

計算物理学B

第5回

データの可視化

藤本 悠輝、野垣 康介
(藤本担当回)

質問等あればメールでも受け付けます:
yuki.fujimoto.phys_at_niigata-u.ac.jp
(_at_ を @ に変えてください)

講義予定

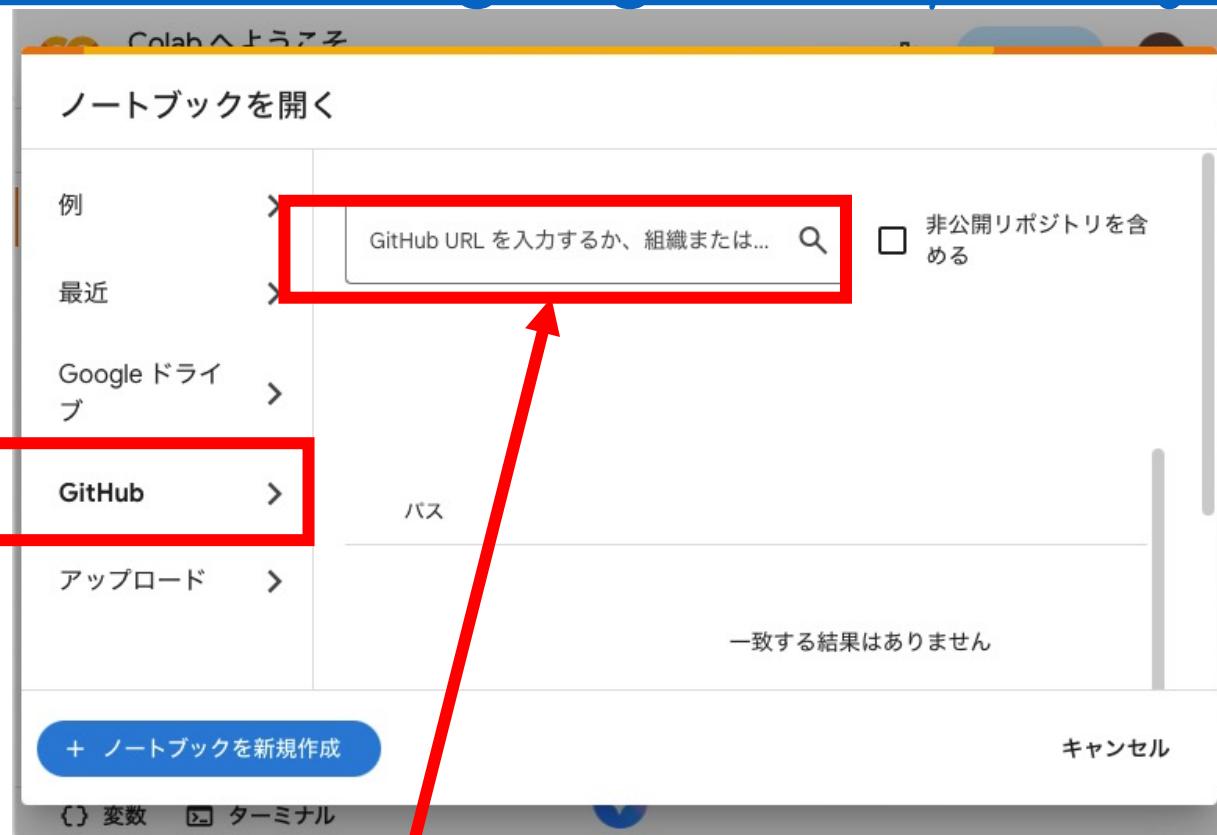
10/07	両名:四則演算	12/09	野垣:モンテカルロ1
10/14	野垣:制御文(for, if)	12/16	野垣:モンテカルロ2
10/21	野垣:関数	12/23	藤本:微分方程式1
10/28	藤本:配列(numpy)	01/13	藤本:微分方程式2
11/04	藤本:可視化(matplotlib)	01/20	藤本:微分方程式3
11/11	野垣:数値微分	01/27	野垣:最適化
11/18	藤本:数値積分 ～中間レポート～	02/03	藤本:機械学習 ～期末レポート～

あくまで予定なので変更の可能性あり

実習環境

まず、Google Colabを開く：

<https://colab.research.google.com/?hl=ja>

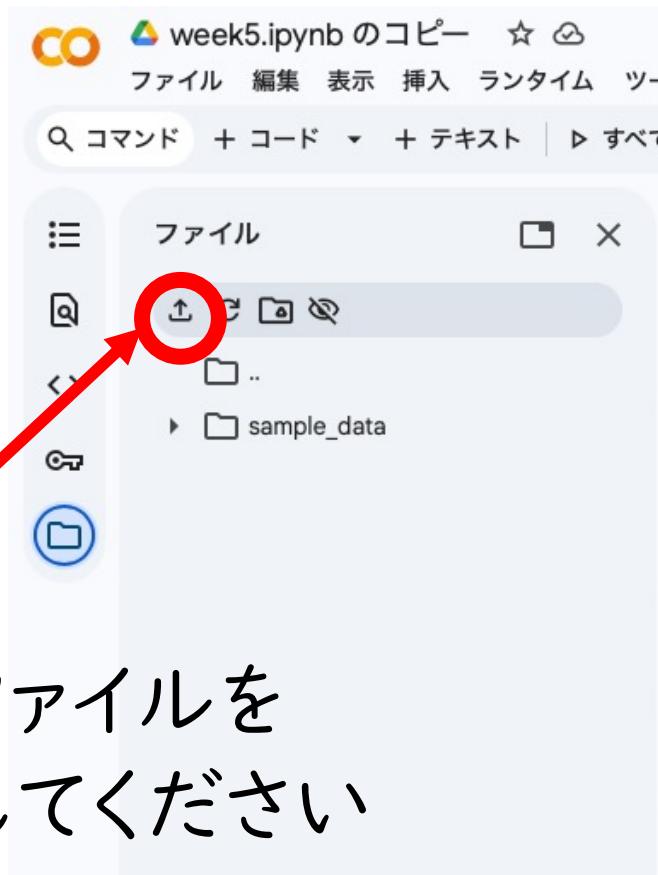


1. GitHubを選択

2. ここに今週のNotebookのURLを入力してEnter:

https://github.com/nogaki/Computational_Physics_B/blob/main/week5/week5.ipynb

今週のノートについての注意点



A screenshot of a Jupyter Notebook interface. At the top, there's a toolbar with a 'CO' icon, a title 'week5.ipynb のコピー', and various menu options: ファイル, 編集, 表示, 挿入, ランタイム, ツール, ヘルプ. Below the toolbar is a search bar with 'コマンド' and a code editor with a search bar. The main area shows a file browser with a red circle and arrow pointing to the 'upload' icon. The file browser also shows a folder named 'sample_data'. To the right, there are two code cells. The first cell is titled 'Pythonでの可視化の基本 (mat' and contains a snippet of Python code for importing matplotlib, numpy, and pandas. The second cell contains code for setting default styles with parameters like 'legend.fontsize': 'large'.

Pythonでの可視化の基本 (mat

```
# 色々なライブラリのインポート
import matplotlib.pyplot as plt # mat
import numpy as np # numpy を np として
import pandas as pd # pandas を pd とし
from mpl_toolkits.mplot3d import Axes
from matplotlib import animation # あ
from IPython.display import HTML # あ
```

```
# デフォルトのスタイルを指定するパラメーターの
# 辞書型 {'key': 'value'} で指定される。
params = {'legend.fontsize': 'large',
          'axes.labelsize': 'x-large',
          'axes.titlesize': 'x-large',
          'xtick.labelsize': 'x-large',
          'ytick.labelsize': 'x-large'}
```

ここからcsvファイルを
アップロードしてください

https://github.com/nogaki/Computational_Physics_B/blob/main/week5/pulsar.csv

前回への補足

PythonのリストあるいはNumPy配列についてfor文を適用する場合は、

```
a = np.linspace(0,1)  
for i in a:  
    print(i)
```

変数

Tabでインデントして各要素についての処理

ここに繰り返し処理したい配列を持ってくる
[0., 0.020, 0.041, 0.061, 0.082, ...]

