

# matplotlib で 3 次元図の作成

sugayu

2025 年 1 月 1 日

## 目次

### 1 基準の図

### 2 基本

#### 2.1 面をプロット . . . . .

### 3 軸の設定

#### 3.1ズーム . . . . .

#### 3.2 視点の角度 . . . . .

#### 3.3 軸を消す . . . . .

#### 3.4 軸を完全に消す . . . . .

### 4 プロットの工夫

#### 4.1 大量の線 . . . . .

#### 4.2 光の角度 . . . . .

### 1 基準の図

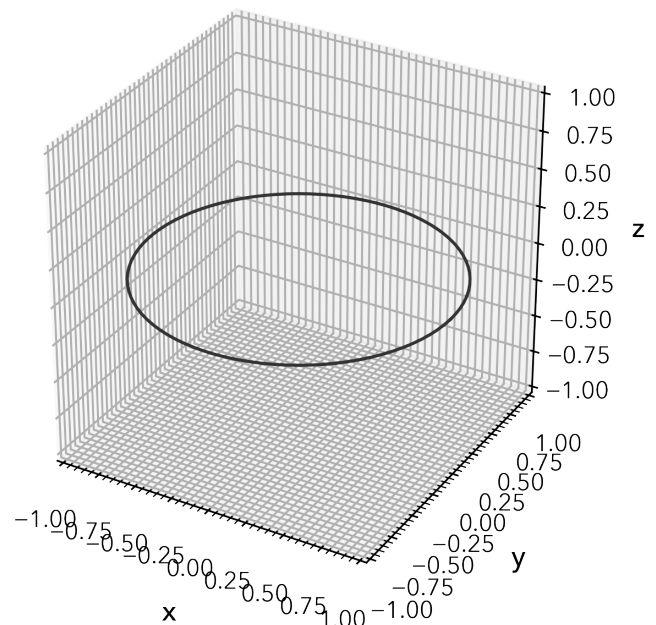
`fig.add_subplot()` に `projection='3d'` を渡して 3 次元図を作成する。このとき返ってくるのは 3 次元専用の軸オブジェクト `Axes3D` である。

```
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
7
8 def xyz(theta, phi):
9     x = np.sin(theta) * np.cos(phi)
10    y = np.sin(theta) * np.sin(phi)
11    z = np.cos(theta) * np.ones_like(phi)
12    return x, y, z
13
14
15 x, y, z = xyz(theta, phi)
16
17
18 def plot_fiducial(left=-0.1) -> Axes3D:
19     fig = plt.figure(figsize=(3.5, 3.5))
20     fig.subplots_adjust(left, 0.1, 0.98, 0.99,
21                          ↪ 0.0, 0.0)
22     ax = fig.add_subplot(1, 1, 1,
23                          ↪ projection='3d')
```

```
22 ax.plot(x, y, z, c=colors.black)
23 ax.set_xlim3d(-1.0, 1.0)
24 ax.set_ylim3d(-1.0, 1.0)
25 ax.set_zlim3d(-1.0, 1.0)
26 ax.set_xlabel('x')
27 ax.set_ylabel('y')
28 ax.set_zlabel('z')
29 ax.set_aspect('equal')
30 return fig, ax
```

```
33 def savefig(fig, fsave) -> str:
34     fig.savefig(fsave)
35     fig.clear()
36     plt.close()
37     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、

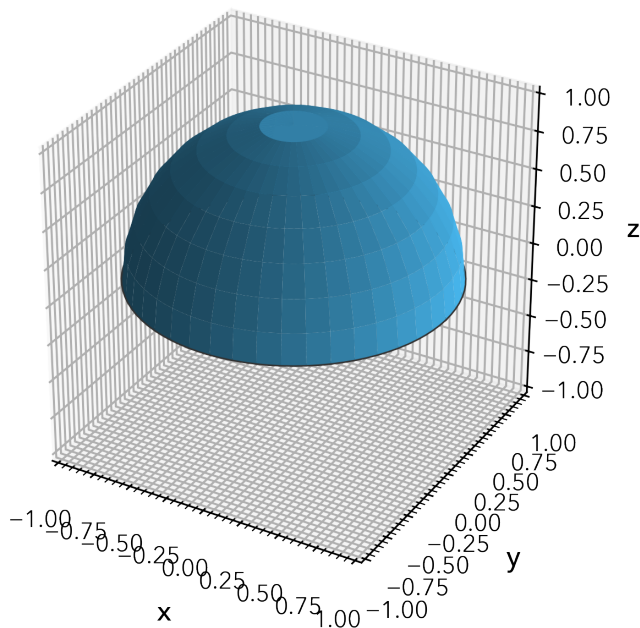
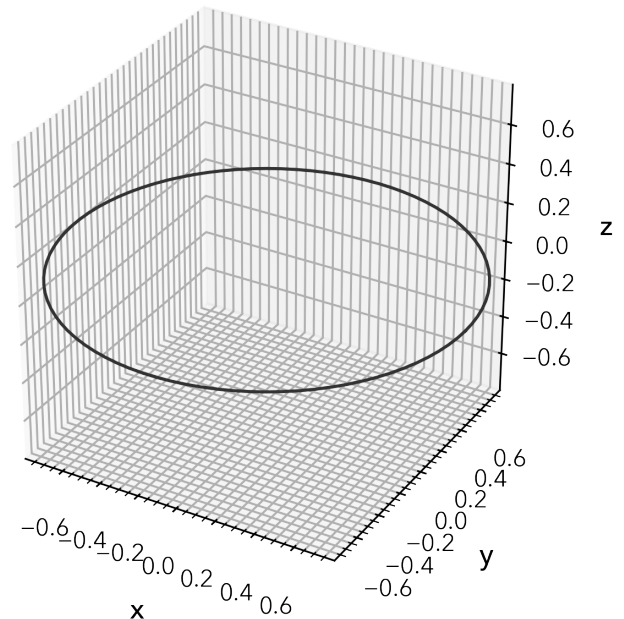
ここでは `ob-ipython` のファイル保存形式ではなく、保存方法と表示方法を指定している。ただし出力のたび、いちいち手作業が必要になる。また、この場合の方がプロットがとても速い。

