

Python 学習会 12月

木下

2019/11/25 Rev. 0

2019/12/01 Rev. 1

今月の内容

- matplotlib を使ったグラフ表示
- 簡単なモデルフィッティング (近似曲線)

使用するモジュール

- matplotlib
- numpy
- scipy

matplotlib を使ったグラフの作成

基本

1. モジュールのインポート

matplotlib.pyplot を plt という名前でインポートする。

```
from matplotlib import pyplot as plt
```

2. 配列 (np.array または list) のデータ x, y をプロットする。

```
plt.plot(x, y)
```

3. グラフを表示する, または保存する。

表示 `plt.show()` →

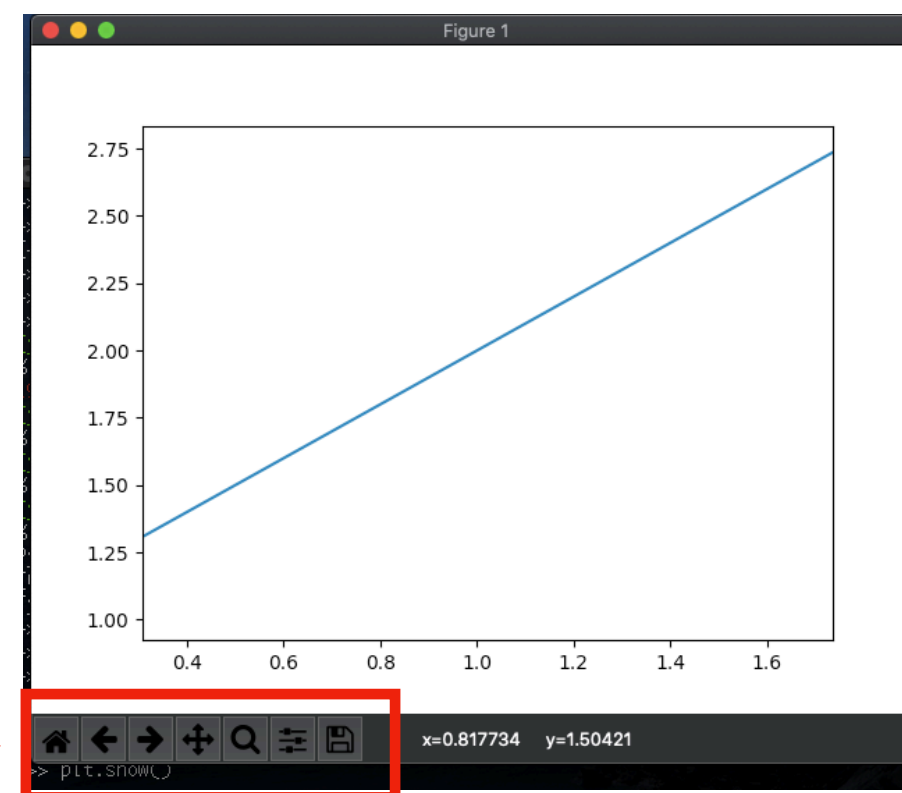
→ プログラムを一時停止し, グラフを画面に表示する
グラフを閉じるとプログラムが再開する。

保存 `plt.savefig("ファイル名")`

→ 各種画像形式 (*.png, *.pdf, *.jpeg など)
で保存する。

ファイル名の拡張子でファイル形式が決まる。
拡張子を指定しない場合 .png で保存する。

ツール→



plt.show() で表示したときの画面 (macOS)
拡大・縮小, 表示中グラフの保存などできる。

matplotlib を使ったグラフの作成

注意点

- 前ページのコマンド的にプロットしていく方法とは別に、オブジェクトを作ってからプロットしていく方法もある。コード行数は増えるが、以下の利点がある。
 - plt.show() の後にグラフを保存できる。(※1)
 - 詳細なグラフの体裁が設定できる。(2段組みグラフなど)
 - 1つのプログラムで複数のグラフを作る場合は変数の管理がしやすい。

コマンド的にプロット

```
plt.plot(x, y)
plt.savefig("test.png")
```

オブジェクトを作ってからプロット

```
fig = plt.figure() # Figure オブジェクトを作成
ax = fig.add_subplot(111) # プロットする領域を作成(※2)
ax.plot(x, y) # プロット
fig.savefig("test.png") # 保存
# 表示する場合は plt.show() (fig.show()ではない)
```