第5回 matplotlibによるデータの可視化

- matplotlib
- 折れ線グラフ
- 散布図
- 等高線・密度・3次元プロット

この講義は以下を参考に準備されています:

「実践計算物理学」野本拓也、是常隆、有田亮太郎 著(共立出版)

<https://github.com/vlvoch/PHYS6350-ComputationalPhysics>

matplotlib

Pythonの計算結果は、matplotlibというライブラリを使ってプロットされることが多い。
通常、pltとしてimportする：

```
import matplotlib.pyplot as plt
```

データの可視化

プログラムは数値データを出力する

吐き出された数値を見ただけでは、
データの性質を理解することは難しい

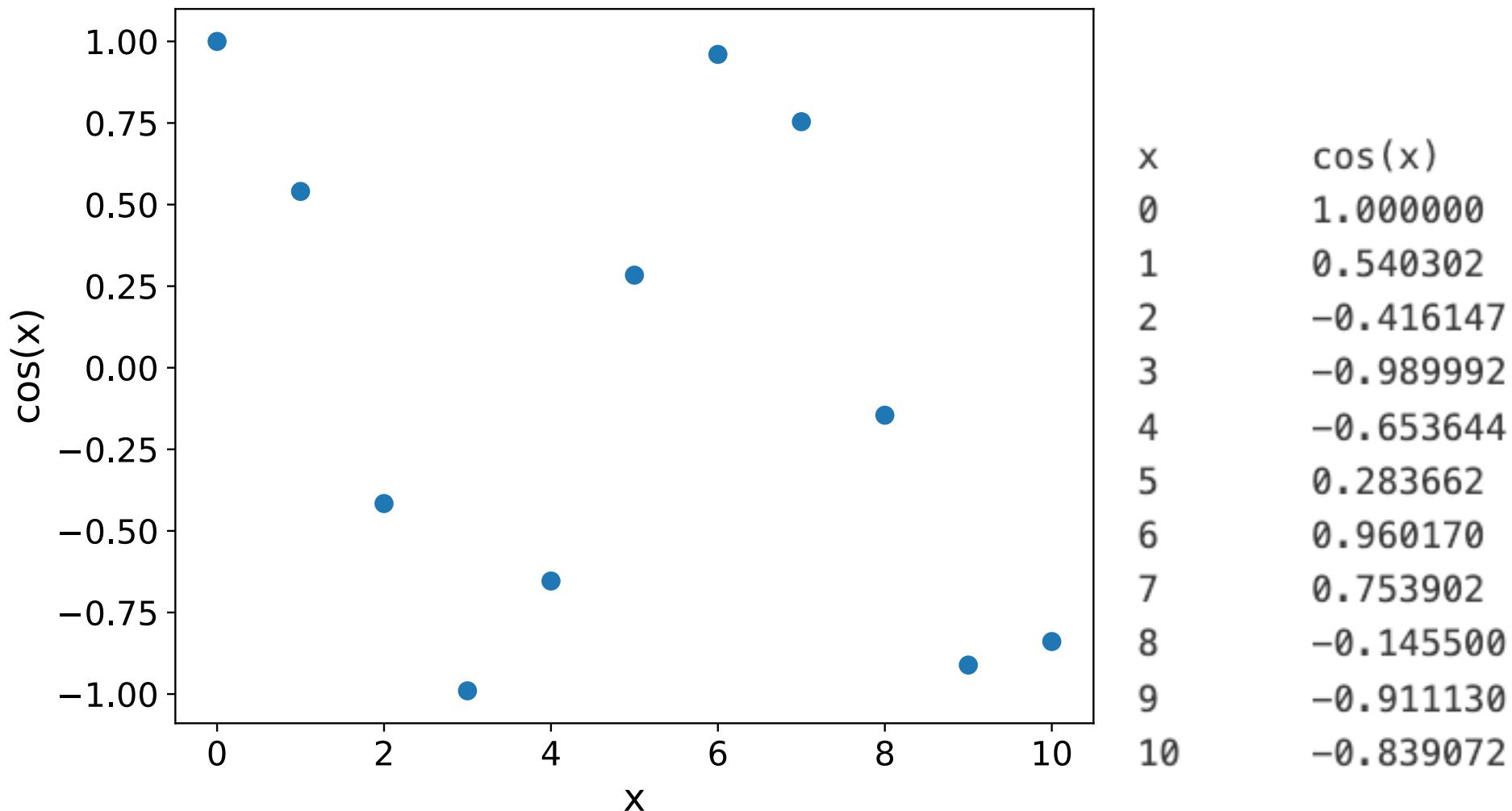
たとえば、以下の関数を考えよう：

$$y = \cos(x)$$

この関数を、 $x=0$ から 10 までの 10 個の等間隔の点で見積もると右のようになる

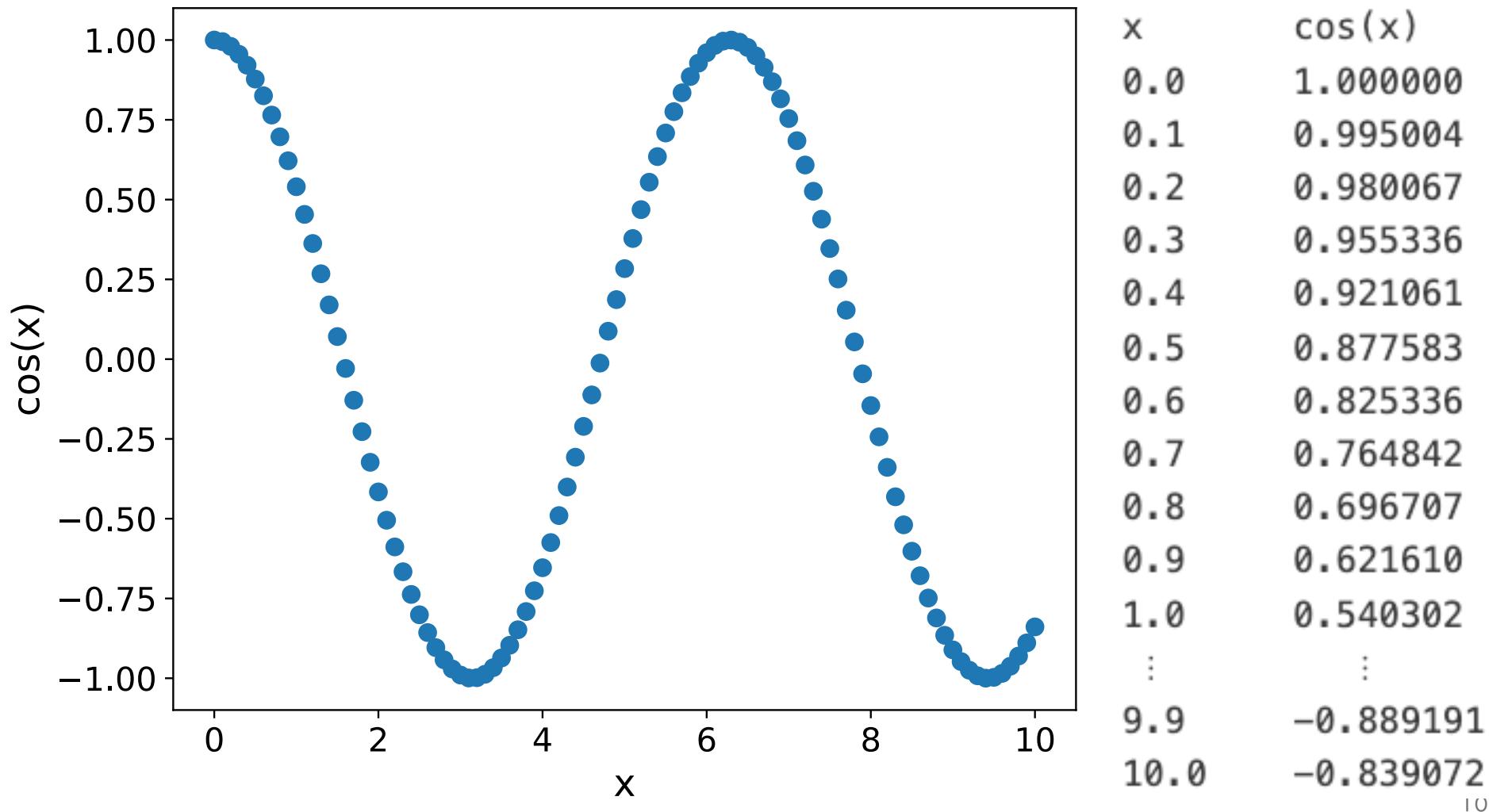
x	$\cos(x)$
0	1.000000
1	0.540302
2	-0.416147
3	-0.989992
4	-0.653644
5	0.283662
6	0.960170
7	0.753902
8	-0.145500
9	-0.911130
10	-0.839072

