

matplotlib で 3 次元図の作成

sugayu

2025 年 1 月 1 日

目次

1 基準の図

2 軸の設定

- 2.1 ズーム
- 2.2 視点の角度
- 2.3 軸を消す
- 2.4 軸を完全に消す

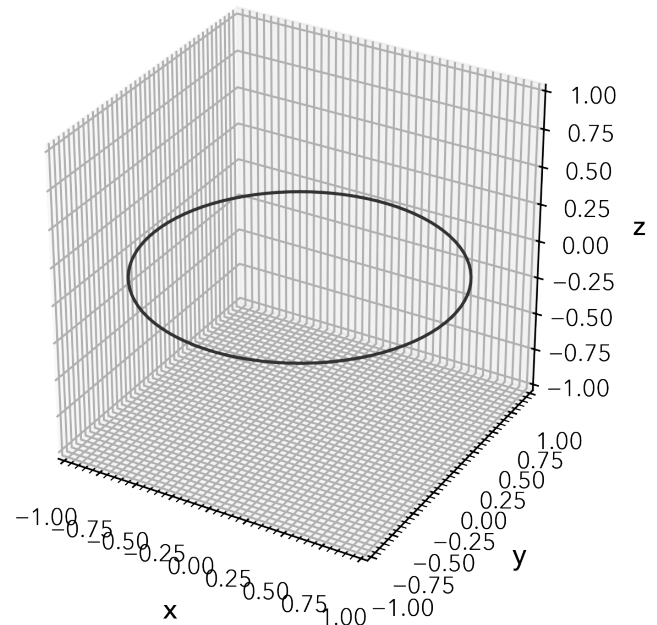
3 プロットの工夫

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。

```
1 from mpl_toolkits.mplot3d.axes3d import Axes3D
2 from sugayutils import colors
3
4 theta = np.pi / 2.0
5 phi = np.linspace(0.0, 2.0 * np.pi, 91)
6 x = np.sin(theta) * np.cos(phi)
7 y = np.sin(theta) * np.sin(phi)
8 z = np.cos(theta) * np.ones_like(phi)
9
10
11 def plot_fiducial(left=-0.1) -> Axes3D:
12     fig = plt.figure(figsize=(3.5, 3.5))
13     fig.subplots_adjust(left, 0.1, 0.98, 0.99,
14         ↪ 0.0, 0.0)
15     ax = fig.add_subplot(1, 1, 1,
16         ↪ projection='3d')
17     ax.plot(x, y, z, c=colors.black)
18     ax.set_xlim3d(-1.0, 1.0)
19     ax.set_ylim3d(-1.0, 1.0)
20     ax.set_zlim3d(-1.0, 1.0)
21     ax.set_xlabel('x')
22     ax.set_ylabel('y')
23     ax.set_zlabel('z')
24     ax.set_aspect('equal')
25     return fig, ax
```

```
26 def savefig(fig, fsave) -> str:
27     fig.savefig(fsave)
28     fig.clear()
29     plt.close()
30     return f'[[file:{fsave}]]'
31
32
33 fig, _ = plot_fiducial()
34 savefig(fig,
35     ↪ './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。

2 軸の設定

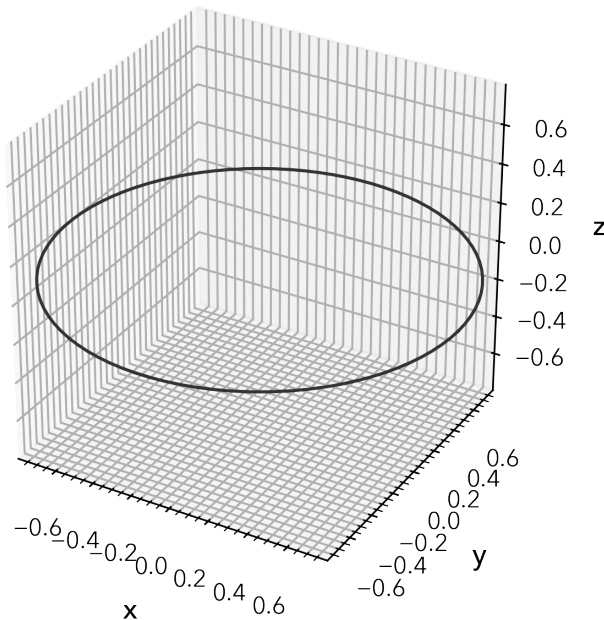
2.1 ズーム

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

```

1 fig, ax = plot_fiducial()
2 zoom = 1.3
3 ax.set_xlim3d(np.array(ax.get_xlim3d()) / zoom)
4 ax.set_ylim3d(np.array(ax.get_ylim3d()) / zoom)
5 ax.set_zlim3d(np.array(ax.get_zlim3d()) / zoom)
6 savefig(fig, './obipy-resources/zoom_3d.png')

```



