

プログラムリスト 4.1 三角関数 $\sin x$ と $\sin^2 x$ をプロットするPythonプログラム

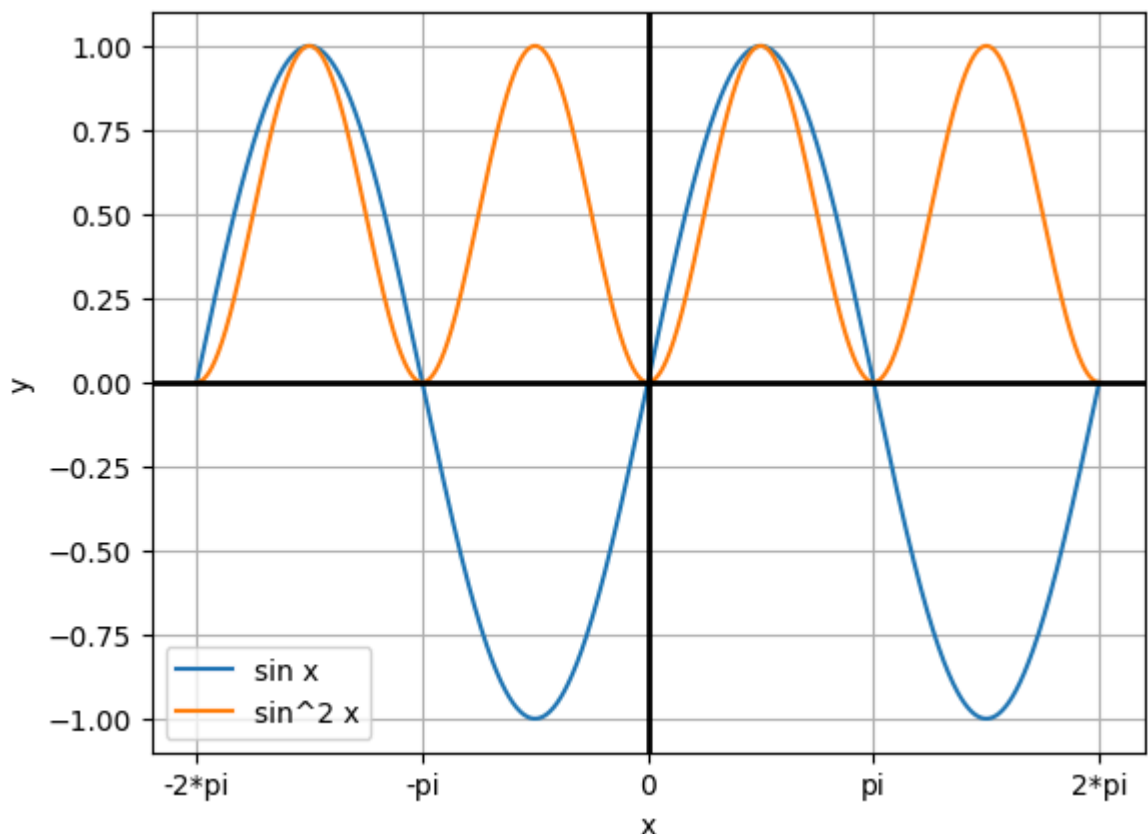
In [16]:

```
import numpy as np          # (A1) numpyライブラリをimport
import matplotlib.pyplot as plt # (A2) matplotlibライブラリをimport

x = np.linspace(-2*np.pi, 2*np.pi, 1000) # (B1) xの範囲を  $-2\pi$  から  $2\pi$  に設定
y1 = np.sin(x)              # (C1)  $y1 = \sin x$ 
y2 = np.sin(x)**2           # (C2)  $y2 = \sin^2 x$ 
plt.plot(x, y1, label="sin x") # (E1)  $\sin x$  のグラフを描画
plt.plot(x, y2, label="sin^2 x") # (E2)  $\sin^2 x$  を描画

plt.grid(True)              # (F1) グリッド線を表示
plt.xlabel("x")              # (F2) x軸ラベルの設定
plt.ylabel("y")              # (F3) y軸ラベルの設定
plt.axvline(x=0, color='black', linewidth=2) # (F4) y軸の線を太く表示
plt.axhline(y=0, color='black', linewidth=2) # (F5) x軸の線を太く表示
plt.legend()                 # (F6) 凡例を表示

plt.ylim(-1.1, 1.1)         # (F8) y軸の表示範囲
ticks=np.arange(-2*np.pi, 2.5*np.pi, np.pi) # (F9) x軸目盛の範囲
labels=["-2*pi", "-pi", "0", "pi", "2*pi"] # (F10) x軸目盛に $\pi(\pi)$ 表記を使用
plt.xticks(ticks, labels)    # (F11) x軸を $\pi$ で表示
plt.show()                  # (G) グラフを表示
```