グラフとして可視化する



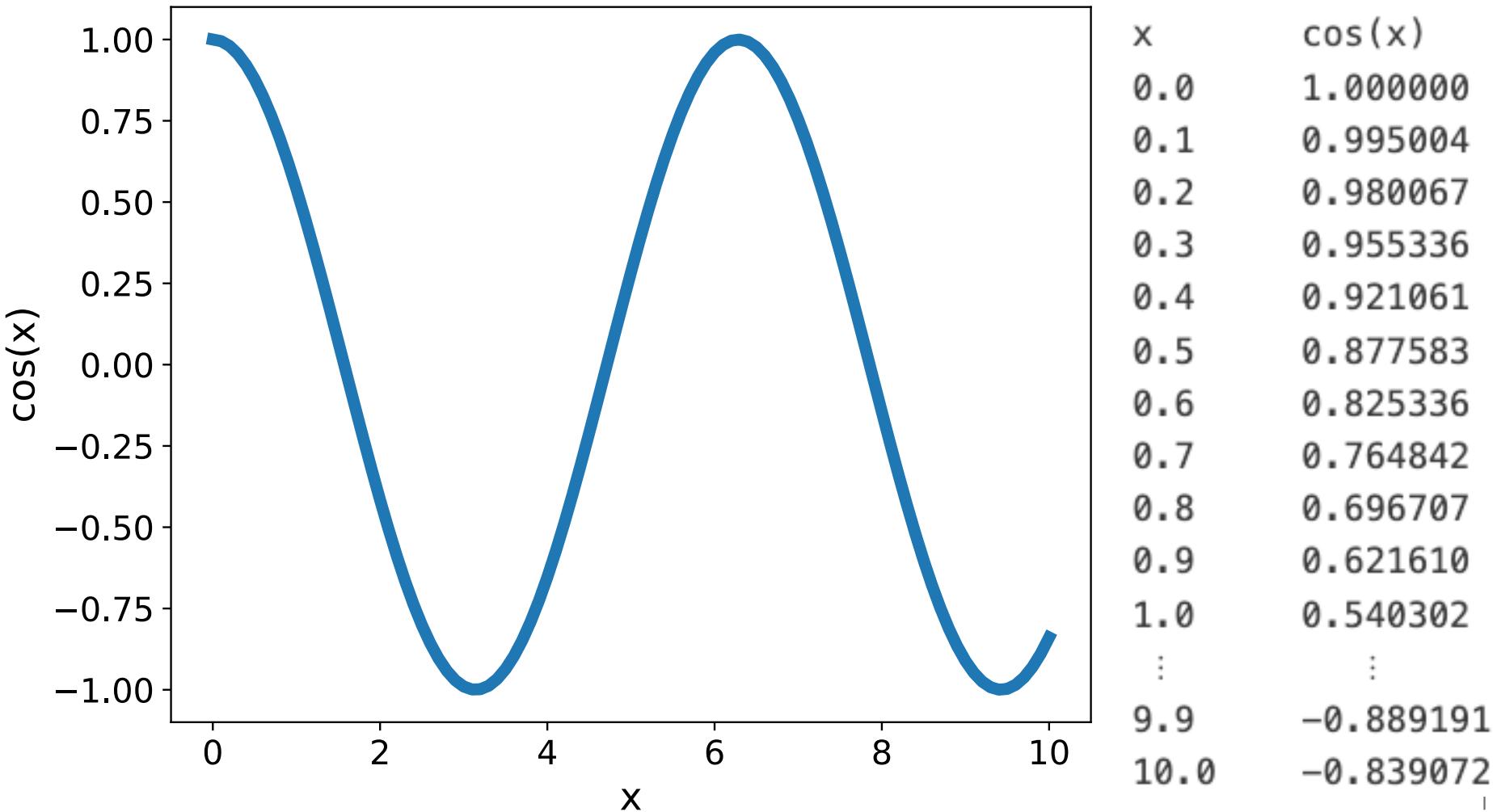
グラフとして可視化する

点の数を増やしてみると…

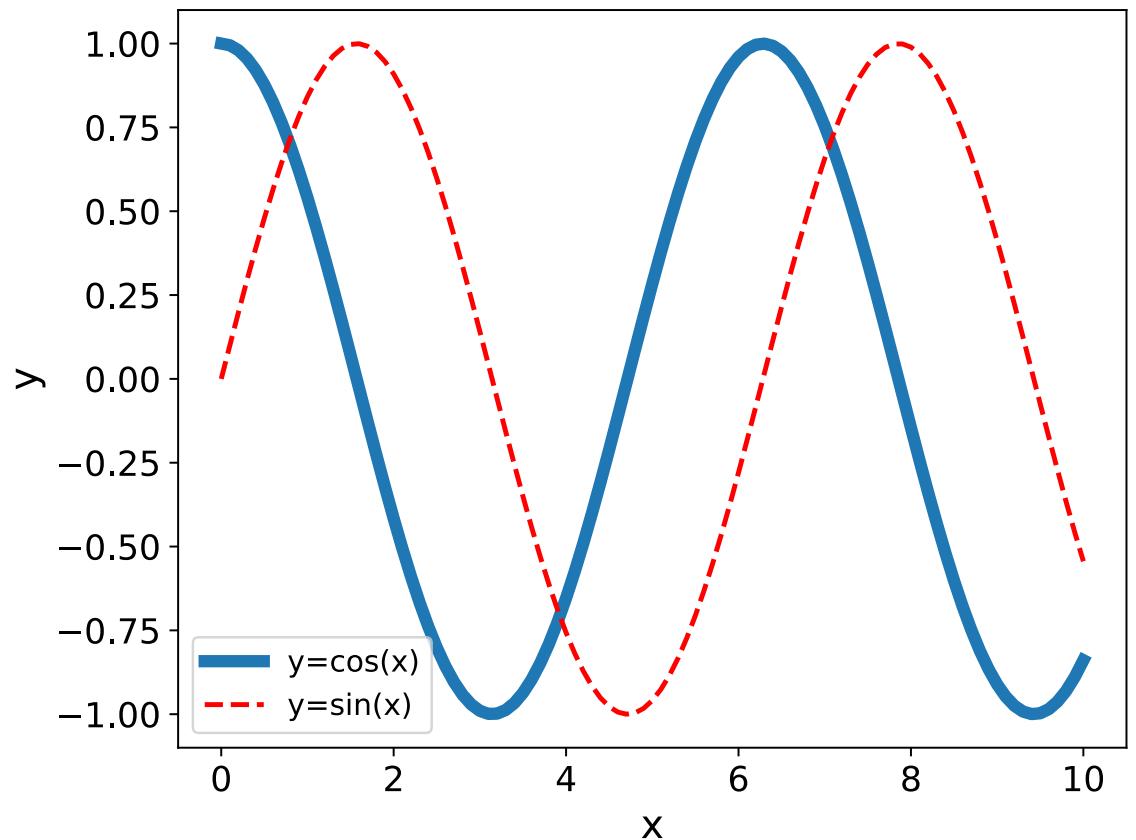


折れ線グラフ

点の数が十分あるので線分で繋いでも良い。
データの連続的な変化を可視化(折れ線グラフ)



複数のデータがある場合



- 凡例を追加する
- データを区別できるように線種・太さ・色などを変更

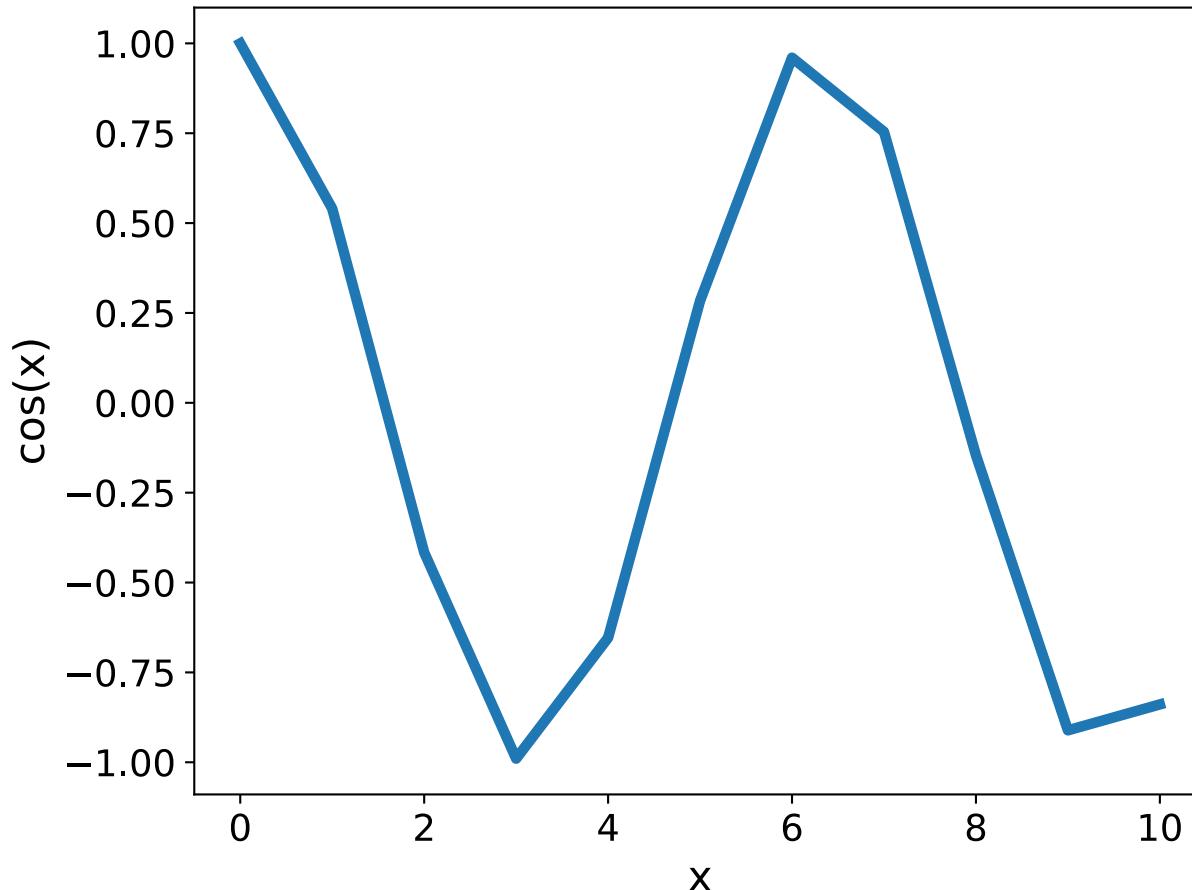
色弱の人がいるので、できれば色だけで区別することは避けるのが好ましい。
参考URL:

<https://www.artiencegroup.com/ja/column/cud/color-vision-defects-simulation.html>