```
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')
```

## 2 基本

### 2.1 面をプロット

```
1 fig, ax = plot_fiducial()
2 theta = np.linspace(0.0, np.pi / 2.0,
   ↳ 9).reshape(-1, 1)
3 _xyz = xyz(theta, phi)
4 ax.plot_surface(
5     *_xyz,
6     color=colors.sky,
7     # facecolors=facecolor,
8     alpha=1.0,
9     # shade=True,
10    # lightsource=light,
11 )
12 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。

```
1 fig, ax = plot_fiducial()
2 zoom = 1.3
```

### 3.2 視点の角度

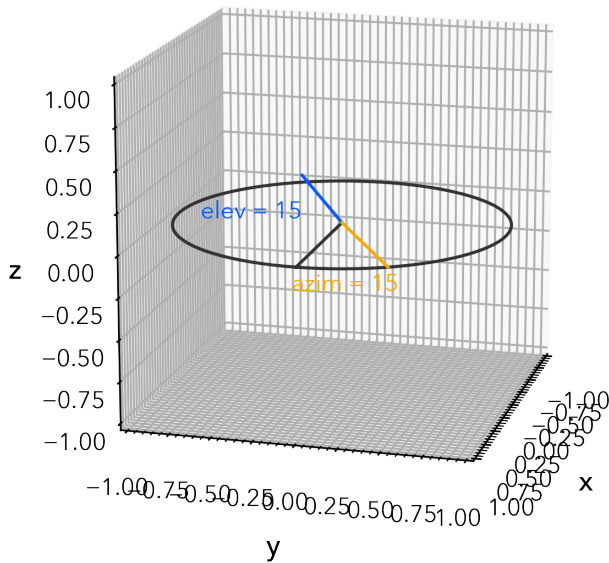
`Axes3D.view_init()` で変更する。パラメータの角度方向は直感の通りで、`azim` は x 軸正の向き ( $y = 0$  の方向) から反時計まわりの方位角、`elev` は  $z = 0$  の方向からの z 軸正の向きに上がる仰角。全ての角度は単位は度で入力する。

```
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)],
20     [0.0, np.sin(np.pi / 6)],
21     [0.0, 0.0],
22     c=colors.orange,
```

```

23 )
24 ax.text(
25     np.cos(np.pi / 6),
26     0.0,
27     np.sin(np.pi / 6) / 2.0,
28     'elev = 15',
29     ha='right',
30     color=colors.blue,
31 )
32 ax.text(
33     np.cos(np.pi / 6),
34     np.sin(np.pi / 6) / 2.0,
35     -0.05,
36     'azim = 15',
37     va='top',
38     ha='center',
39     color=colors.orange,
40 )
41
42 savefig(fig, './obipy-resources/view_3d.png')

