

第 1 版第 2 刷の正誤表

2021 年 8 月 5 日

● 準備

- 1 ページ、注 1、日付を変更
 - * 誤：2020 年 11 月時点
 - * 正：2021 年 6 月時点
 - * 事由：最新の状況に合わせる
- 2 ページ、13 行目、インストールするライブラリを変更
 - * 誤：pip3 install `jupyter`
 - * 正：pip3 install `jupyterlab`
 - * 事由：0.9 節で JupyterLab に触れるため
- 2 ページ、15 行目、Jupyter を JupyterLab に変更
 - * 誤：MatplotLib、PIL、`Jupyter`^{注 5} の各ライブラリ
 - * 正：MatplotLib、PIL、`JupyterLab`^{注 5} の各ライブラリ
 - * 事由：2 ページ 13 行目の変更に関連して
- 2 ページ、注 5、文言を追加
 - * 誤：Jupyter Notebook を利用
 - * 正：Jupyter Notebook および JupyterLab を利用
 - * 事由：2 ページ 13 行目、15 行目の変更に関連して
- 3 ページ、6 行目、文章を追加して太文字で強調
 - * 誤：(次をゴシック体で追加)
 - * 正：**Raspberry Pi 4 または 400 は、10 ページ「0.6 VPython の利用」の Raspberry Pi の場合のインストールを先に実行してください。**
 - * 事由：3 ページ 13 行目の変更に関連して
- 3 ページ、13 行目、ライブラリ Jupyter のインストールを削除
 - * 誤：`sudo apt install jupyter`
 - * 正：(上を削除)
 - * 事由：このやり方では、VPython インストール時にエラーが生じるようになったため
- 3 ページ、16 行目、Jupyter を削除
 - * 誤：SciPy、SymPy、MatplotLib、`Jupyter` の各ライブラリ
 - * 正：SciPy、SymPy、MatplotLib の各ライブラリ
 - * 事由：3 ページ 13 行目の変更に関連して
- 3 ページ、注 6、OS 名とタイムスタンプを変更
 - * 誤：本書執筆時の最新のイメージファイル `Raspbian Buster with desktop and recommended software` のバージョンは `August 2020` です。
 - * 正：本書執筆時の `Raspberry Pi OS (32bit)` の最新リリースは `2021-05-07` です。

* 事由：OS の正式名称が変更になったため

– 10, 11 ページ、文章を差し替え

* 誤：最新の Raspberry Pi 4 Model B では 3D や動画の表示がパソコンに近くなったので、VPython をインストールすることができます。apt コマンドでインストールできるパッケージが用意されていないので、pip コマンドを使ってインストールします。Python2 の pip コマンドと区別するために、次のように pip3 コマンドを用います。

```
sudo pip3 install --upgrade pip
sudo pip3 install vpython
```

apt コマンドを使ってインストールできる最新のライブラリと、pip コマンドを使ってインストールできる最新のライブラリのバージョンが違うことが原因で、VPython のインストールが失敗するかもしれません。その場合は、VPython のバージョンを下げてインストールするか、apt コマンドでインストールした Jupyter に含まれるライブラリを pip コマンドでアップグレードする必要が生じます注²²。VPython のインストールに失敗した場合や、あるいはその性能面の理由から Raspberry Pi 3 以下のモデルでは、VPython をインストールするのを諦めます。しかし、Raspberry Pi OS にはその欠点を補って余りある、Mathematica がプレインストールされています。3D の画像を描きたいときには、勉強して利用してみるのもよいでしょう。

* 正：Raspberry Pi 4/400 は性能がパソコンに匹敵し、VPython や後述する Jupyter Notebook が使えます。apt コマンドでインストールできるパッケージが用意されていないので、3 ページのインストールを行う前に、以下のコマンドを実行します。

```
python3 -m pip install --upgrade --user pip
python3 -m pip install --user jupyterlab
python3 -m pip install --user vpython
```

Python のライブラリは、多くの他のライブラリに依存しており、必要とするバージョンが違っていると正しく動きません。apt コマンドも pip コマンドも、それぞれライブラリ間の整合性を保つように管理していますが、別々のデータベースで管理されるので、併用すると矛盾が生じることがあります。ここでは pip コマンドを apt コマンドより先に、しかも利用者権限で用いることでそれを回避しています注²²。Raspberry Pi 3 以下のモデルでは性能面の理由から、VPython や Jupyter Notebook のインストールを諦めます。しかし、Raspberry Pi OS には Mathematica が Recommended Software として用意されており、Raspberry Pi 3 以下でも動きます。3D の画像を描きたいときには、勉強して利用してみるのもよいでしょう。

