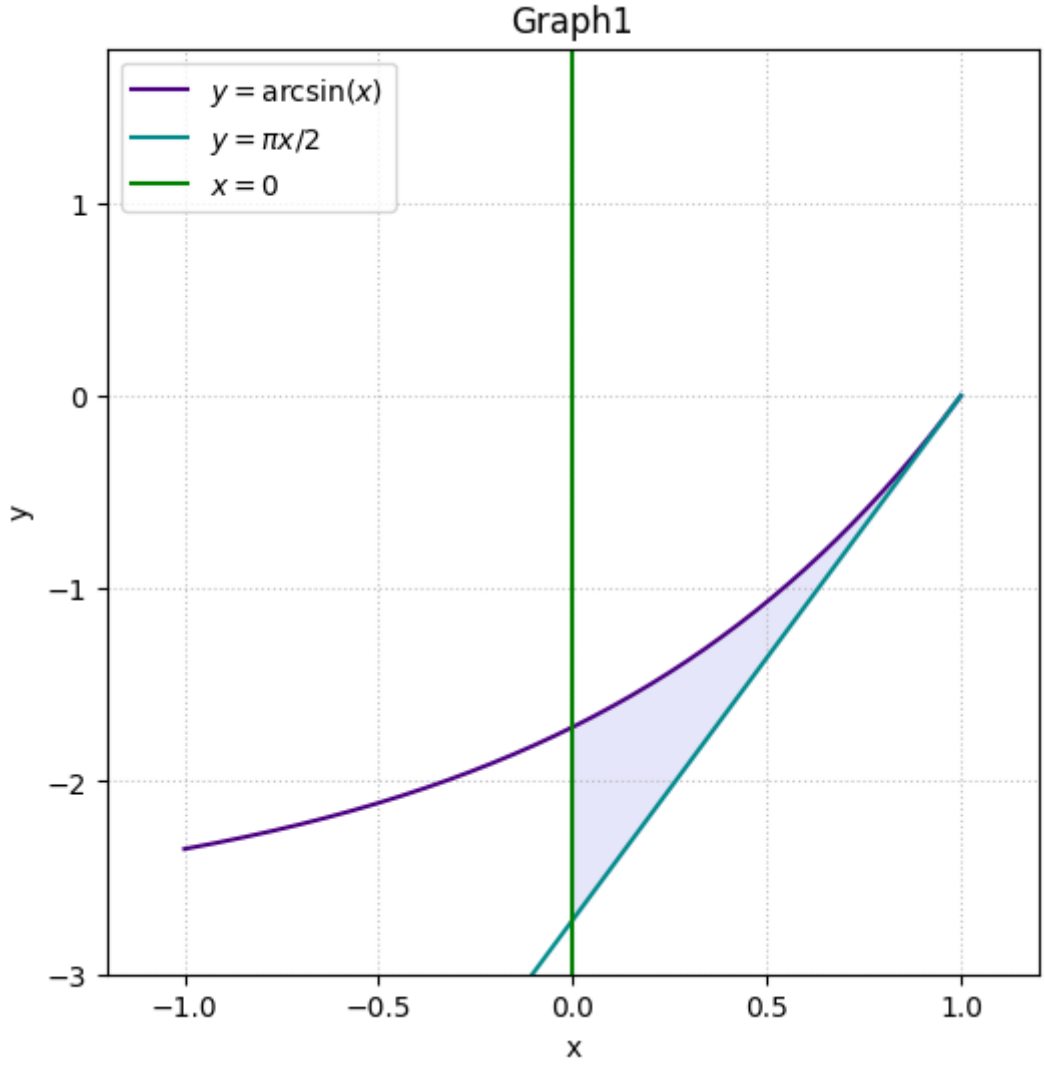
Graphics

На следующих страницах представлены 5 графиков функций в декартовых и полярных координатах, написанные на языке Python с использованием библиотек numpy и matplotlib.