避けるべきこと

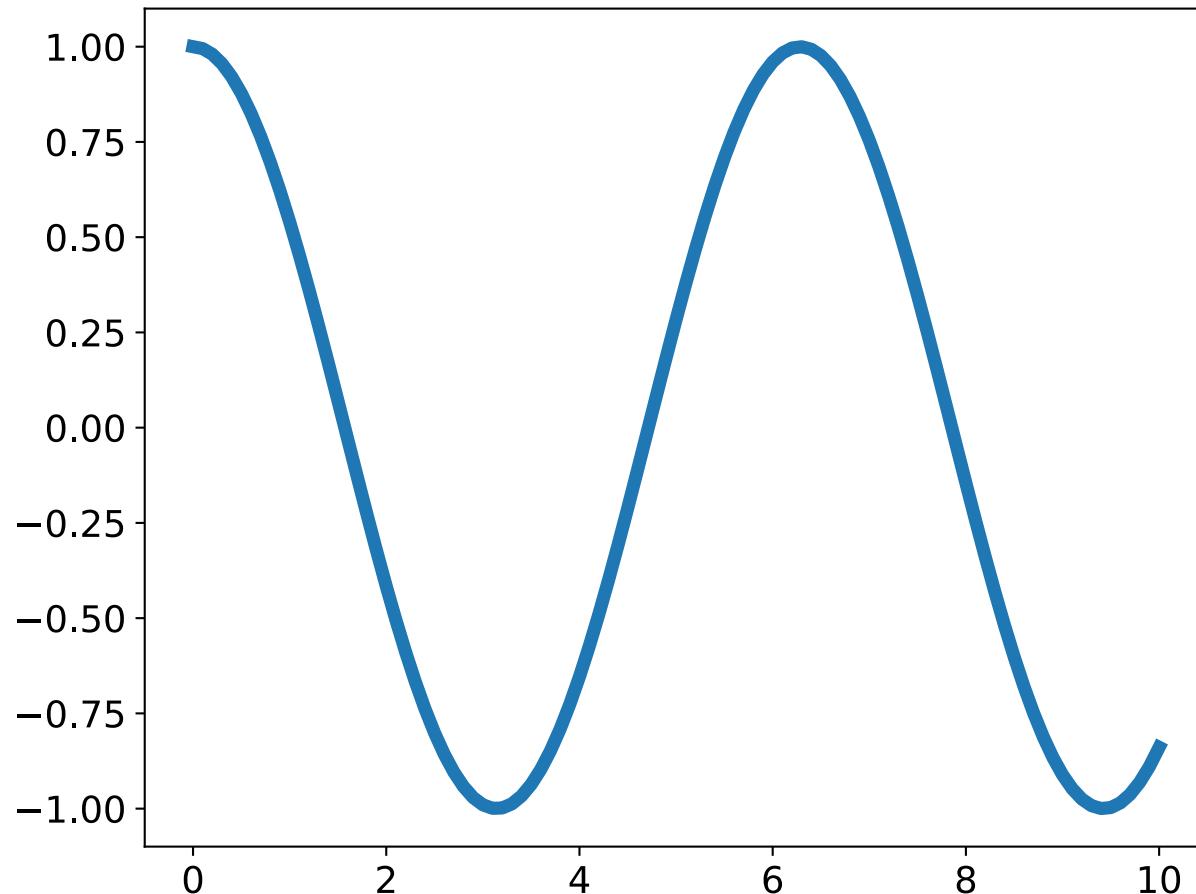
点の数が少ない

→データの傾向や変化が正確に表現できない



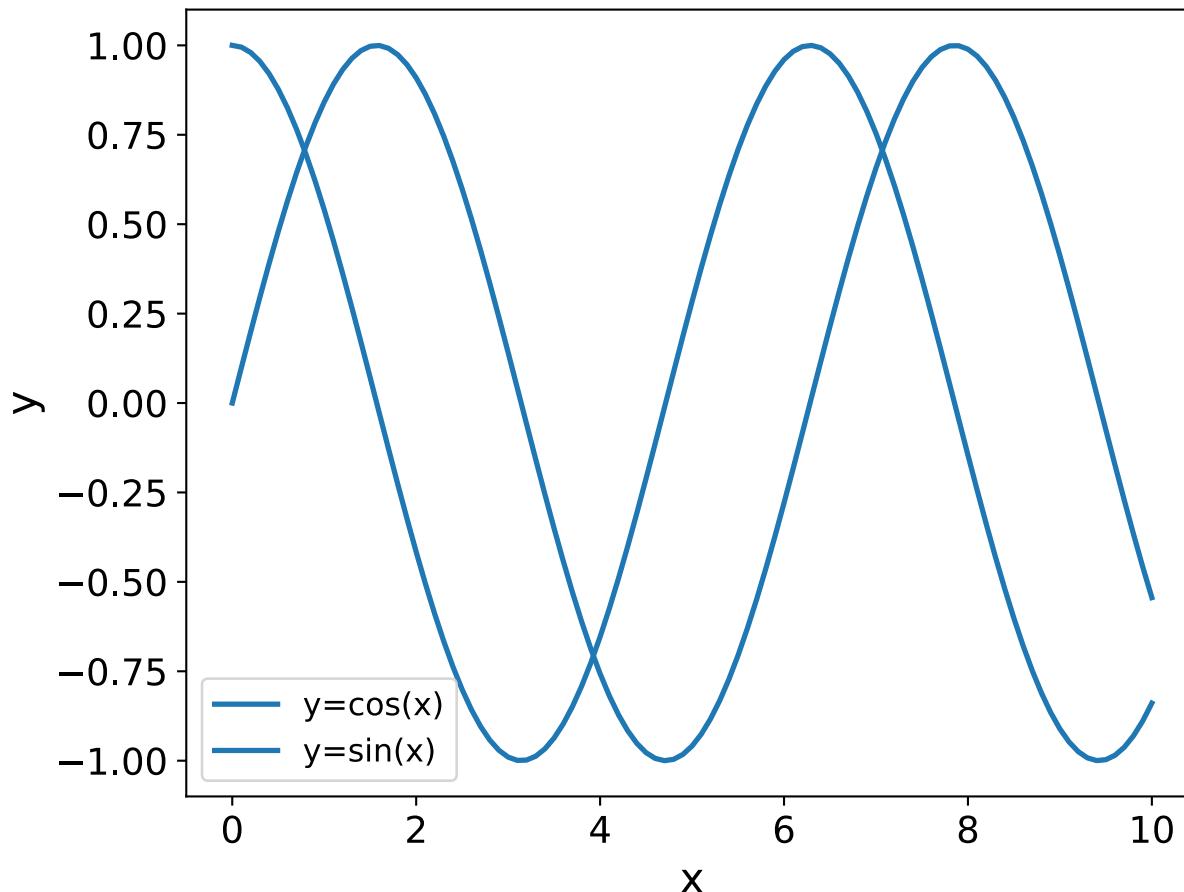
避けるべきこと

x , y 軸にラベルがない、あるいはラベルの文字が小さい
→ 何のデータをプロットしているのかわからない



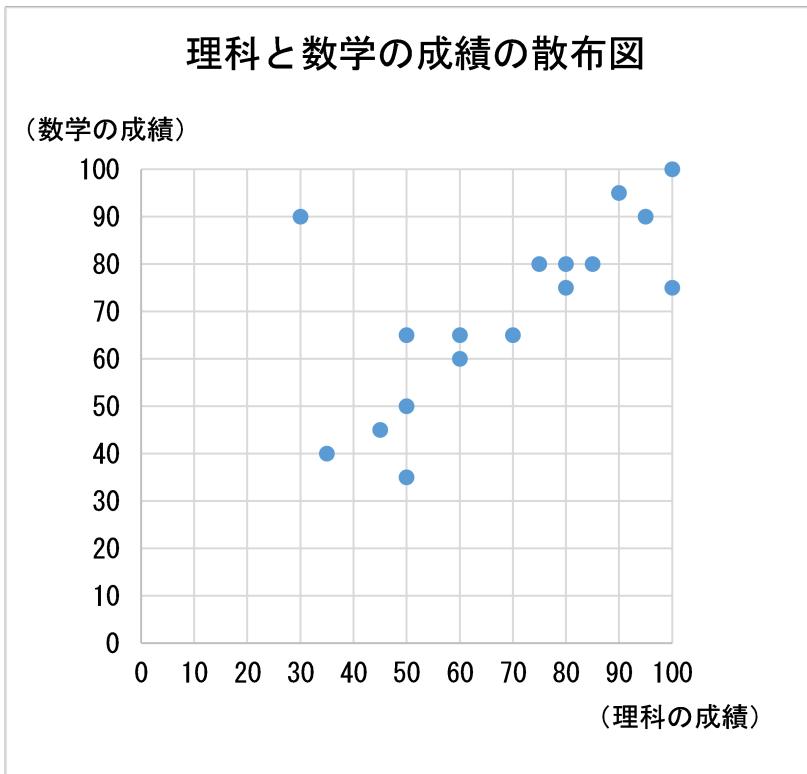
避けるべきこと

同じ色・線種で異なるデータをプロットする
→異なるデータが区別できない



散布図

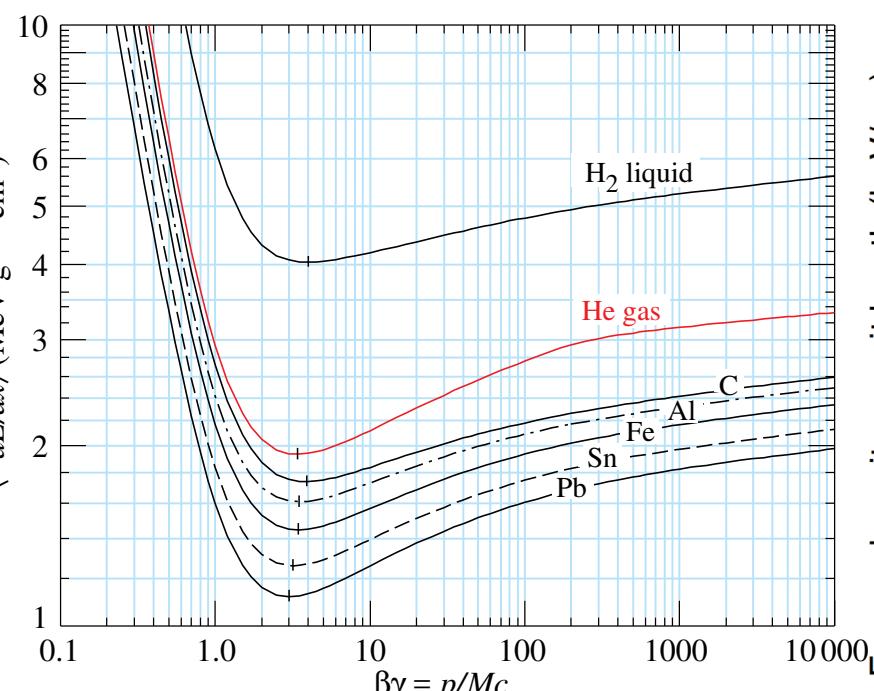
- 全てのデータ点は折れ線で繋ぐのが適切とは限らない
- データの間に相関や構造があるかを調べる際には散布図を用いる