* 事由：本書のやり方ではエラーが生じるようになったため

– 11 ページ、注 22、文章を差し替え

* 誤：最新のバージョンの違いで、対処の仕方が異なることがあります。VPython のダウングレードについては 0.9 節で触れているので、そちらを参照してください。VPython の Raspberry Pi 4 への実装は本書執筆時はまだインターネットにほとんど報告がなく、ここでは試験的なものと考えてください。

* 正：VPython および Jupyter のシステムは非常に多くの他のライブラリに関連するため、それらのライブラリとの整合性の状況が日々変わることがあります。本書の記述のままインストールがうまくいかないときは、本書のサポートページを参考にしてください。

* 事由：本文の変更に伴って

– 18 ページ、11 ~ 13 行目、文言を削除

* 誤：再度 Notebook を起動して notebook/ に移動し、カーネルとして「VPython」(もし「New」のプルダウンメニューに「VPython」というカーネル名が見当たらなければ注²⁵「Python3」)を選択して、新しい Notebook 「My2ndNotebook」を作ります。(赤字を削除、注 25 は後の変更箇所へ移動)

- * 正：再度 Notebook を起動して notebook/ に移動し、新しい Notebook 「My2ndNotebook」を作ります。
- * 事由：VPython という項目があるなしはシステムにより変わり、たとえある場合でもどちらを選択しても構わないため。

– 18, 19 ページ、文章を差し替え

- * 誤：macOS の場合も、Notebook から VPython を使う場合は、Anaconda をインストールして利用する方法が確実で早道です。本書の執筆時に、これまで説明してきた仕方で macOS や Raspberry Pi に Jupyter と VPython (執筆時点での最新バージョン 7.6.1) をインストールしたところ、Notebook は正常に機能しましたが、Notebook から VPython を使うと画像が表示されないという現象が起きました。次のコマンドで VPython を一つ前のバージョン 7.5.1 にダウングレードすると解決できたので、参考にしてください (Raspberry Pi の場合は pip3 の部分を sudo pip3 に置き換えます)。

```
pip3 install vpython==7.5.1
```

Python の環境は日進月歩で変化するため、注意が必要です。本書では第 1 章以降では Notebook を用いることはありませんが、読者は本書のコードを Notebook で実行することにも挑戦してみてください。図 8 右は、Raspberry Pi 4 で Notebook を立ち上げて VPython を実行した例です。

- * 正：なお、Jupyter Notebook を含む様々な Web アプリが利用できる JupyterLab というツールもあり、Anaconda3 のメニューから「JupyterLab」をクリックするか、ターミナルから「jupyter-lab」または「jupyter lab」というコマンドで立ち上がります。JupyterLab はアドオンで機能を付け加えることができます。この場合、Node.js というシステムを予めインストールしておく必要があります。Node.js は <https://nodejs.org> からダウンロードできます。Windows と macOS には専用のインストーラが、Raspberry Pi 4/400 には Linux Binaries (ARM) が用意されています^{注 25}。JupyterLab で VPython を機能させるには次のコマンドで機能拡張します。

```
jupyter labextension install vpython
```

JupyterLab の Notebook で、Jupyter Notebook と同じように VPython が利用できます (図 8)。pip コマンドにより Node.js が不要な jupyterlab-vpython をインストールする方法もあります。labextension でエラーが出る場合は、こちらを利用してみてください。

- * 事由：JupyterLab について新たに言及したため

– 18 ページ、注 25、文章を差し替え

- * 誤：プラットフォームの違いやインストールの仕方の違いで変わることがあります。
- * 正：32 ビットの ARMv7 を使います。詳しくはサポートページをご覧ください。
- * 事由：注を付ける場所の変更に伴って