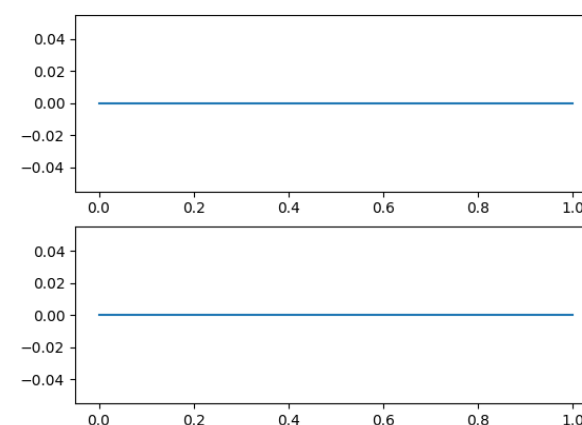
※1 コマンド的にプロットした場合、plt.show() した後はグラフがクリアされるので注意

※2 「111」の意味は「Figure を 1 行 1 列に区切ったときの 1 番目の要素に追加する」である。

2段組グラフを作るときは、以下のように書く。

```
ax1 = fig.add_subplot(211) # 1 段目
ax2 = fig.add_subplot(212) # 2 段目
```

今回は右のオブジェクトを作ってからプロットしていく方法で解説する。



2段組みグラフ

グラフの体裁, 見た目関係

`ax.plot(x, y, ...)` にオプションパラメータを指定することで見た目を変更する。

```
ax.plot(x, y, ".", lw=1.0, c="r", label="label1", ms=1.0)
```

第三引数 (`"."`) : プロット点の形 (他には `"-"`, `"o"`, `"--"`, など)

`lw` (または `linewidth`) : 線の幅

`c` (または `color`) : 色 (指定可能な色一覧 : <https://pythondatascience.plavox.info/wp-content/uploads/2016/06/colorpalette.png>)

`label` : 凡例の文字列

`ms` (または `markersize`) : マーカの大きさ

その他の設定

- `ax.set_xlim(1.0, 10.0)` でグラフの x 軸の範囲を指定 (y 軸は `set_ylim`)
- `ax.set_xscale("log")` で x 軸を対数スケールに (y 軸も同様)
- `ax.legend()` で凡例を表示 (これを書かないと `label=` で指定した文字が表示されない)
- `ax.grid()` でグリッド線を表示

他にもたくさん設定できます。こだわりたい人はぜひ調べてみてください。

numpy を使ったフィッティング

numpy.polyfit

n 次関数によるフィッティングができる。(Excel で言うところの「近似曲線の追加」)

```
par = np.polyfit(x, y, 1) # 1 次関数
```

```
par = np.polyfit(x, y, n) # n 次関数
```

par に近似曲線のパラメータが list で入る。

1 次関数の場合「 $y = \text{par}[0] * x + \text{par}[1]$ 」ということになる。

参考：https://qiita.com/wrblue_mica34/items/0f26612207b78b68c4a1

scipy.optimize.curve_fit を使ったフィッティング

spicy.optimize.curve_fit

ユーザー定義の任意関数によるフィッティングができる。

```
par, cov = curve_fit(func, x, y)
```

参考：<https://qiita.com/hik0107/items/9bdc236600635a0e61e8>

func はユーザー定義の関数で、事前に定義しておく必要がある。

例：2次関数

```
def func2d(x, a, b, c):  
    return a * x * x + b * x + c
```

例：振幅 A, 周期 T の三角関数

```
def func_sin(x, A, T):  
    return A * np.sin((2 * np.pi / T) * x)
```

