matplotlib で 3 次元図の作成

sugayu

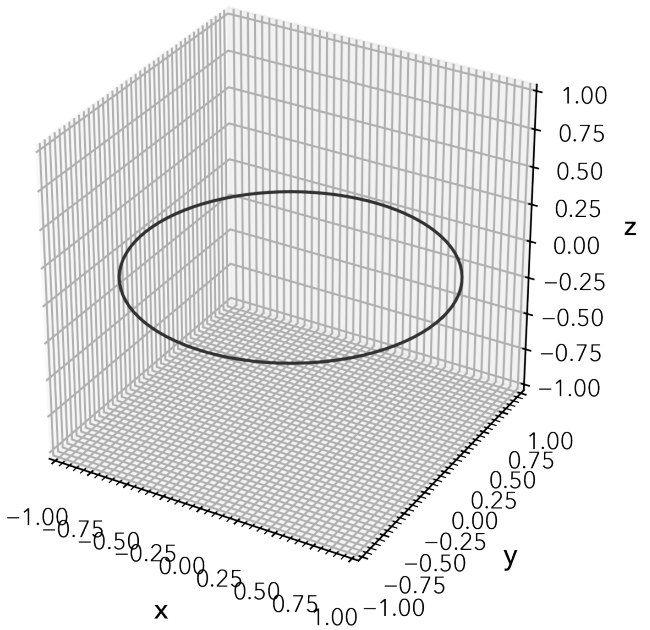
2025 年 1 月 1 日

目次

1 基準の図

`fig.add_subplot()` に `projection='3d'` を渡して 3 次元図を作成する。このとき返ってくるのは 3 次元専用の軸オブジェクト `Axes3D` である。

```
from mpl_toolkits.mplot3d.axes3d import Axes3D
from sugayutils import colors

theta = np.pi / 2.0
phi = np.linspace(0.0, 2.0 * np.pi, 91)


def xyz(theta, phi):
    x = np.sin(theta) * np.cos(phi)
    y = np.sin(theta) * np.sin(phi)
    z = np.cos(theta) * np.ones_like(phi)
    return x, y, z


x, y, z = xyz(theta, phi)


def plot_fiducial(left=-0.1) -> Axes3D:
    fig = plt.figure(figsize=(3.5, 3.5))
    fig.subplots_adjust(left, 0.1, 0.98, 0.99,
    ↪  0.0, 0.0)
    ax = fig.add_subplot(1, 1, 1,
    ↪  projection='3d')
    ax.plot(x, y, z, c=colors.black)
    ax.set_xlim3d(-1.0, 1.0)
    ax.set_ylim3d(-1.0, 1.0)
    ax.set_zlim3d(-1.0, 1.0)
    ax.set_xlabel('x')
    ax.set_ylabel('y')
    ax.set_zlabel('z')
    ax.set_aspect('equal')
    return fig, ax


def savefig(fig, fsave) -> str:
    fig.savefig(fsave)
    fig.clear()
    plt.close()
    return f'[[file:{fsave}]]'


fig, _ = plot_fiducial()
savefig(fig,
↪  './obipy-resources/fiducial_3d.png')
```



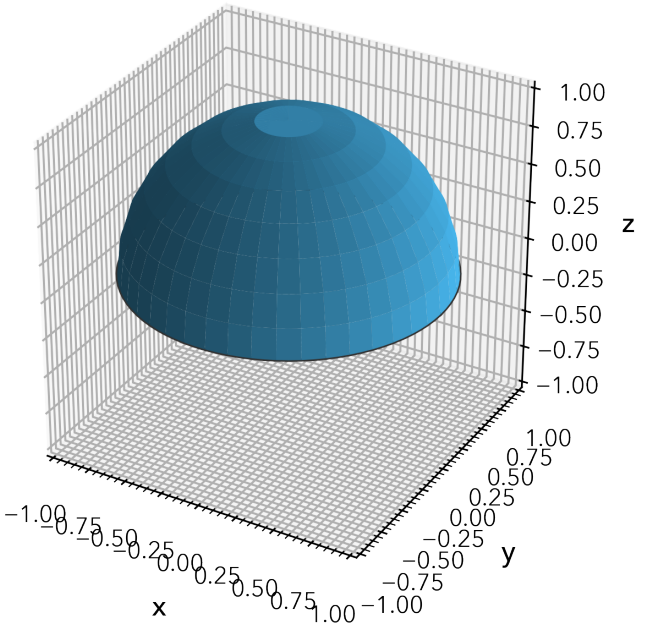
なお、おそらく ~Jupyter notebook~ の仕様で図の余白が自動的に調整されてしまうので、