プログラムリスト 4.2 三角関数 $\sin^2 x + \cos^2 x$ をプロットするPythonプログラム

In [17]:

```
import numpy as np          # (A1) numpyライブラリをimport
import matplotlib.pyplot as plt # (A2) matplotlibライブラリをimport

x = np.linspace(-2*np.pi, 2*np.pi, 1000) # (B1) xの範囲を  $-2\pi$  から  $2\pi$  に設定

#変更部分始まり
y1 = np.sin(x)**2           # (C1)  $y1 = \sin^2 x$ 
y2 = np.cos(x)**2           # (C2)  $y2 = \cos^2 x$ 
```

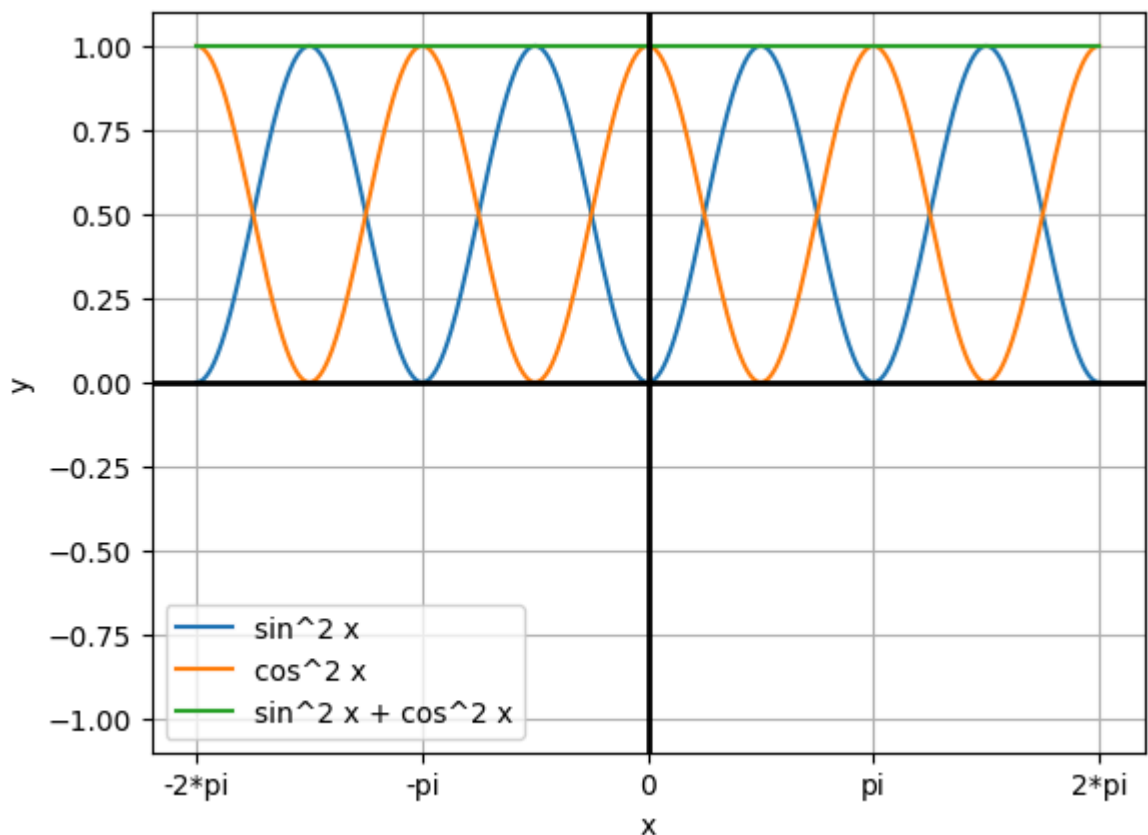
```

y_sum = y1 + y2                                # (C3) y_sum にy1とy2の和を代入する
plt.plot(x, y1, label="sin^2 x")                # (E1) sin^2 x を描画
plt.plot(x, y2, label="cos^2 x")                # (E2) cos^2 x を描画
plt.plot(x, y_sum, label="sin^2 x + cos^2 x")    # (E3) sin^2 x + cos^2 x を描画
#変更部分終わり

plt.grid(True)                                  # (F1) グリッド線を表示
plt.xlabel("x")                                  # (F2) x軸ラベルの設定
plt.ylabel("y")                                  # (F3) y軸ラベルの設定
plt.axvline(x=0, color='black', linewidth=2)    # (F4) y軸の線を太く表示
plt.axhline(y=0, color='black', linewidth=2)    # (F5) x軸の線を太く表示
plt.legend()                                     # (F6) 凡例を表示

plt.ylim(-1.1, 1.1)                             # (F8) y軸の範囲
ticks=np.arange(-2*np.pi, 2.5*np.pi, np.pi) # (F9) x軸目盛の範囲
labels=["-2*pi", "-pi", "0", "pi", "2*pi"]      # (F10) x軸目盛にpi(π)表記を使用
plt.xticks(ticks, labels)                       # (F11) x軸をπで表示
plt.show()                                       # (G) グラフを表示