統計局ウェブサイトより:
https://www.stat.go.jp/naruhodo/9_graph/jyokyu/sanpu.html

散布図

- たとえば、加速器実験での粒子の識別に使われる
- 理論

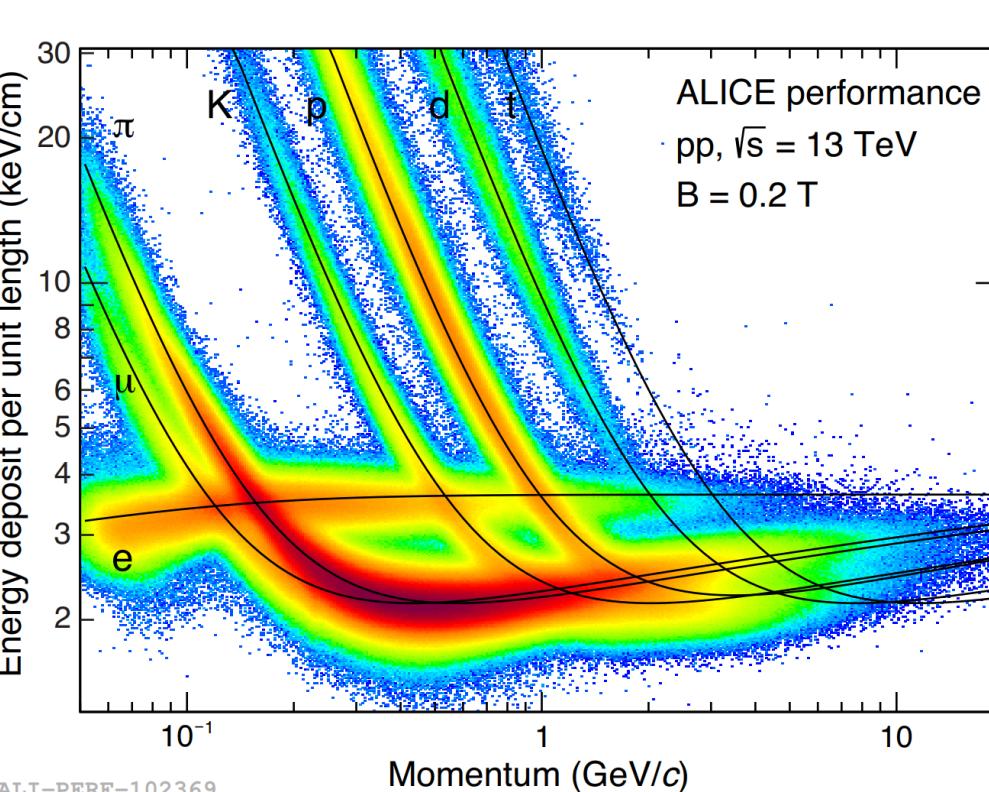


$\langle -dE/dx \rangle$ (MeV g $^{-1}$ cm 2)

$\beta\gamma = p/Mc$

taken from PDG

実験の散布図



Energy deposit per unit length (keV/cm)

Momentum (GeV/c)