ここでは ~ob-ipython~ のファイル保存形式ではなく、保存方法と表示方法を指定している。
ただし出力のたび、いちいち手作業が必要になる。
また、この場合の方がプロットがとても速い。

2 基本

2.1 面をプロット

```
fig, ax = plot_fiducial()
theta = np.linspace(0.0, np.pi / 2.0, 9).reshape(-1, 1)
_xyz = xyz(theta, phi)
ax.plot_surface(
    *_xyz,
    color=colors.sky,
    # facecolors=facecolor,
    alpha=1.0,
    # shade=True,
    # lightsource=light,
)
savefig(fig, './obipy-resources/surface_3d.png')
```



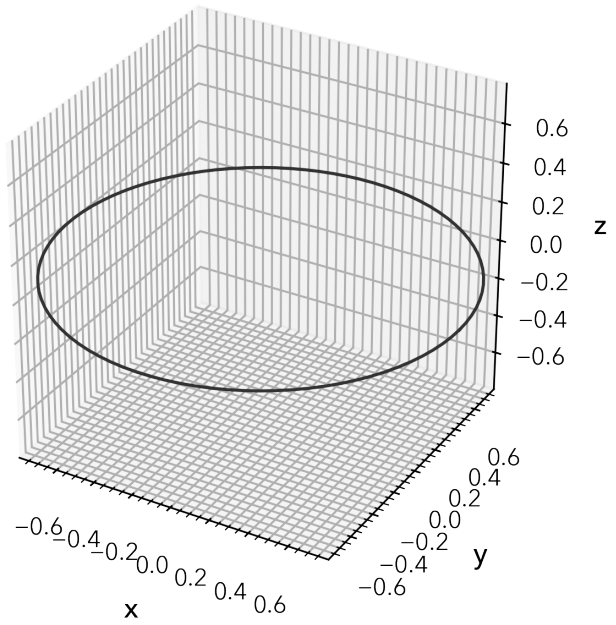
光が当たる角度は引数の `lightsource` に `matplotlib.colors.LightSource` を入れて指定できる。詳しくは

3 軸の設定

3.1 ズーム

ズームをする関数は用意されていない。軸の表示範囲を調整して似た機能を実現する。

```
fig, ax = plot_fiducial()
zoom = 1.3
ax.set_xlim3d(np.array(ax.get_xlim3d()) / zoom)
ax.set_ylim3d(np.array(ax.get_ylim3d()) / zoom)
ax.set_zlim3d(np.array(ax.get_zlim3d()) / zoom)
savefig(fig, './obipy-resources/zoom_3d.png')
```



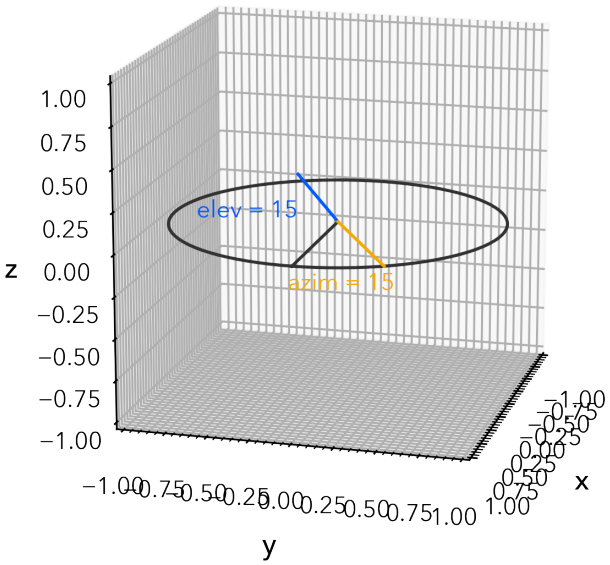
参照：[python - How to Zoom with Axes3D in Matplotlib - Stack Overflow