```



プログラムリスト 4.3 逆三角関数 $\arcsin x$ と $\arccos x$ をプロットするPythonプログラム

```

In [18]: import numpy as np                    # (A1) numpyライブラリをimport
import matplotlib.pyplot as plt                # (A2) matplotlibライブラリをimport

x = np.linspace(-1, 1, 200)                  # (B1) xの範囲を -1 から 1 に設定
y1 = np.arcsin(x)                            # (C1) y1 = arcsin x
y2 = np.arccos(x)                            # (C2) y2 = arccos x
plt.plot(x, y1, label="arcsin x")             # (E1) arcsin x を描画
plt.plot(x, y2, label="arccos x")             # (E2) arccos x を描画

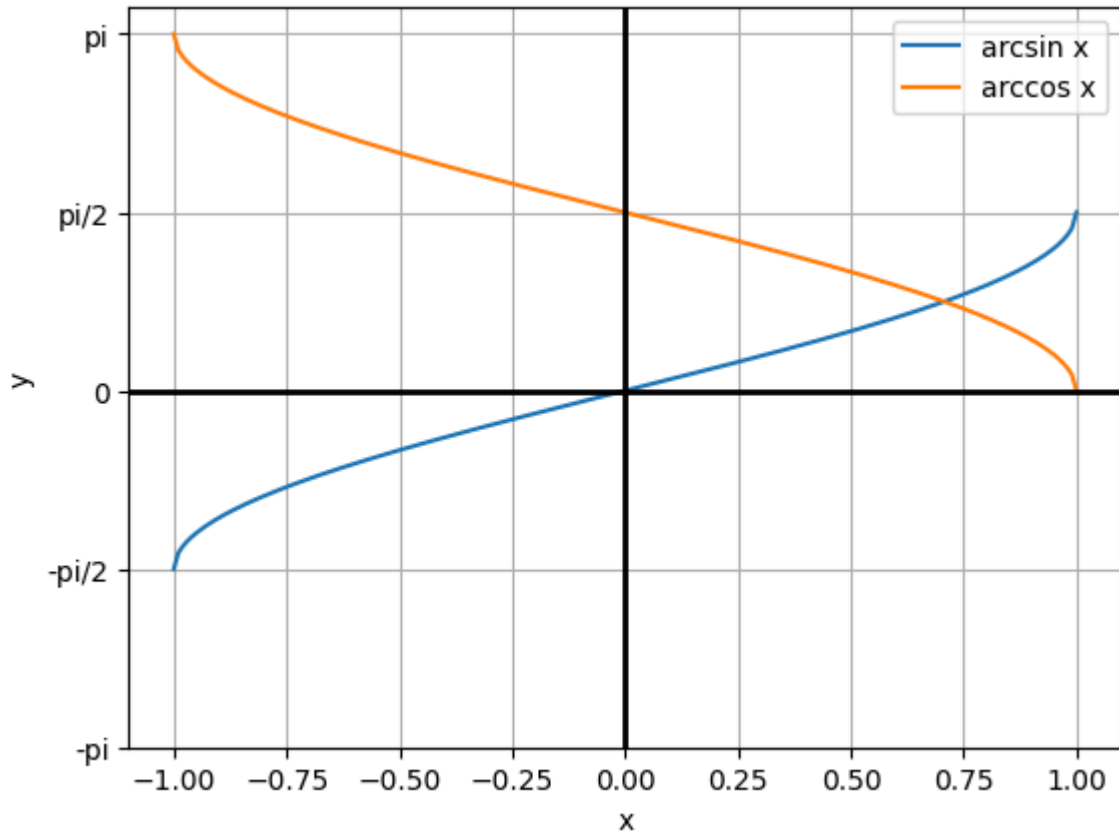
plt.grid(True)                                # (F1) グリッド線を表示
plt.xlabel("x")                                # (F2) x軸ラベルの設定
plt.ylabel("y")                                # (F3) y軸ラベルの設定
plt.axvline(x=0, color='black', linewidth=2)  # (F4) y軸の線を太く表示
plt.axhline(y=0, color='black', linewidth=2)  # (F5) x軸の線を太く表示
plt.legend()                                   # (F6) 凡例を表示