– 19 ページ、図 8 およびキャプションを差し替え

- * 誤：

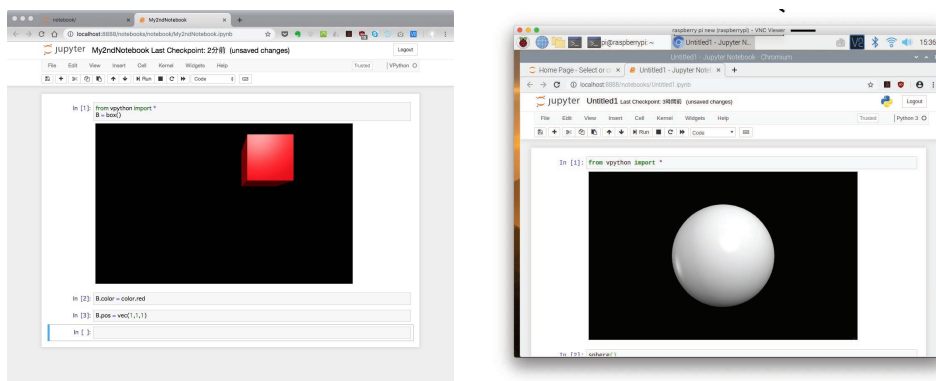


図 8 Notebook からの VPython の利用 (左) と Raspberry Pi での実行例 (右)

* 正:

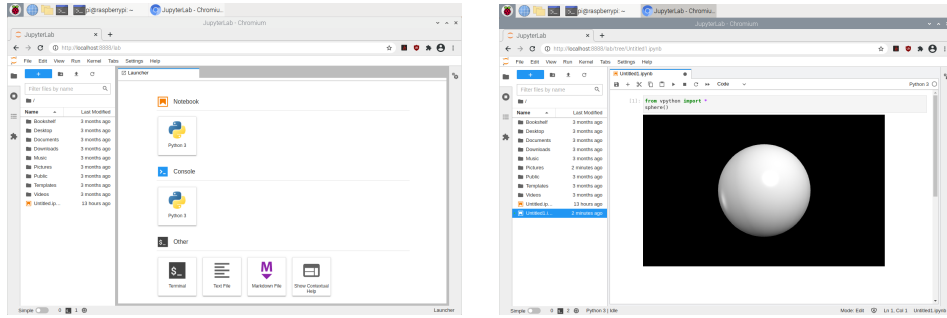


図 8 Raspberry Pi による JupyterLab の起動画面 (左) と VPython の実行例 (右)

* 事由: JupyterLab の記述を加えたため

- 23 ページ、注 27、文章を差し替え

* 誤: 0.9 節で触れた VPython のダウングレード方法を参考にしてください。

* 正: pip コマンドの場合、`vpypython==7.5.1` のようにバージョンを指定してインストールできます。

* 事由: 本文の変更に伴って

● 第 1 章 数学の基礎と Python による表現

- 46 ページ、プログラム `lena1.py`、行番号 14 のコードを差し替え

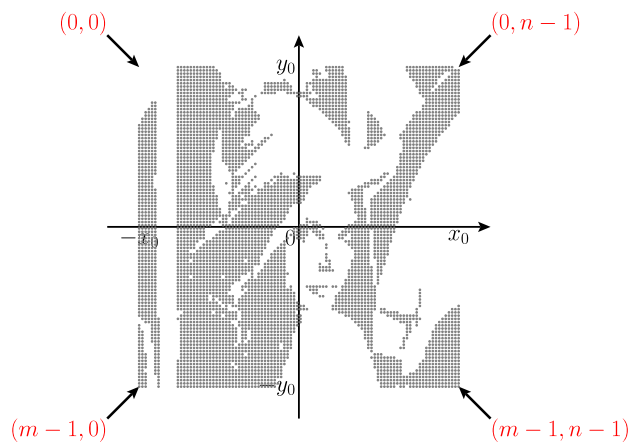
* 誤: `return (y0 * (-1+2*j/(n-1)), x0 * (1-2*i/(m-1)))`

* 正: `return (x0 * (-1+2*j/(n-1)), y0 * (1-2*i/(m-1)))`

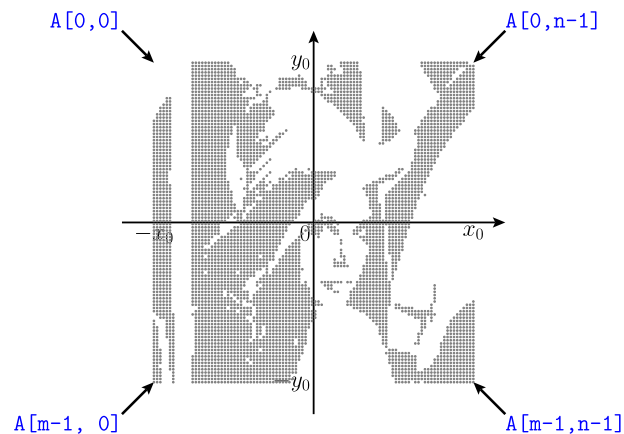
* 事由: 画像が正方形でないとき、縦横比が逆になる

- 48 ページ、図 1.3 を差し替え

* 誤:



* 正:



* 事由: 図が誤解を招いたため

- 第2章 線形空間と線形写像

- 65 ページ、プログラム cfunc.py、コードを差し替え

* 誤：

```
1 from numpy import exp, pi, linspace
2 import matplotlib.pyplot as plt
3 from mpl_toolkits.mplot3d import Axes3D
4
5 f = lambda n, x: exp(1j * n * x)
6 x = linspace(0, 2 * pi, 1001)
7 ax = Axes3D(plt.figure())
8 for n in range(-2, 3):
9     ax.plot(x, f(n, x).real, f(n, x).imag)
10 plt.show()
```

* 正：3 行目を削除して詰める、7 行目を変更、9 行目を 2 行に分けて書く

```
1 from numpy import exp, pi, linspace
2 import matplotlib.pyplot as plt
3
4 f = lambda n, x: exp(1j * n * x)
5 x = linspace(0, 2 * pi, 1001)
6 plt.subplot(111, projection='3d')
7 for n in range(-2, 3):
8     z = f(n, x)
9     plt.plot(x, z.real, z.imag)
10 plt.show()
```

* 事由：MatplotLib の仕様変更でワーニングが出るようになったため、推奨される書き方に変更。

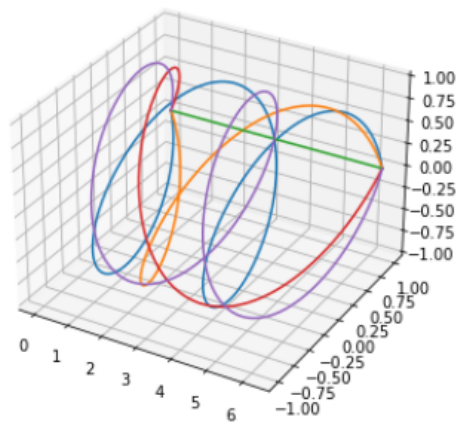
【参考】今のところ、pip コマンドで最新の MatplotLib をインストールしなければ、このワーニング（下図参照）は出ないかもしれないが、この書き方のほうがわかりやすいので修正することとした。

In [1]:

```
from numpy import exp, pi, linspace
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

f = lambda n, x: exp(1j * n * x)
x = linspace(0, 2 * pi, 1001)
ax = Axes3D(plt.figure())
for n in range(-2, 3):
    ax.plot(x, f(n, x).real, f(n, x).imag)
plt.show()
```

```
<ipython-input-1-74b63c9e24f5>:7: MatplotlibDeprecationWarning: Axes3D
(fig) adding itself to the figure is deprecated since 3.4. Pass the ke
yword argument auto_add_to_figure=False and use fig.add_axes(ax) to su
ppress this warning. The default value of auto_add_to_figure will chan
ge to False in mpl3.5 and True values will no longer work in 3.6. Thi
s is consistent with other Axes classes.
  ax = Axes3D(plt.figure())
```



– 66 ページ、1 行目を削除

- * 誤：3 行目 … 3 次元グラフを描くために Axes3D を用います。
- * 正：(この行をすべて削除、詰める)
- * 事由：プログラムのコード変更のため

– 66 ページ、2 行目、行番号をずらす

- * 誤：5 行目
- * 正：4 行目
- * 事由：プログラムのコード変更のため

– 66 ページ、3 行目、行番号をずらす

- * 誤：6 行目
- * 正：5 行目
- * 事由：プログラムのコード変更のため

– 66 ページ、4 行目、行番号をずらす

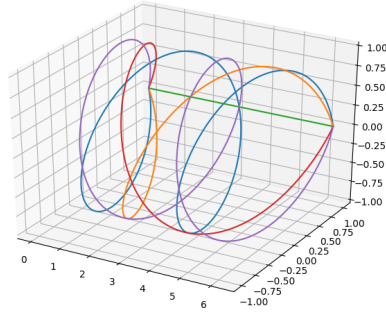
- * 誤：7 行目 … 3 次元グラフを設定します。
- * 正：6 行目 … subplot の最初の引数が 3 桁の整数 lmn のとき、図を l 行 m 列に分割して、左上隅から数えて n 番目の位置に座標軸を設定することを意味します。 l, m, n は最初の 3 つの引数にもできます。このコードでは一つだけなので 111 を与え、ここに 3 次元の座標軸を設定します。
- * 事由：プログラムのコード変更のため。行のずれをここで吸収、必要ならば図 2.5 の大きさを変える。