3.2 視点の角度

`Axes3D.view_init()` で変更する。パラメータの角度方向は直感の通りで、`azim` は x 軸正の向き (`y = 0` の方向) から反時計まわりの方位角、`elev` は `z = 0` の方向からの z 軸正の向きに上がる仰角。全ての角度は単位は度で入力する。

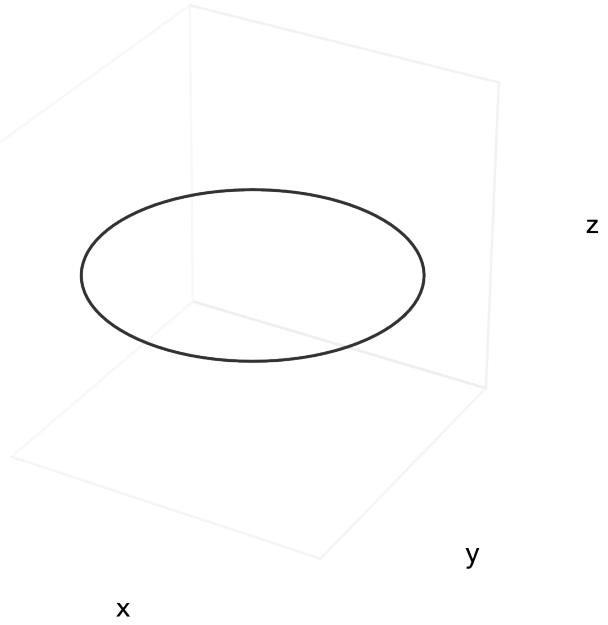
```
fig, ax = plot_fiducial(left=0.1)

ax.view_init(elev=15.0, azim=15.0)
# ax.view_init(elev=30.0, azim=-60.0)  # default

ax.plot(
    [1.0, 0.0],
    [0.0, 0.0],
    [0.0, 0.0],
    c=colors.black,
)
ax.plot(
    [0.0, np.cos(np.pi / 6)],
    [0.0, 0.0],
    [0.0, np.sin(np.pi / 6)],
    c=colors.blue,
)
ax.plot(
    [0.0, np.cos(np.pi / 6)],
    [0.0, np.sin(np.pi / 6)],
```

```
    [0.0, 0.0],
    c=colors.orange,
)
ax.text(
    np.cos(np.pi / 6),
    0.0,
    np.sin(np.pi / 6) / 2.0,
    'elev = 15',
    ha='right',
    color=colors.blue,
)
ax.text(
    np.cos(np.pi / 6),
    np.sin(np.pi / 6) / 2.0,
    -0.05,
    'azim = 15',
    va='top',
    ha='center',
    color=colors.orange,
)

savefig(fig, './obipy-resources/view_3d.png')
```



3.3 軸を消す

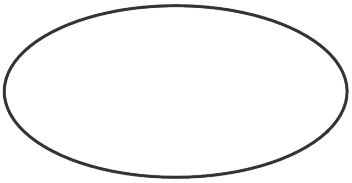
```
fig, ax = plot_fiducial()
ax.grid(False)  # grid を消す
ax.xaxis.pane.fill = False  # 壁を白くする
ax.yaxis.pane.fill = False
ax.zaxis.pane.fill = False
ax.set_xticks([])  # メモリを消す
ax.set_yticks([])
ax.set_zticks([])
ax.xaxis.line.set_color((1.0, 1.0, 1.0, 0.0))  # 軸を消す
ax.yaxis.line.set_color((1.0, 1.0, 1.0, 0.0))
ax.zaxis.line.set_color((1.0, 1.0, 1.0, 0.0))
ax.tick_params(  # ラベルを消す？ 消せない
    which='both',
    labelcolor='none',
    top=False,
    bottom=False,
    left=False,
    right=False,
)
savefig(fig, './obipy-resources/axis_3d.png')
```