```

```

ticks = np.arange(-np.pi, 1.1*np.pi, np.pi/2) # (F9) y軸目盛の範囲
labels = ["-pi", "-pi/2", "0", "pi/2", "pi"] # (F10) y軸目盛にpi表記を使用
plt.yticks(ticks, labels) # (F11) y軸を $\pi$ で表示
plt.show() # (G) グラフを表示

```



プログラムリスト 4.4 逆三角関数 $\arctan x$ をプロットするPythonプログラム

```

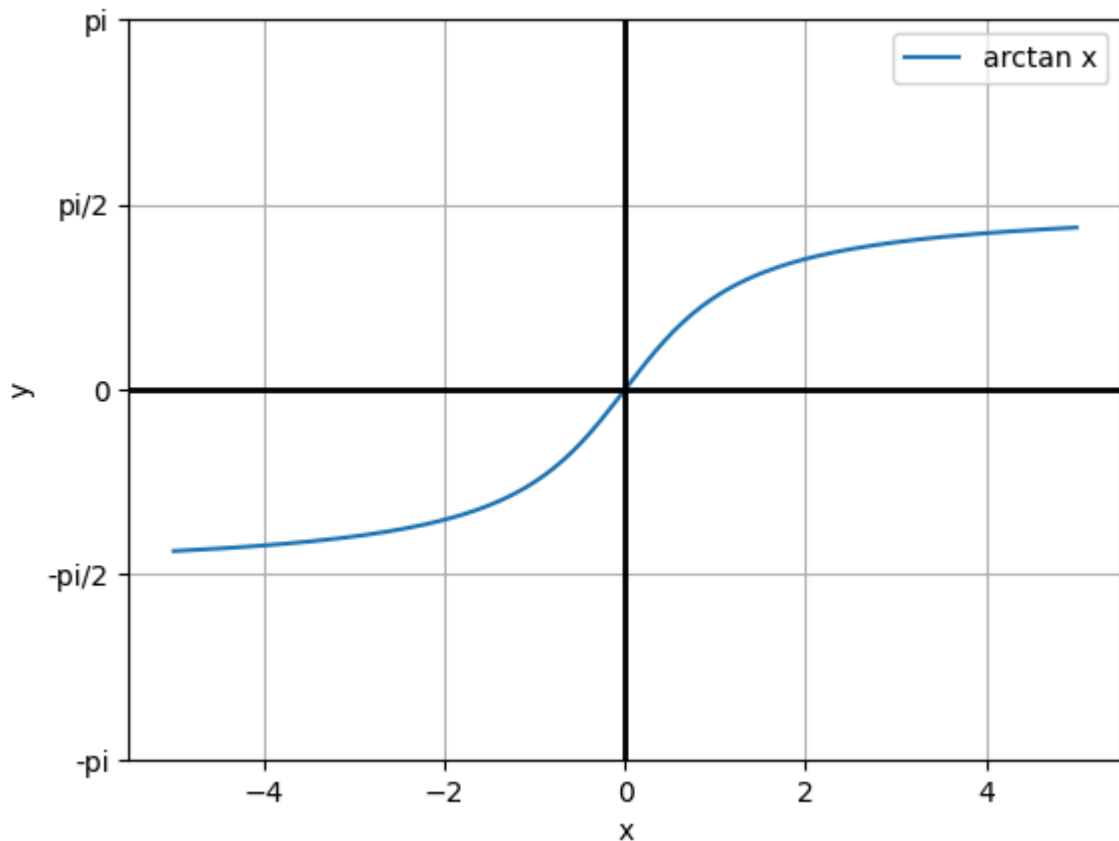
In [19]: import numpy as np # (A1) numpyライブラリをimport
import matplotlib.pyplot as plt # (A2) matplotlibライブラリをimport

#変更部分始まり
x = np.linspace(-5, 5, 200) # (B1) xの範囲を -5 から 5 に設定
y1 = np.arctan(x) # (C1) y1 = arctan x
plt.plot(x, y1, label="arctan x") # (E1) arctan x を描画
#変更部分終わり

plt.grid(True) # (F1) グリッド線を表示
plt.xlabel("x") # (F2) x軸ラベルの設定
plt.ylabel("y") # (F3) y軸ラベルの設定
plt.axvline(x=0, color='black', linewidth=2) # (F4) y軸の線を太く表示
plt.axhline(y=0, color='black', linewidth=2) # (F5) x軸の線を太く表示
plt.legend() # (F6) 凡例を表示

ticks = np.arange(-np.pi, 1.1*np.pi, np.pi/2) # (F9) y軸目盛の範囲
labels = ["-pi", "-pi/2", "0", "pi/2", "pi"] # (F10) y軸目盛にpi表記を使用
plt.yticks(ticks, labels) # (F11) y軸を $\pi$ で表示
plt.show() # (G) グラフを表示