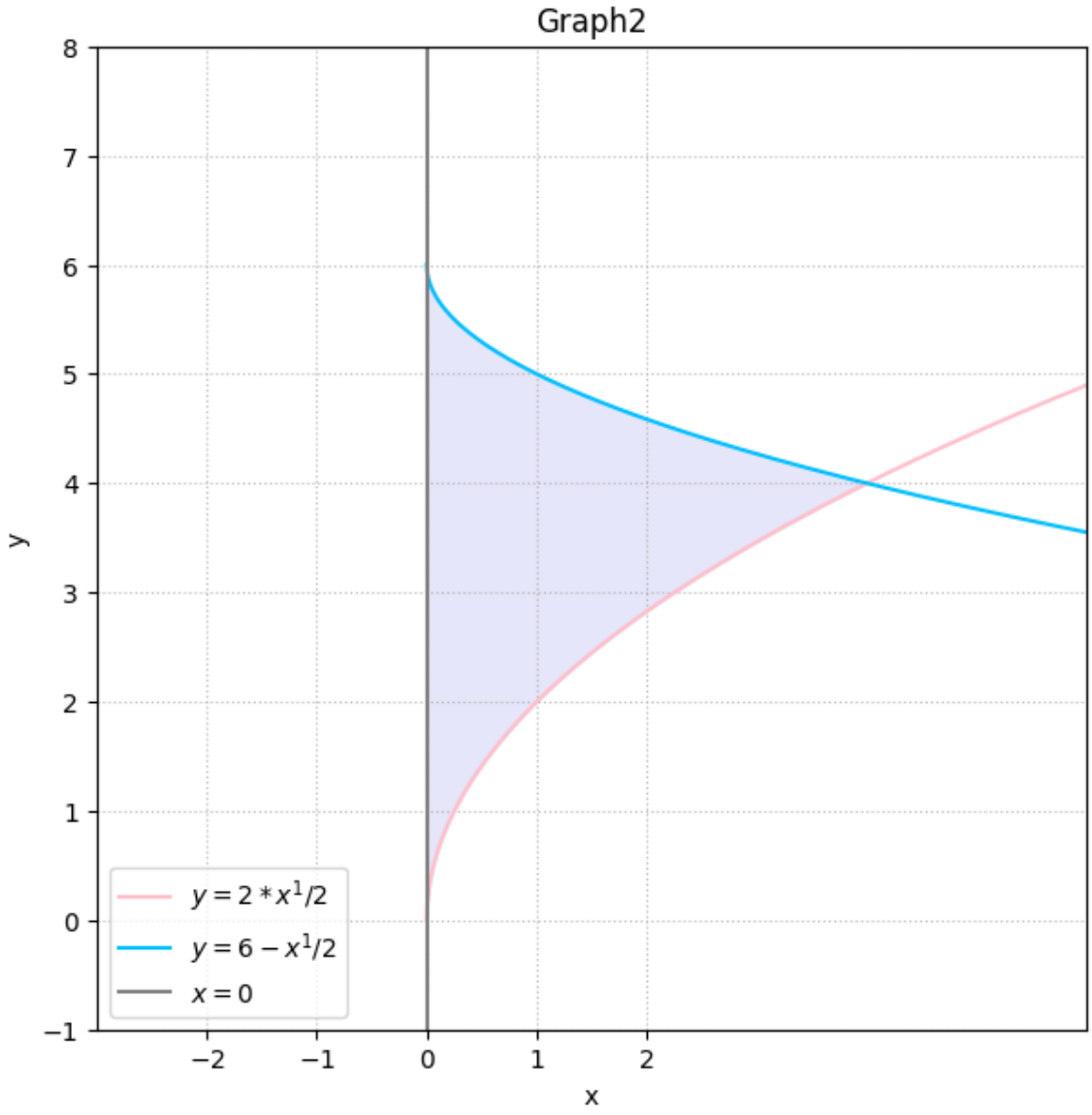
In [54]:

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

plt.figure (figsize=(6,6))

t = np.linspace(-1, 1, 1000)
y1 = np.exp(t) - np.exp(1)
y2 = np.exp(1)*t-np.exp(1)
t3 = np.linspace(0, 0, 10)
y3 = np.linspace(-5, 5, 10)

ax = plt.gca ()
ax.set_xlabel ('x')
ax.set_ylabel ('y')
plt.plot (t, y1, color ='indigo', label =r'$y =\mathrm{arcsin}(x)$')
plt.plot (t, y2, color ='darkcyan', label =r'$y =\pi x/2$')
plt.plot (t3, y3, color ='green', label =r'$x = 0$')
plt.fill_between (t, y1, y2,where=t>=0, color ='lavender')
plt.grid (alpha =0.75, linestyle =':')
plt.xlim (-1.2 ,1.2)
plt.ylim (-3 ,1.8)
plt.title ('Graph1')
plt.legend (loc=2)
# plt.savefig ('Graph1. eps', format ='eps')
plt.show ()
```



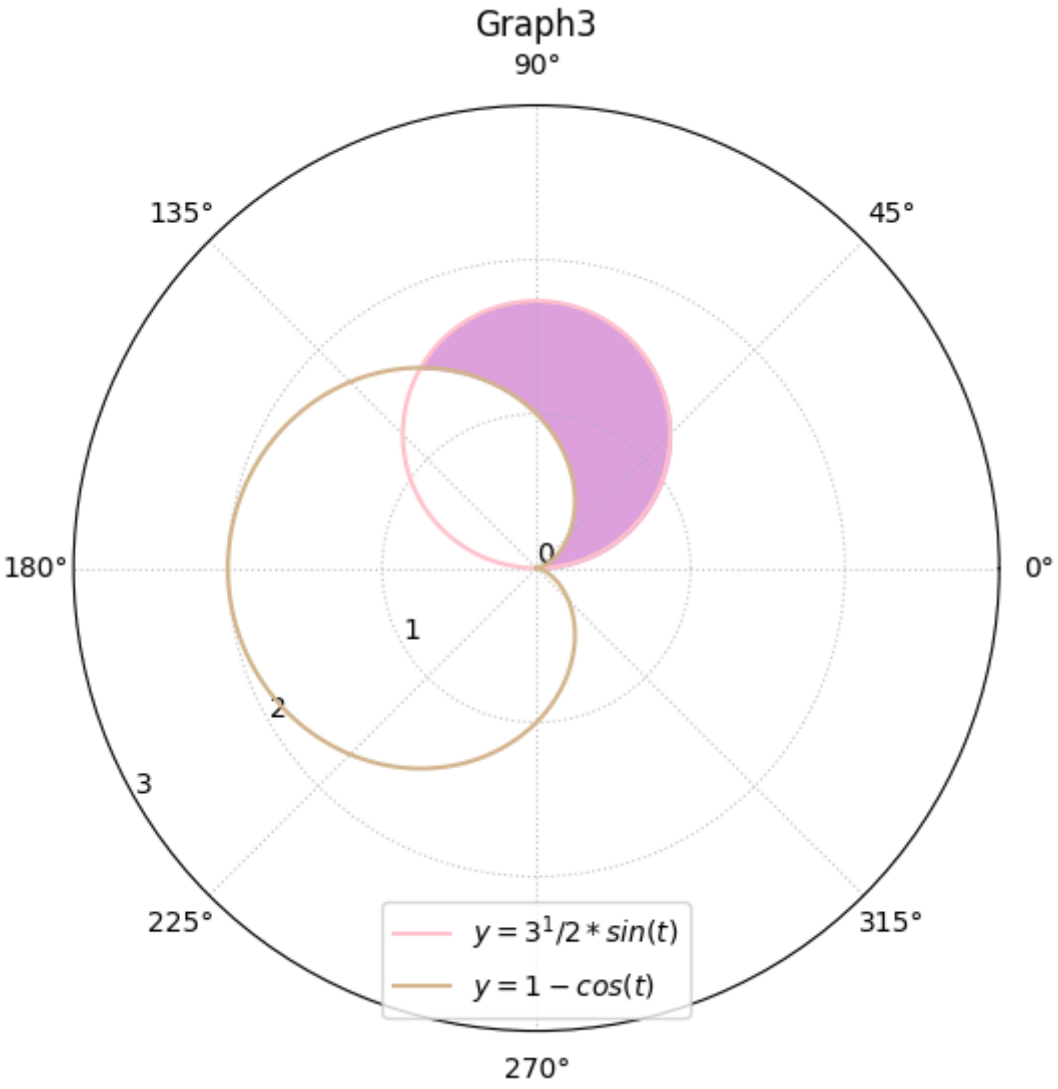
In [55]:

```
plt.figure (figsize=(7,7))

t = np.linspace(0, 10, 2000)

y1 = 2*t**0.5
y2 = 6-t**0.5
t3 = np.linspace(0, 0, 10)
y3 = np.linspace(-4, 8, 10)

ax = plt.gca()
ax.set_xlabel('x')
ax.set_ylabel('y')

plt.plot(t,y1, color = 'pink', label=r'$y=2*x^1/2$')
plt.plot(t,y2, color = 'deepskyblue', label=r'$y=6-x^1/2$')
plt.plot(t3,y3, color = 'gray', label=r'$x=0$')

plt.grid(alpha = 0.8, linestyle=':')
points = np.array([-2,-1,0,1,2])
labels = [r'$-2$',r'$-1$',r'$0$',r'$1$',r'$2$']
plt.xticks(points, labels)
plt.xlim(-3,6)
plt.ylim(-1,8)
plt.fill_between(t,y1,y2,where=(y1<y2), color = 'lavender')

plt.title('Graph2')
plt.legend(loc=3)
# plt.savefig('Graph2.eps', format='eps')

plt.show()
```


Graph2
8
7
6
5
4
3
2
1
0
−1
y
−2
−1
0
1
2
x
$y = 2 * x^1/2$
$y = 6 - x^1/2$
$x = 0$

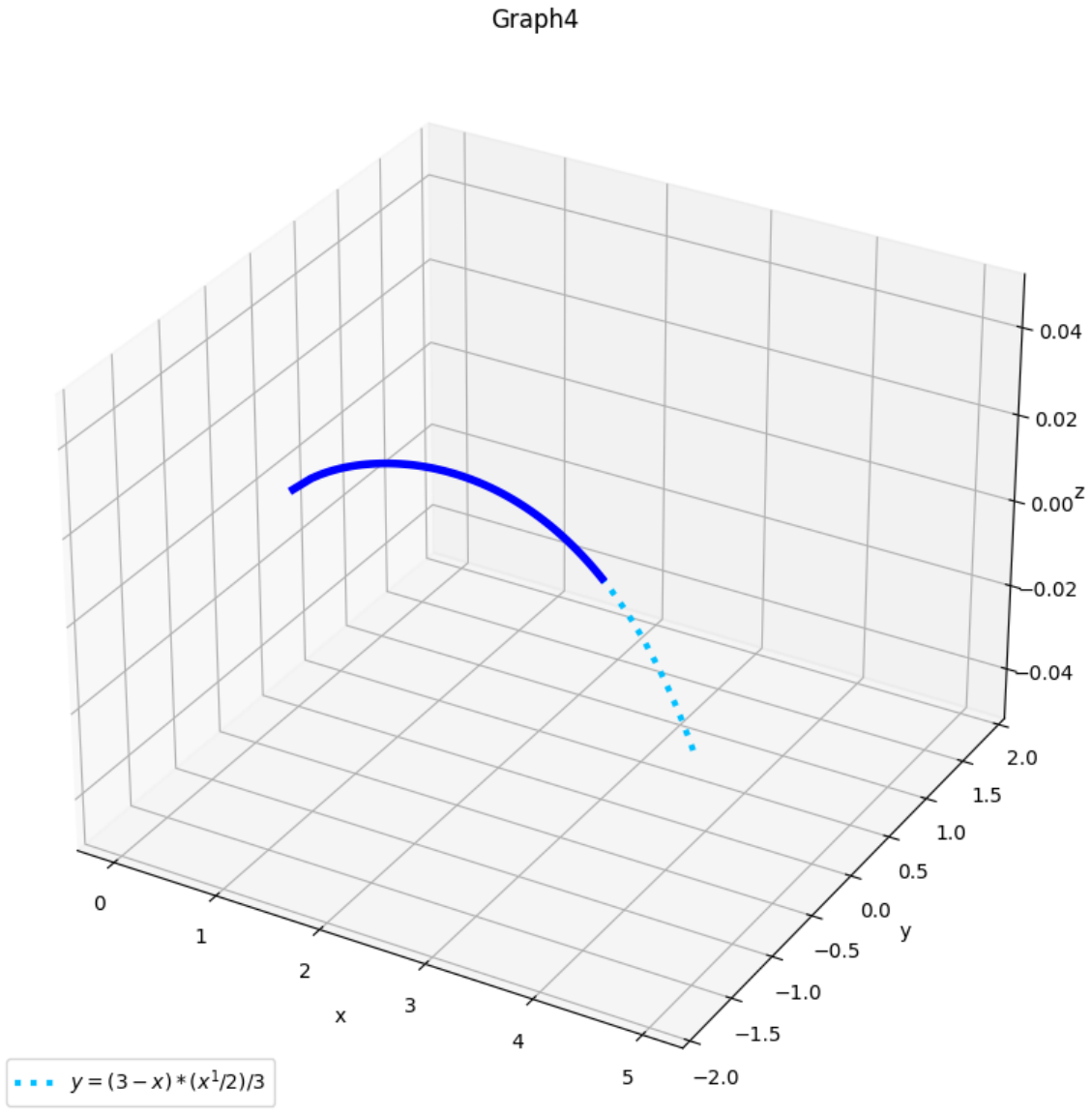
In [56]:

```
plt.figure (figsize=(6,6))

t = np.linspace(-10, 10, 2000)

t = np.linspace (0, 2*np.pi, 1000)
y1=3**0.5*np.sin(t)
y2=1-np.cos(t)

plt.subplot(projection='polar')
plt.ylim(bottom =0)

ax = plt.gca ()
ax.set_rgrids([i for i in range (4)])
ax.set_rlabel_position(210)

plt.polar (t, y1, color='pink', label =r'$y = 3^1/2*sin(t)$')
plt.polar (t, y2, color='tan', label =r'$ y = 1-cos(t)$')
plt.fill_between (t, y1, y2, where=( y1 > y2 ) , color ='plum')

plt.title ('Graph3')
plt.legend (loc=8)
plt.grid (alpha=0.75, linestyle =':')
# plt.savefig ('Graph3. eps', format ='eps')

plt.show()
```


Graph3
90°
135°
45°
180°
0°
225°
315°
270°
0
1
2
3
$y = 3^1/2 * sin(t)$
$y = 1 - cos(t)$

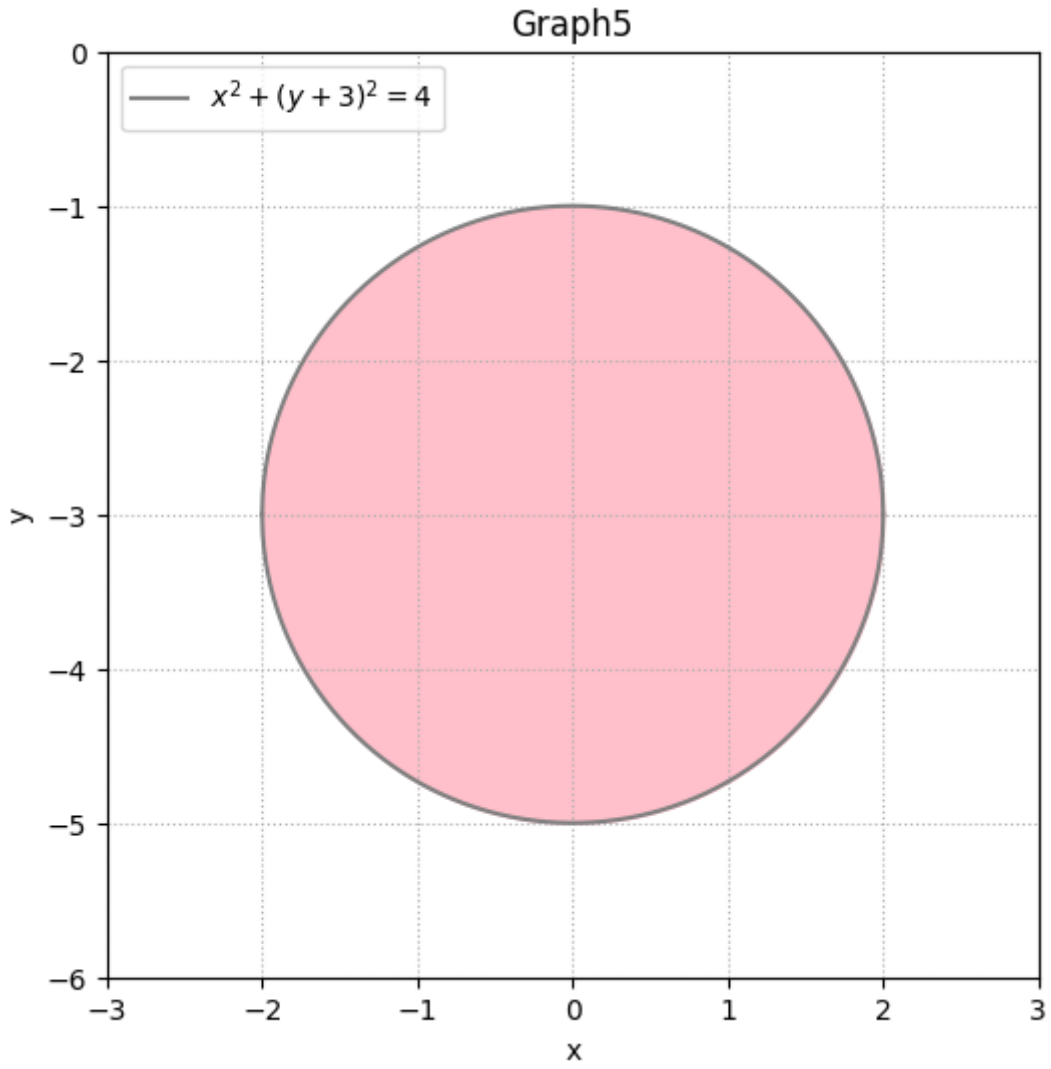
In [57]:

```
fig = plt.figure(figsize=(8,8))

def curve(t):
    x = t
    y = (3-t)*((t**0.5)/3)
    return (x,y)

t = np.linspace(0, 5, 1000)
t_0 = np.linspace(0, 3, 100)

x,y=curve(t)
x_0,y_0=curve(t_0)

ax = fig.add_subplot(111, projection='3d')