– 66 ページ、5 行目、行番号をずらす

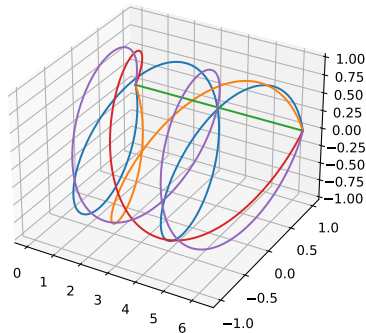
- * 誤：8～10行目
- * 正：7～10行目
- * 事由：プログラムのコード変更のため

– 66 ページ、図 2.5 を差し替え

- * 誤：



- * 正：



- * 事由：変更後のプログラムの出力を用いる

– 66 ページ、図 2.5 のキャプションの数式、上付き内を変更

- * 誤： $\mathbb{C}^{[-\pi, \pi]}$
- * 正： $\mathbb{C}^{[0, 2\pi]}$
- * 事由：タイポ

● 第4章 行列

– 115 ページ、7 行目、文言を追加

- * 誤： xy -座標平面にグラフ化します。行列は `matrix` を使って
- * 正： xy -座標平面にグラフ化します (図 4.2)。行列は `matrix` を使って
- * 事由：116 ページの図 4.2 の引用がなかったため

● 第6章 内積とフーリエ展開

– 172 ページ、3 行目、数式を変更

- * 誤：

$$e_k(t) \stackrel{\text{def}}{=} \begin{cases} \frac{\sin(2\pi kt)}{\sqrt{2}} & (k < 0 \text{ のとき}) \\ 1 & (k = 0 \text{ のとき}) \\ \frac{\cos(2\pi kt)}{\sqrt{2}} & (k > 0 \text{ のとき}) \end{cases} \quad k = 0, \pm 1, \pm 2, \dots$$

- * 正：

$$e_k(t) \stackrel{\text{def}}{=} \begin{cases} \sqrt{2} \sin(2\pi kt) & (k < 0 \text{ のとき}) \\ 1 & (k = 0 \text{ のとき}) \\ \sqrt{2} \cos(2\pi kt) & (k > 0 \text{ のとき}) \end{cases} \quad k = 0, \pm 1, \pm 2, \dots$$

* 事由：プログラム `trigonometric.py` と食い違いがあるが、本文中の数式の方が誤り（読者からの指摘によるもの）

– 176 ページ、プログラム `fourier.py`、行番号 14 以下のコード差し替え

* 誤：

```

14 if __name__ == '__main__':
15     import matplotlib.pyplot as plt
16     from mpl_toolkits.mplot3d import Axes3D
17     k = 2
18     t = arange(0, 1, 1 / 1000)
19     x = [z.real for z in e(k, t)]
20     y = [z.imag for z in e(k, t)]
21     fig = plt.figure()
22     ax = Axes3D(fig)
23     ax.scatter(t, x, y, s=1)
24     ax.plot(t, x, y)
25     plt.show()