ALICE performance
pp, $\sqrt{s} = 13$ TeV
 $B = 0.2$ T

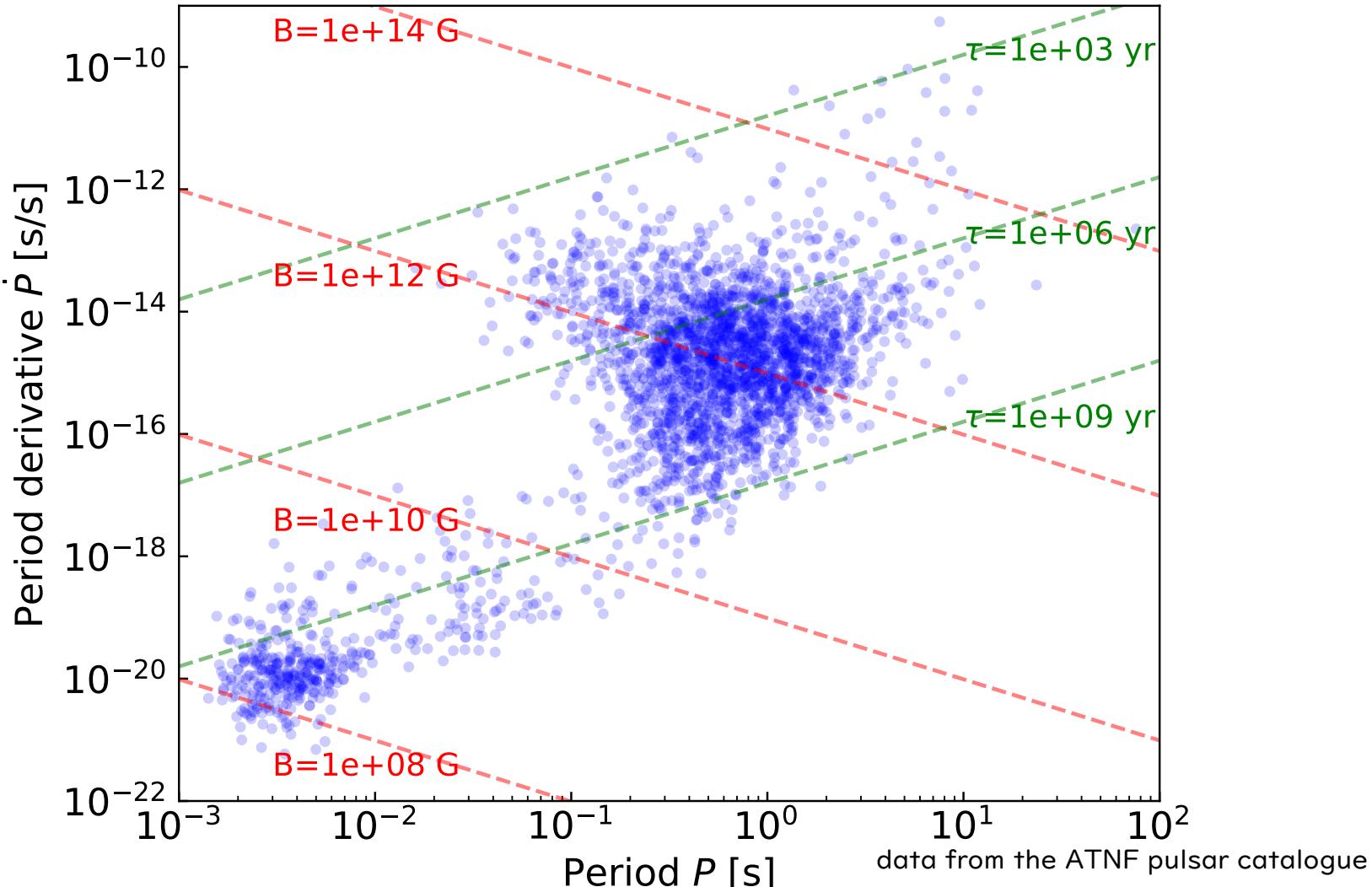
ALI-PERF-102369

いわゆるBethe(-Bloch) formula

taken from: <https://cds.cern.ch/record/2047855>