```



### 3.3 軸を消す

```

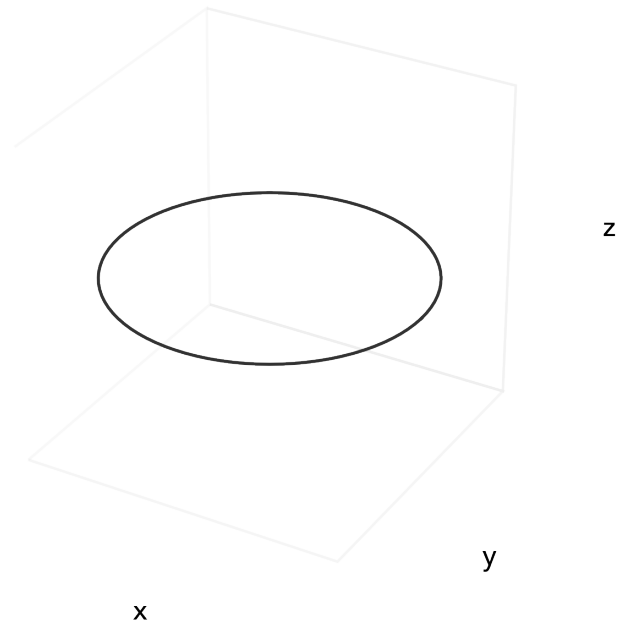
1 fig, ax = plot_fiducial()
2 ax.grid(False) # gridを消す
3 ax.xaxis.pane.fill = False # 壁を白くする
4 ax.yaxis.pane.fill = False
5 ax.zaxis.pane.fill = False
6 ax.set_xticks([]) # メモリを消す
7 ax.set_yticks([])
8 ax.set_zticks([])
9 ax.xaxis.line.set_color((1.0, 1.0, 1.0, 0.0)) #
  ↳ 軸を消す
10 ax.yaxis.line.set_color((1.0, 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')

```

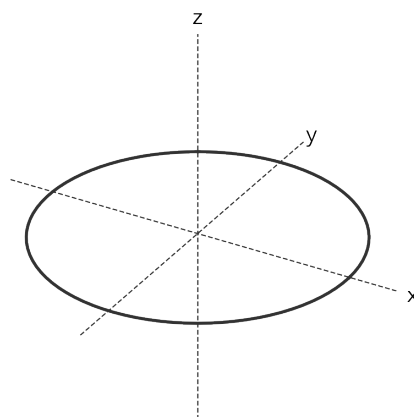
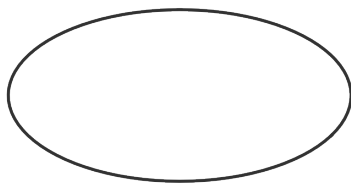


### 3.4 軸を完全に消す

```

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

```



## 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。

---

```

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 =
13     ↪ art3d.Line3DCollection(segments,
14     ↪ colors=colors.black, lw=0.5, ls='--')
15 ax.add_collection(linecollection)
16 ax.text(lim + 0.1, 0.0, 0.0, 'x', ha='center',
17     ↪ va='center')
18 ax.text(0.0, lim + 0.1, 0.0, 'y', ha='center',
19     ↪ va='center')
20 ax.text(0.0, 0.0, lim + 0.1, 'z', ha='center',
21     ↪ va='center')
22 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。

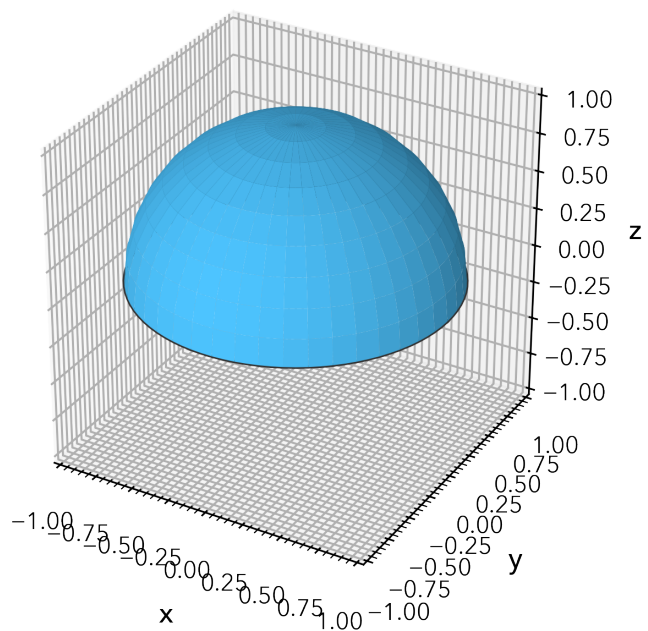
---

```

1 from matplotlib.colors import LightSource
2
3 fig, ax = plot_fiducial()
4 light = LightSource(azdeg=0.0, altdeg=-20.0)
5 # light = LightSource(azdeg=270.0, altdeg=15.0)
6 theta = np.linspace(1e-5, np.pi / 2.0,
7     ↪ 10).reshape(-1, 1)
8 _xyz = xyz(theta, phi)
9 ax.plot_surface(
10     *_xyz,
11     color=colors.sky,
12     # facecolors=facecolor,
13     alpha=1.0,
14     shade=True,
15     lightsource=light,
16 )
17 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。