```

* 正：16 行目を空行にする。22 行目および 25 行目を変更

```

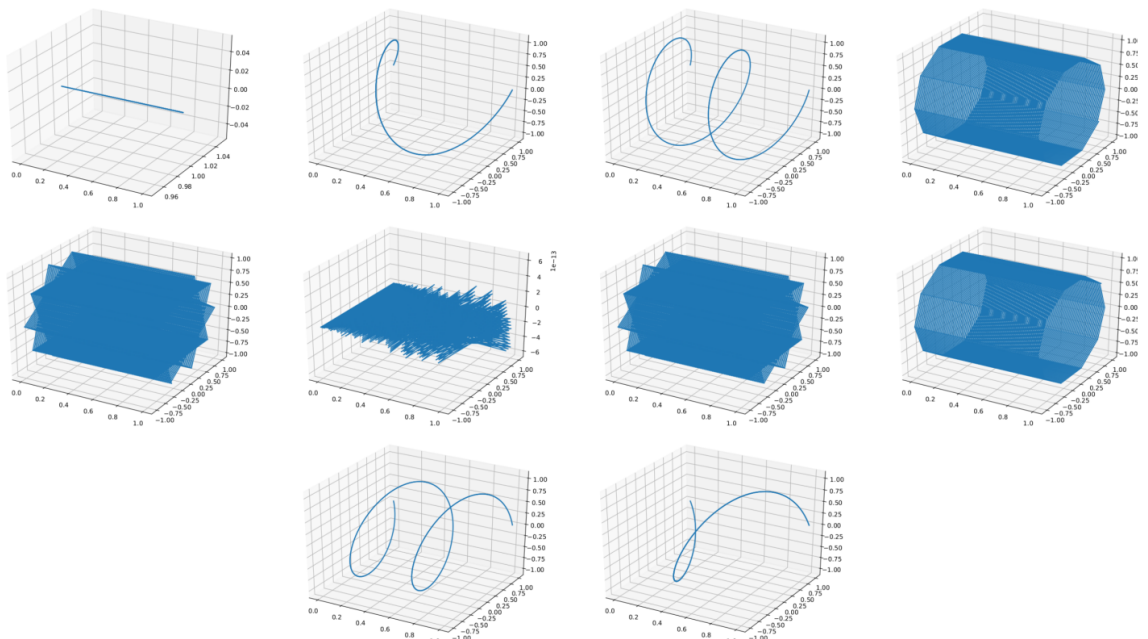
14 if __name__ == '__main__':
15     import matplotlib.pyplot as plt
16
17     k = 2
18     t = arange(0, 1, 1 / 1000)
19     x = [z.real for z in e(k, t)]
20     y = [z.imag for z in e(k, t)]
21     fig = plt.figure()
22     ax = fig.add_subplot(111, projection="3d")
23     ax.scatter(t, x, y, s=1)
24     ax.plot(t, x, y)
25     ax.set_xlim(0, 1), ax.set_ylim(-1, 1), ax.set_zlim(-1, 1), plt.show()

```

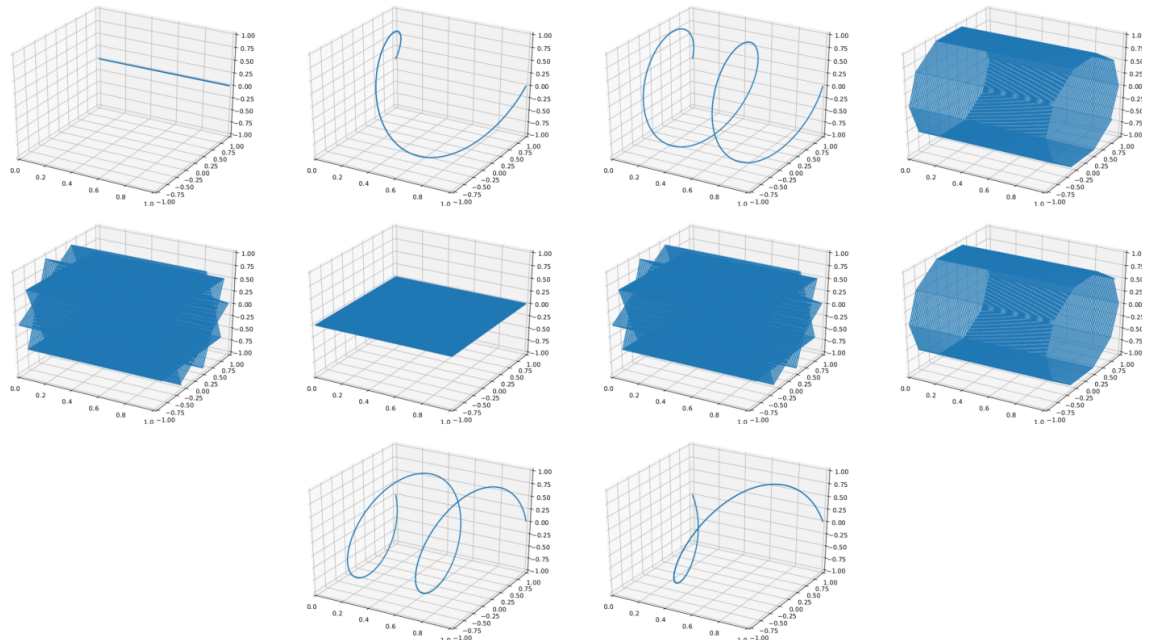
* 事由：Matplotlib の仕様変更でワーニングが出るようになったため、推奨される書き方に変更。グラフの目盛りのスケールを統一。特に、 $k=0$ および $k=500$ で不統一が顕著になるのを改めた。