```



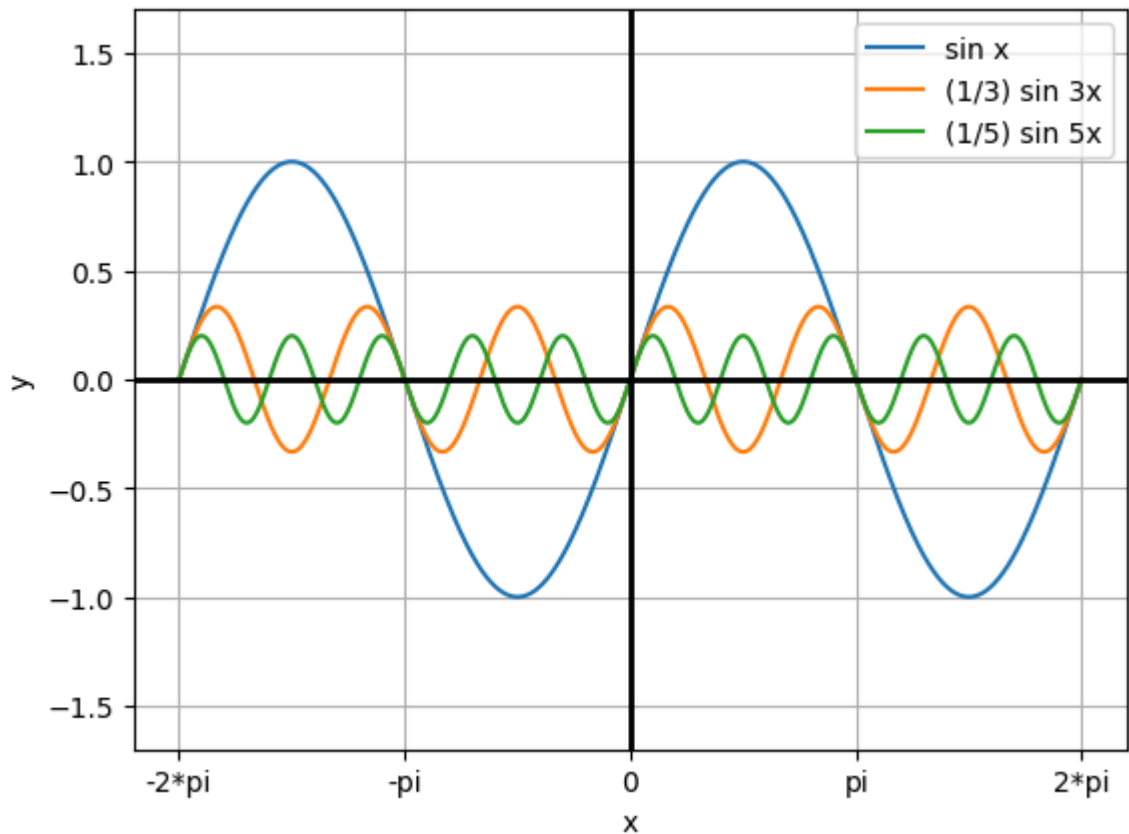
プログラムリスト 4.5 振幅・周波数を変えた三角関数

```
In [21]: import numpy as np                # (A1) numpyライブラリをimport
import matplotlib.pyplot as plt          # (A2) matplotlibライブラリをimport
x = np.linspace(-2*np.pi, 2*np.pi, 1000) # (B1) xの範囲を  $-2\pi$  から  $2\pi$  に設定

y1 = np.sin(x)                          # (C1)  $y1 = \sin x$ 
y2 = (1/3) * np.sin(3*x)                 # (C2)  $y2 = (1/3) \sin 3x$ 
y3 = (1/5) * np.sin(5*x)                 # (C3)  $y3 = (1/5) \sin 5x$ 
plt.plot(x, y1, label="sin x")            # (E1)  $\sin x$  を描画
plt.plot(x, y2, label="(1/3) sin 3x")    # (E2)  $(1/3) \sin 3x$  を描画
plt.plot(x, y3, label="(1/5) sin 5x")    # (E3)  $(1/5) \sin 5x$  を描画

plt.grid(True)                          # (F1) グリッド線を表示
plt.xlabel("x")                          # (F2) x軸ラベルの設定
plt.ylabel("y")                          # (F3) y軸ラベルの設定
plt.axvline(x=0, color='black', linewidth=2) # (F4) y軸の線を太く表示
plt.axhline(y=0, color='black', linewidth=2) # (F5) x軸の線を太く表示
plt.legend()                             # (F6) 凡例を表示

plt.ylim(-1.7, 1.7)                     # (F8) y軸の範囲
ticks=np.arange(-2*np.pi, 2.5*np.pi, np.pi) # (F9) x軸目盛の範囲
labels=["-2*pi", "-pi", "0", "pi", "2*pi"] # (F10) x軸目盛に  $\pi(\pi)$  表記を使用
plt.xticks(ticks, labels)                # (F11) x軸を  $\pi$  で表示
plt.show()                               # (G) グラフを表示
```