ax.plot(x,y,c='deepskyblue',ls=':',lw=3,label=r'$y=(3-x)*(x^1/2)/3$')
ax.plot(x_0,y_0,c='blue',lw=4)

plt.legend(loc=3)
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('z')
ax.zaxis.labelpad = -1

plt.ylim(-2, 2)

plt.title('Graph4')
plt.tight_layout()
# plt.savefig('Graph4.eps',format='eps')

plt.show()
```


Graph4
0.04
0.02
0.00
z
−0.02
−0.04
2.0
1.5
1.0
0.5
0.0
y
−0.5
−1.0
−1.5
−2.0
0
1
2
x
3
4
5
$y = (3 - x) * (x^1/2)/3$

In [58]:

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

plt.figure(figsize=(6,6))

def y1(x):
    return (4-x**2)**0.5-3

def y2(x):
    return -((4-x**2)**0.5)-3

x = np.linspace(-2,2,1000)

plt.plot(x, y1(x), c='gray',label=r'$x^2+(y+3)^2=4$')
plt.plot(x, y2(x), c='gray')
plt.fill_between(x,y1(x),y2(x),color='pink')

ax = plt.gca()
ax.set_xlabel('x')
ax.set_ylabel('y')

plt.title('Graph5')
plt.legend(loc=2)
plt.grid(linestyle=':')
plt.xlim(-3,3)
plt.ylim(-6,0)

# plt.savefig('Graph5.eps',format='eps')
plt.show()
```


Graph5
$x^2 + (y+3)^2 = 4$
0
−1
−2
−3
−4
−5
−6
y
−3
−2
−1
0
1
2
3
x