– 177 ページ、図 6.8 を差し替え

* 誤：



* 正：



* 事由：変更したプログラムの出力図にする

– 185 ページ、10 行目、数式を変更

* 誤： $\int_{-\infty}^{\infty} e^{-2\pi i k \omega t} x(t) dt$ (k をトル)

* 正： $\int_{-\infty}^{\infty} e^{-2\pi i \omega t} x(t) dt$

* 事由：タイポ

● 第 7 章 固有値と固有ベクトル

– 195 ページ、下から 7 行目、 A の場所を移動

* 誤： $|\langle Ax | y \rangle|^2 \leq \langle x | Ax \rangle \langle y | Ay \rangle$

* 正： $|\langle x | Ay \rangle|^2 \leq \langle x | Ax \rangle \langle y | Ay \rangle$

* 事由：タイポ

– 199 ページ、4, 5 行目、文章を修正

* 誤：一方、 $\text{range}(A)$ は 1 次元なので、列ベクトルで零ベクトルでないものは、

* 正：一方、 $\text{range}(A)$ のベクトルで $\text{kernel}(A)$ の要素でないものは、

* 事由：数学的に主張が間違っていたので

● 第 8 章 ジョルダン標準形とスペクトル集合

– 224 ページ、下から 7 行目、上付き n を上付き k に変更

* 誤： $f^n : V \rightarrow V$

* 正： $f^k : V \rightarrow V$

* 事由：タイポ

– 227 ページ、プログラムを除く下から 3 行目、 I に下付き i を付加

* 誤： $(K_i - \lambda_i I)^{n_i}$

* 正： $(K_i - \lambda_i I_i)^{n_i}$

* 事由：タイポ

● 第 9 章 力学系

– 261 ページ、図 9.7 の下 2 行目の数式、総和記号 \sum の下の j を i に変更

* 誤: $\sum_{j=1}^n x(\omega_j)$

* 正: $\sum_{i=1}^n x(\omega_i)$

* 事由: タイポ

- 265 ページ、8 行目の数式、総和記号 \sum の下の \in を \subseteq に変更

* 誤: $\sum_{B \in X}$

* 正: $\sum_{B \subseteq X}$

* 事由: タイポ

- 273 ページ、17 行目、 i を j に変更

* 誤: 確率的に動き回り、状態 i に

* 正: 確率的に動き回り、状態 j に

* 事由: タイポ

● 第 10 章 線形代数の応用と発展

- 281 ページ、図 10.1 の下の行、大文字 N を小文字 m (エム) に変更

* 誤: N 等分

* 正: m 等分

* 事由: タイポ

- 299 ページ、3 行目、 K (**mathbold**) を \mathbb{R} (**mathbb**) に変更

* 誤: K^n の正規直交行列

* 正: \mathbb{R}^n の正規直交行列

* 事由: タイポ

- 299 ページ、5 行目、 \mathbb{K} を \mathbb{R} に変更

* 誤: \mathbb{K}^n の正規直交基底

* 正: \mathbb{R}^n の正規直交基底

* 事由: タイポ

- 308 ページ、13~15 行目、数式を変更

* 誤: であり、 $\text{range}(A^T)$ 上への直交射影は $A^T(A^T)^\dagger = (A^\dagger A)^T$ なので、

$$z_1 = (A^\dagger A)^T e_1, z_2 = (A^\dagger A)^T e_2, \dots, z_n = (A^\dagger A)^T e_n$$

がそれぞれ Z_1, Z_2, \dots, Z_n の表現になります。

* 正: であり、 $\text{range}(A^T)$ 上への直交射影は $A^\dagger A$ なので、

$$z_1 = (A^\dagger A) e_1, z_2 = (A^\dagger A) e_2, \dots, z_n = (A^\dagger A) e_n$$