プログラム 4.6 三角関数の合成による方形波近似

```
In [22]: import numpy as np                # (A1) numpyライブラリをimport
import matplotlib.pyplot as plt        # (A2) matplotlibライブラリをimport
x = np.linspace(-2*np.pi, 2*np.pi, 1000) # (B1) xの範囲を  $-2\pi$  から  $2\pi$  に設定

#変更部分始まり
y1 = np.sin(x)                        # (C1)  $y1 = \sin x$ 
y2 = (1/3) * np.sin(3*x)              # (C2)  $y2 = (1/3) \sin 3x$ 
y3 = (1/5) * np.sin(5*x)              # (C3)  $y3 = (1/5) \sin 5x$ 
y_sum = 4/np.pi * (y1 + y2 + y3)      # (C4)  $y\_sum$  に3つの波の和を代入
y_square = np.sign(np.sin(x))          # (C5) 方形波を生成 ( $\sin$ 波の符号)
plt.plot(x, y_sum, label="combined wave") # (E4) 合成波を描画
plt.plot(x, y_square, label="square wave") # (E5) 方形波を破線で描画
#変更部分終わり

plt.grid(True)                        # (F1) グリッド線を表示
plt.xlabel("x")                       # (F2) x軸ラベルの設定
plt.ylabel("y")                       # (F3) y軸ラベルの設定
plt.axvline(x=0, color='black', linewidth=2) # (F4) y軸の線を太く表示
plt.axhline(y=0, color='black', linewidth=2) # (F5) x軸の線を太く表示
plt.legend()                          # (F6) 凡例を表示

plt.ylim(-1.7, 1.7)                  # (F8) y軸の範囲
ticks=np.arange(-2*np.pi, 2.5*np.pi, np.pi) # (F9) x軸目盛の範囲
labels=["-2*pi", "-pi", "0", "pi", "2*pi"] # (F10) x軸目盛に  $\pi$  表記を使用
plt.xticks(ticks, labels)             # (F11) x軸を  $\pi$  で表示
plt.show()                           # (G) グラフを表示
```