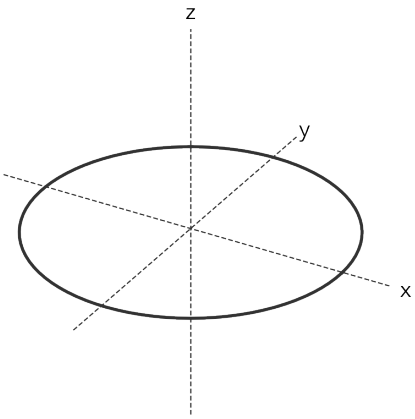


参照: python - Remove border from matplotlib 3D pane - Stack Overflow

3.4 軸を完全に消す

```
fig, ax = plot_fiducial()
ax.axis('off')
savefig(fig, './obipy-resources/noaxis_3d.png')
```





4　プロットの工夫

4.1　大量の線

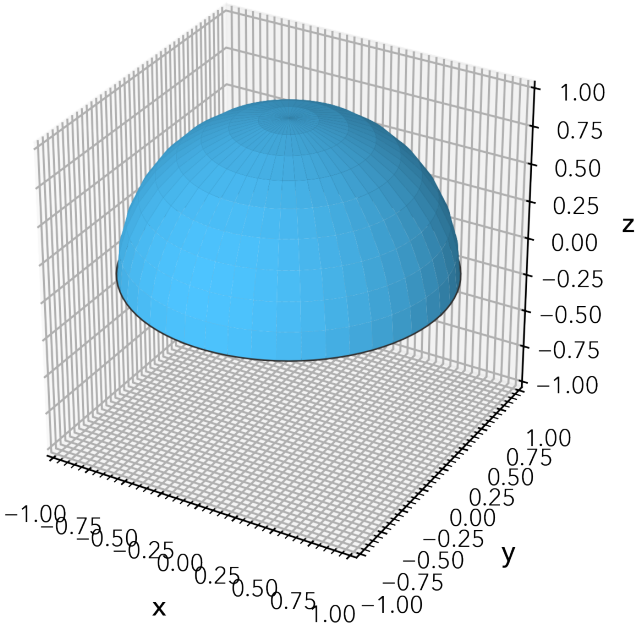
一斉に同じ種類の線をプロットするには `art3d.Line3DCollection` を使って、返り値を `ax.add_collection()` で加えると良い。

```
from mpl_toolkits.mplot3d import art3d

fig, ax = plot_fiducial()
ax.axis('off')

lim = 1.3
segments = (
    ((-lim, 0.0, 0.0), (lim, 0.0, 0.0)),
    ((0.0, -lim, 0.0), (0.0, lim, 0.0)),
    ((0.0, 0.0, -lim), (0.0, 0.0, lim)),
)
linecollection = art3d.Line3DCollection(segments, colors=colors.black, lw=0.5, ls='--')
ax.add_collection(linecollection)
ax.text(lim + 0.1, 0.0, 0.0, 'x', ha='center', va='center')
ax.text(0.0, lim + 0.1, 0.0, 'y', ha='center', va='center')
ax.text(0.0, 0.0, lim + 0.1, 'z', ha='center', va='center')
savefig(fig, './obipy-resources/lines_3d.png')
```

4.2　光の角度

光の角度は `matplotlib.colors.LightSource` で指定できる。パラメータの角度方向は直感に反していて、`azdeg` は y 軸負の向き (`x = 0` の方向) から時計まわりの方位角、`altdeg` は `z = 0` の方向からの z 軸負の向きに下がる仰角。つまり上からの照明は `altdeg = 90` で指定する。

```
from matplotlib.colors import LightSource

fig, ax = plot_fiducial()
light = LightSource(azdeg=0.0, altdeg=-20.0)
# light = LightSource(azdeg=270.0, altdeg=15.0)
theta = np.linspace(1e-5, np.pi / 2.0, 10).reshape(-1, 1)
_xyz = xyz(theta, phi)

ax.plot_surface(
    *_xyz,
    color=colors.sky,
    # facecolors=facecolor,
    alpha=1.0,
    shade=True,
    lightsource=light,
)
savefig(fig, './obipy-resources/light_3d.png')
```



LightSource は他に照明を当てた際の色の変化も指定できるが、**Axes3D.plot_surface()** が作る配色と異なるので注意する。