```

[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')

```

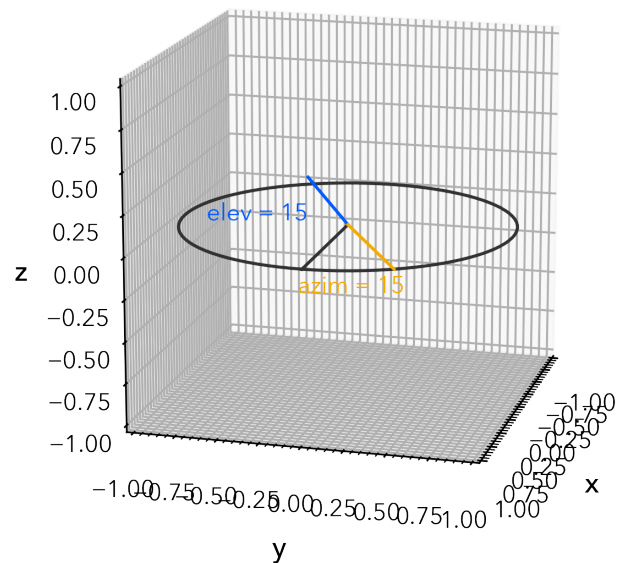
2.2 視点の角度

`Axes3D.view_init()` で変更する。パラメータの **azim** は **x** 軸正の向き (**y** = 0 の方向) から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。

```

1 fig, ax = plot_fiducial(left=0.1)
2
3 ax.view_init(elev=15.0, azim=15.0)
4 # ax.view_init(elev=30.0, azim=-60.0) #
  ↳ default
5
6 ax.plot(
7     [1.0, 0.0],
8     [0.0, 0.0],
9     [0.0, 0.0],
10    c=colors.black,
11 )
12 ax.plot(
13     [0.0, np.cos(np.pi / 6)],
14     [0.0, 0.0],
15     [0.0, np.sin(np.pi / 6)],
16     c=colors.blue,
17 )
18 ax.plot(
19     [0.0, np.cos(np.pi / 6)],

```



2.3 軸を消す

```

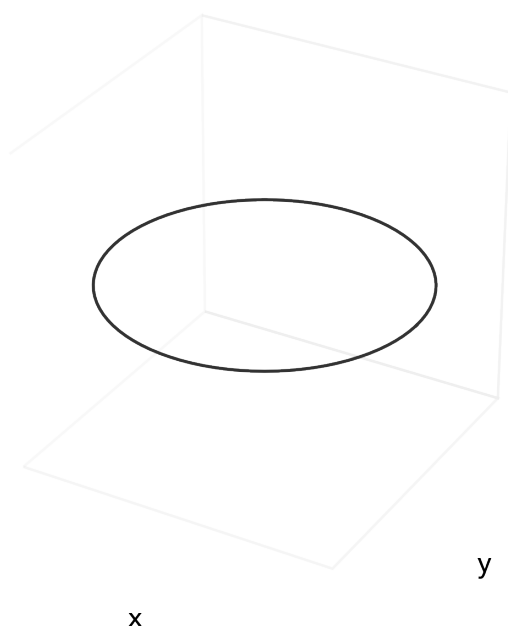
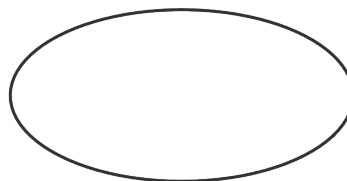
1 fig, ax = plot_fiducial()
2 ax.grid(False) # gridを消す
3 ax.xaxis.pane.fill = False # 壁を白くする
4 ax.yaxis.pane.fill = False
5 ax.zaxis.pane.fill = False
6 ax.set_xticks([]) # メモリを消す
7 ax.set_yticks([])
8 ax.set_zticks([])

```

```

9  ax.xaxis.line.set_color((1.0, 1.0, 1.0, 0.0)) #
   ↳ 軸を消す
10 ax.yaxis.line.set_color((1.0, 1.0, 1.0, 0.0))
11 ax.zaxis.line.set_color((1.0, 1.0, 1.0, 0.0))
12 ax.tick_params( # ラベルを消す? 消せない
13     which='both',
14     labelcolor='none',
15     top=False,
16     bottom=False,
17     left=False,
18     right=False,
19 )
20 savefig(fig, './obipy-resources/axis_3d.png')

```



2.4 軸を完全に消す

```

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

```

3 プロットの工夫

z 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。(2D のときは `mcoll.LineCollection`)

```

1  from mpl_toolkits.mplot3d import art3d
2
3  fig, ax = plot_fiducial()
4  ax.axis('off')
5
6  lim = 1.3
7  segments = (
8      ((-lim, 0.0, 0.0), (lim, 0.0, 0.0)),
9      ((0.0, -lim, 0.0), (0.0, lim, 0.0)),
10     ((0.0, 0.0, -lim), (0.0, 0.0, lim)),
11 )
12 linecollection =
   ↳ art3d.Line3DCollection(segments,
   ↳ colors=colors.black, lw=0.5, ls='--')
13 ax.add_collection(linecollection)
14 ax.text(lim + 0.1, 0.0, 0.0, 'x', ha='center',
   ↳ va='center')
15 ax.text(0.0, lim + 0.1, 0.0, 'y', ha='center',
   ↳ va='center')
16 ax.text(0.0, 0.0, lim + 0.1, 'z', ha='center',
   ↳ va='center')
17 savefig(fig, './obipy-resources/lines_3d.png')

```