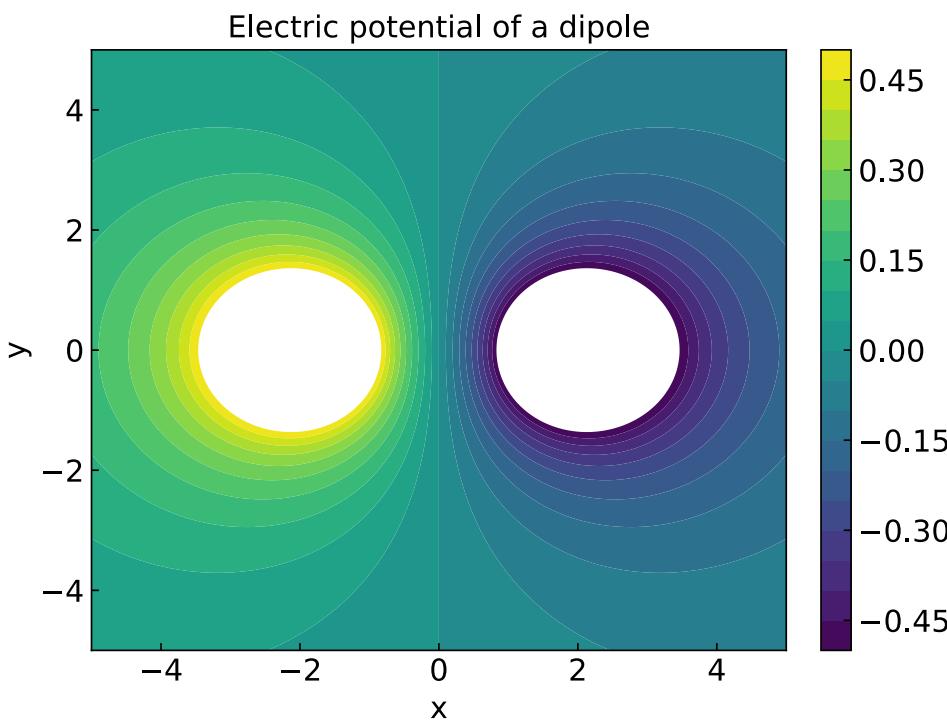
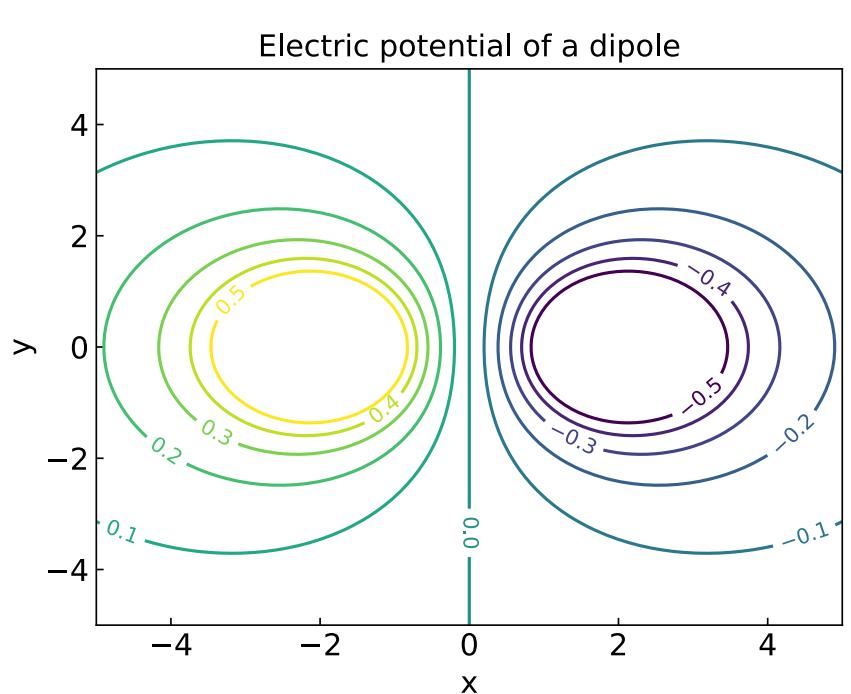
散布図

- 回転する中性子星の周期とその時間微分



等高線図 (contour plot)

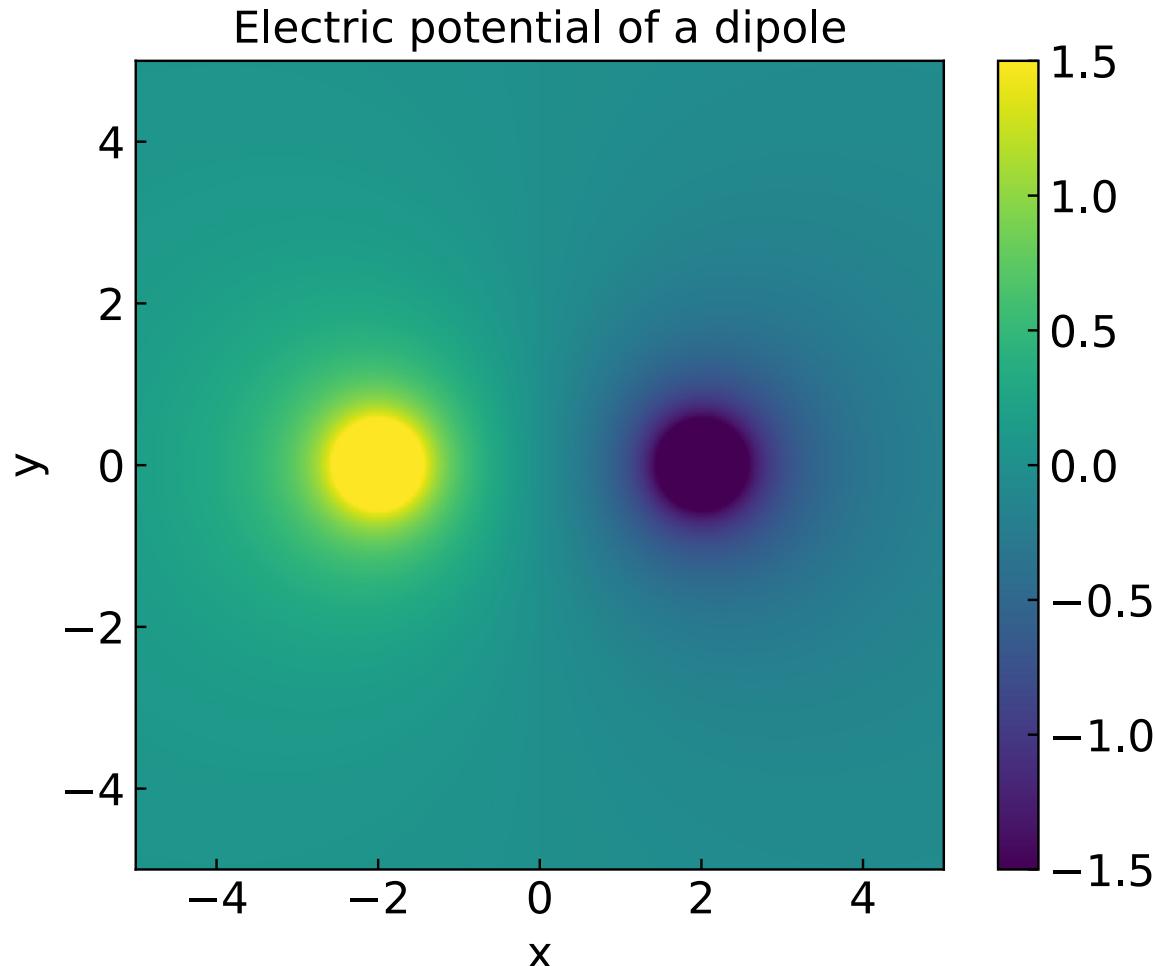
- 場の配位など、多変数関数のプロットに用いる。
たとえば、電気双極子の作るポテンシャルとか