がそれぞれ Z_1, Z_2, \dots, Z_n の表現になります。

* 事由: 簡潔に表現できるため

- 308 ページ、下から 7 行目、0.5 を $\sqrt{3}$ に変更

* 誤: -0.5 と 0.5 の間の一様分布

* 正: $-\sqrt{3}$ と $\sqrt{3}$ の間の一様分布

* 事由: 理論的に計算が違っていたため

- 308 ページ、下から 5 行目、文言を追加

* 誤: 規分布とすると

- * 正：規分布とし、 X_1 、 X_2 、 X_3 、 X_4 が互いに独立とすれば
 - * 事由：条件が不足していたため
- 309 ページ、プログラム estimate1.py、行番号 1 のコードを変更
- * 誤：


```
from numpy import array, random, linalg
```
 - * 正：

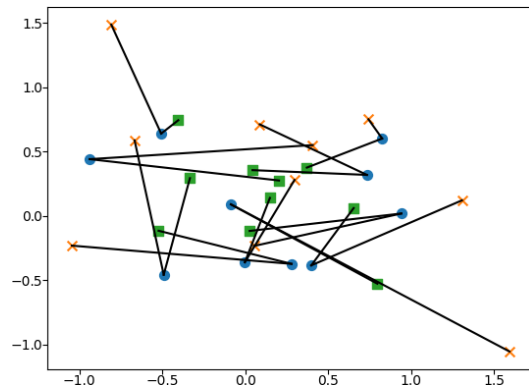

```
from numpy import sqrt, array, random, linalg
```
 - * 事由：308 ページ下から 7 行目の変更に伴って
- 309 ページ、プログラム estimate1.py、行番号 4~10 のコードを変更
- * 誤：


```
n = 10
x1 = [random.uniform(-0.5, 0.5) for n in range(n)]
x2 = [random.uniform(-0.5, 0.5) for n in range(n)]
x3 = [random.normal(0, 1) for n in range(n)]
x4 = [random.normal(0, 1) for n in range(n)]
A = array([[1, 0, 1, 0],
           [0, 1, 0, 1]])
```
 - * 正：

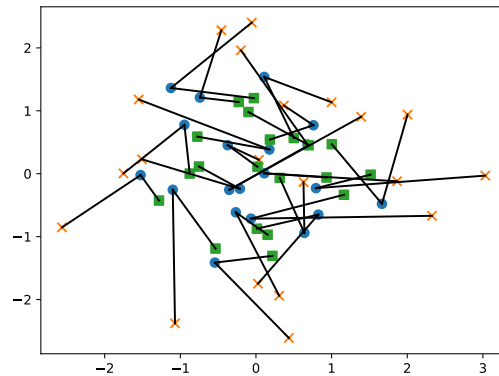

```
n = 20
random.seed(123)
x1 = random.uniform(-sqrt(3), sqrt(3), n)
x2 = random.uniform(-sqrt(3), sqrt(3), n)
x3 = random.normal(0, 1, n)
x4 = random.normal(0, 1, n)
A = array([[1, 0, 1, 0], [0, 1, 0, 1]])
```
 - * 事由：308 ページ下から 7 行目の変更に伴って。また、サンプル数を増やす、乱数の種は estimate2.py と同じ 123 を与える、読みやすいコードにするなど改良。
- 310 ページ、実行例、数値を変更
- * 誤：


```
||x-y||^2 = 1.7952353126663972
||x-z||^2 = 0.36093586643070026
```
 - * 正：


```
||x-y||^2 = 2.414280376687304
||x-z||^2 = 0.7913634686181286
```
 - * 事由：プログラムの変更に伴って出力が変わるため
- 310 ページ、図 10.11 を差し替え
- * 誤：



* 正：



* 事由：プログラムの変更に伴って出力が変わるため

– 314 ページ、注意 2、枠内の文章に記号を追加

* 誤：時間に依存して ρ_k と変化

* 正：時間に依存して σ_k 、 τ_k 、 ρ_k と変化

* 事由：タイポ

– 315 ページ、プログラム kalman.py、行番号 15 のコードを変更

* 誤：

```
a = r * c + s**2
```

* 正：

```
a = r**2 * c + s**2
```

* 事由：タイポ。プログラムの変更に伴って出力に変化はないので、312 ページの実行結果、及び、図 10.13 は変更なし。

● あとかぎに代えて

– 317 ページ、9 行目、文言を追加

* 誤：[5] 佐武一郎、『線型代数学』、裳華房、1958.

* 正：[5] 佐武一郎、『線型代数学』、裳華房、1958. 『行列と行列式』、1974 (増補改題).

* 事由：初版時とタイトルが異なるため