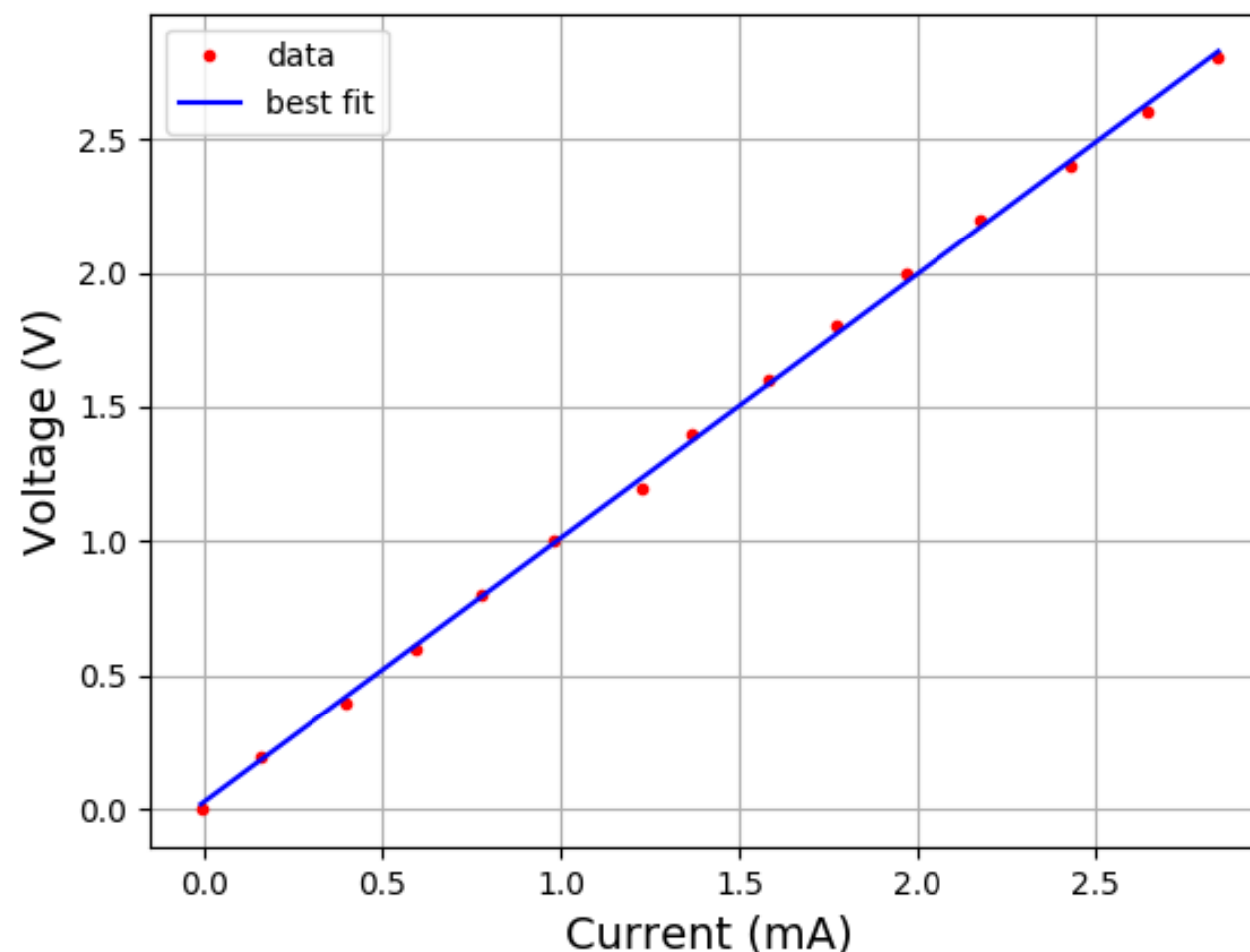
par はフィッティングの最適パラメータが array として入る。

cov は 2Darray の分散共分散行列。フィッティングの誤差を知りたいときに使う。

課題①：電圧と電流のデータからインピーダンスを推定

データ：https://github.com/Yuya-Kinoshita/master/blob/master/python_sample/1912/data/iv_data.csv

1. ファイルを読み込んでください。
2. 電圧を y 軸，電流を x 軸として，1 次関数でフィッティングして傾きを求めてください。
傾きが抵抗値となります。(1 k Ω)
3. 下図のように，データとベストフィットを同時プロットしたグラフを作ってください。
(完全に同じでなくて良いです。)



課題②：ローパスフィルタのカットオフ周波数の推定

データ：https://github.com/Yuya-Kinoshita/master/blob/master/python_sample/1912/data/low_pass_filter_data.csv

データはローパスフィルタ回路の周波数応答を測定したファイルです。

(アナログ電気回路の研修で実際に取得したデータを使っています。)