密度プロット (density plot)

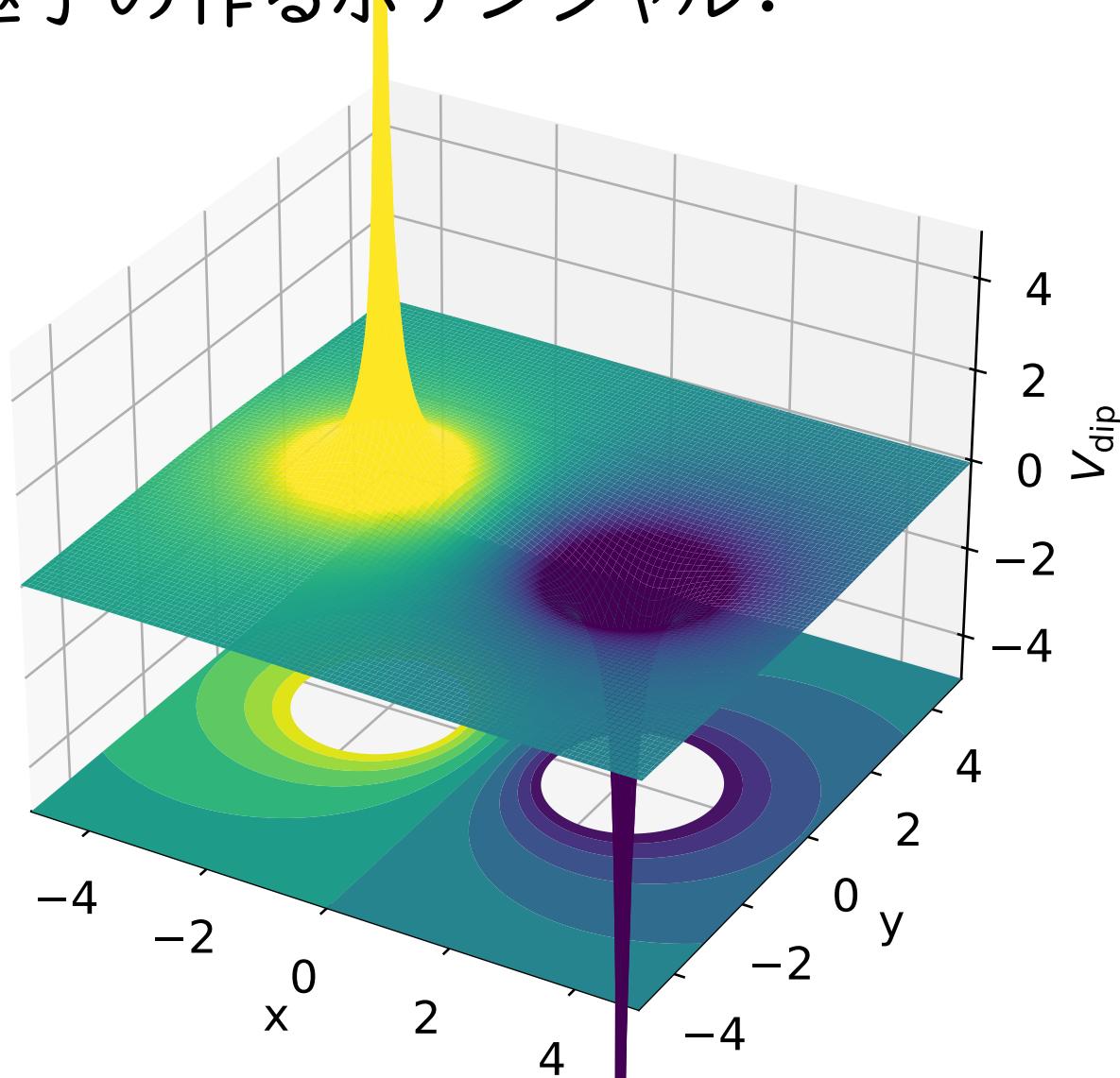
- 確率分布や場の配位など。

電気双極子の作るポテンシャル：



3次元プロット

電気双極子の作るポテンシャル:



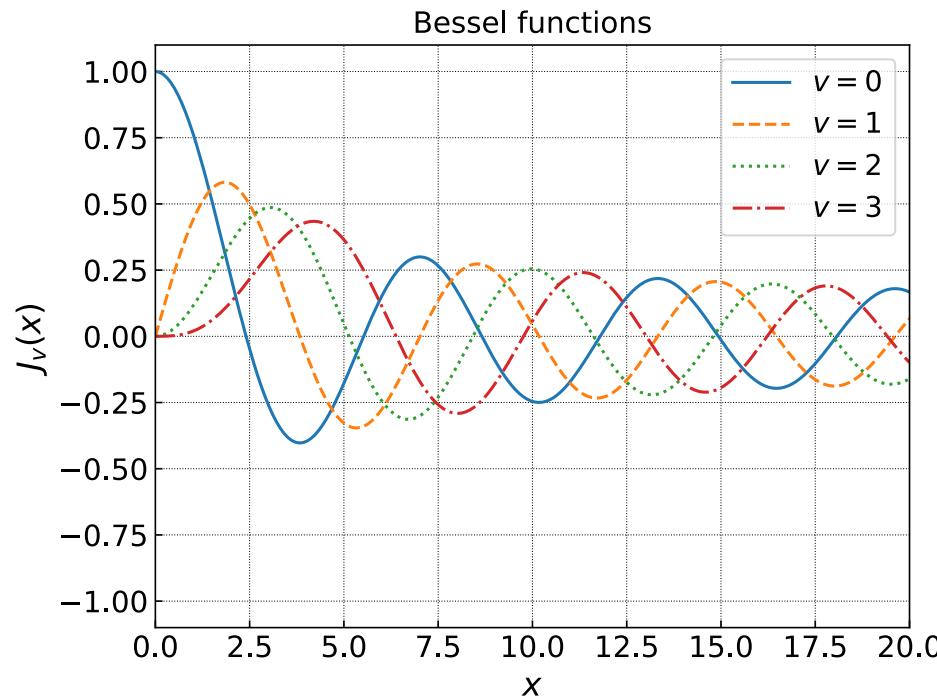
実習タイム

- 第一種Bessel関数 $J_v(x)$ を $v=0, 1, 2, 3$ について $x=0$ から 20 までプロットせよ。
ただし、Bessel関数はscipyからimportせよ

```
from scipy.special import jv
```

線種やラベルについては、右図のようにせよ：

プロットした結果をpdfファイルに保存して提出



発展課題

- 以下のフラクタル図形のうち、どれかを調べてプロットしてみよ：
 - マンデルブロ集合
 - コッホ曲線
 - バーンズリーのシダ
 - シェルピンスキーのギャスケット
 - ジュリア集合
 - ...
- さらに時間が余ってしかたなければ、アニメーションにしてみよう