1. ファイルを読み込んで周波数ごとの減衰比 (出力 / 入力) を計算して、デシベルに変換してください。

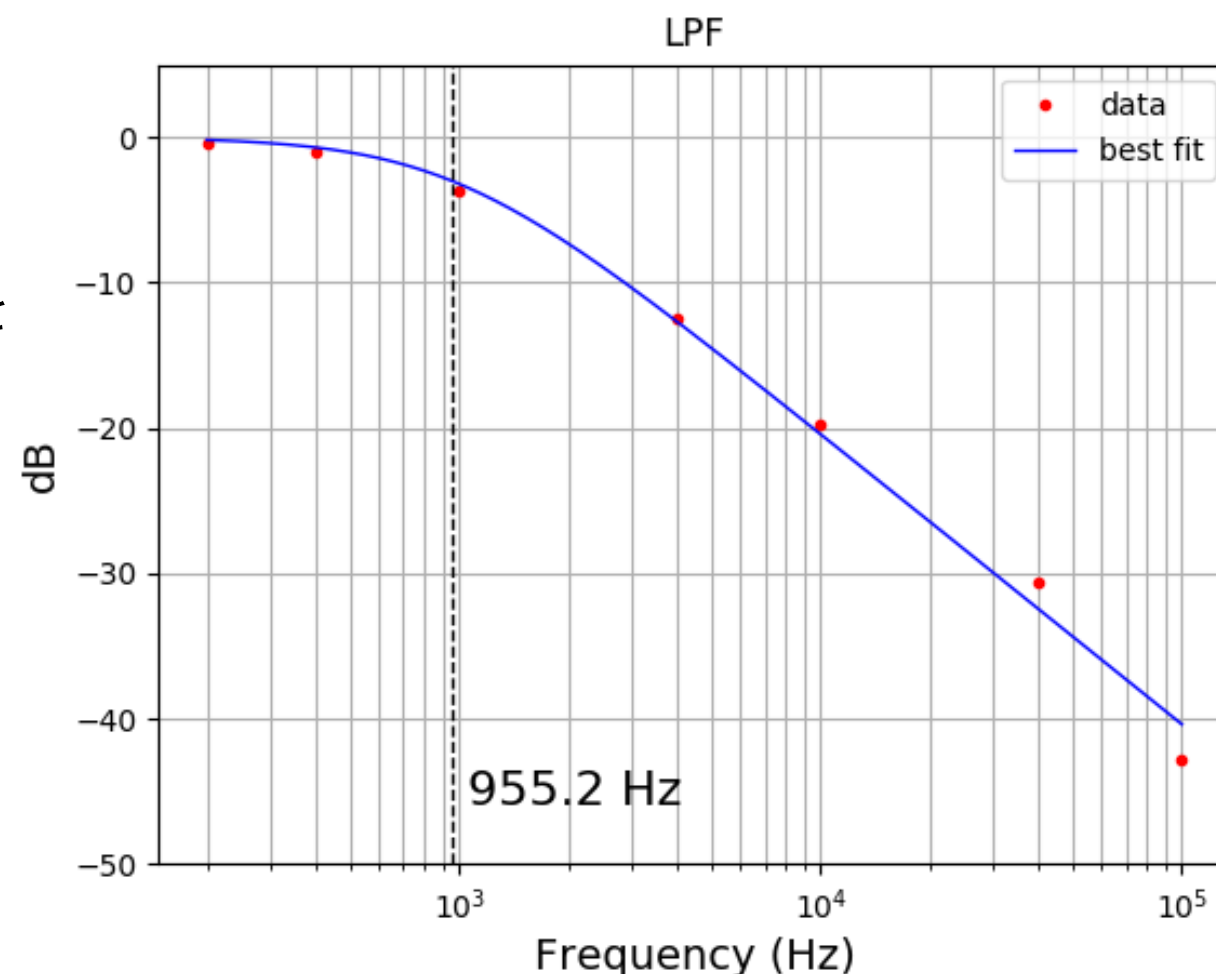
※ $\text{dB} = \log_{10}(\text{出力} / \text{入力})$

2. `scipy.optimize.curve_fit` を使って、データからローパスフィルタのカットオフ周波数を計算してください。

$R = 1.5 \text{ k}\Omega$ のときの電気容量 C を計算してください。

※ 測定では $C = 0.1 \text{ }\mu\text{F}$ を使いました。

3. 右の図のように、データとベストフィットを同時プロットしたグラフを作ってください。
(完全に同じでなくて良いです。)



課題② 補足：ローパスフィルタの関数

$$Z_0 = \frac{1}{\sqrt{1 + (2\pi fCR)^2}}$$
$$= \frac{1}{\sqrt{1 + \left(\frac{f}{f_{co}}\right)^2}} \quad f_{co}: \text{カットオフ周波数}$$

```
def low_pass_filter(freq, cut_off):  
    """  
    ローパスフィルタの周波数特性 (電圧)  
    :param freq: 周波数  
    :param cut_off: カットオフ周波数 (1 / CR)  
    :return: 振幅 (dB)  
    """  
    # 周波数での減衰率  
    ratio = 1.0 / np.sqrt(1 + (freq / cut_off) ** 2)  
    # デシベルに変換  
    db = 20.0 * np.log10(ratio)  
    return db
```