

計算物理学 II 第 4 回レポート課題

提出期限：2026 年 1 月 9 日

以下の課題 1, 2, 3, 4 に取り組み、その結果を L^AT_EX でレポートにまとめよ。なお、以下の点に留意せよ。

- レポートにはタイトルを付け、氏名、学籍番号、所属、レポート作成日を記載すること。
- レポート作成時に、この.pdf ファイルのソースファイル (lecture10/report4/main.tex) を活用しても構わない。
- 読みやすく、体裁の整ったレポート作成を心がけて欲しい。
- 作成したレポートの.pdf ファイルと**課題 1, 課題 2, 課題 4**で使用したソースコード (.py ファイル) を manaba に提出すること。

1

以下の行列の特異値を全て求めよ。^{*1}

$$(1) \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 3 & 2 & -3 \\ 0 & 1 & 0 & 1 \end{bmatrix} \quad (2) \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \quad (3) \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 \\ 0 & \sqrt{2} & 0 \\ 1 & 0 & -1 \end{bmatrix}$$

(4) 対角成分が 2, 残りの成分は全て 1 で与えられる 2025 次の正方形行列

2

図 1 のように、一边の長さが 1 の正方形の内部に存在する四分円を考える。この正方形に向かってランダムにダーツを投げる。投げたダーツは必ず正方形の内部のどこかに刺さるものとする。ダーツを投げた回数を N とし、投げたダーツが四分円の内部に刺さった回数を $M (< N)$ とする。ダーツの刺さる場所がランダムであれば、 N が十分に大きい場合、 N と M の比は正方形と四分円の面積の比になることが予想される。つまり、

$$\frac{M}{N} \sim \frac{\pi}{4}$$

となることが期待できる。したがって、図 1 を的としてランダムにダーツを投げるだけで、円周率 π を

$$\pi \sim 4 \frac{M}{N} \tag{2.1}$$

^{*1} 実は、(1) から (4) の中には具体的な特異値分解をせずとも答えが分かる行列がある。ぜひ考えてみて下さい。

と推定することができる。式(2.1)をプログラムを書いて確かめてみよう。

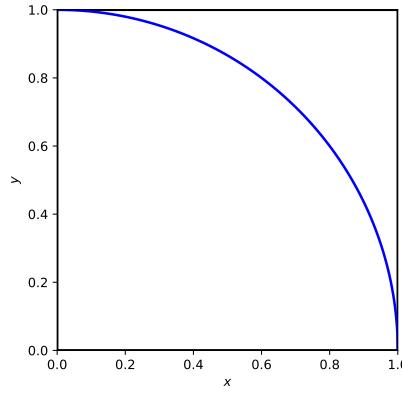


図1 正方形内の四分円

(A) $N = 10^2, 10^3, 10^4$ の三つの場合について、 K 回だけ π の推定値を求め、そのヒストグラムを作成し、レポートに掲載せよ。そして、三つのヒストグラムを比較することで、得られる π の推定値と N の関係について議論せよ。なお、 K は $O(10^3)$ から $O(10^4)$ 程度とし、各自で自由に決定せよ。選んだ N_{sample} の値についてはレポートに明記すること。

(B) $N = 2^k$ ($k = 4, 5, \dots, 19, 20$) における π の推定値を求め、その結果を図にまとめよ。ただし、横軸を N 、縦軸を π の推定値とし、エラーバーをつけること。また、得られた図から分かることを簡潔に述べよ。

3

m 行 n 列の行列 A の低ランク近似は、特異値分解によって以下のように与えられる。

$$A_{ij} = \sum_{k=1}^{\min(m,n)} U_{ik} \sigma_k V_{kj}^\dagger \simeq \sum_{k=1}^{\chi} U_{ik} \sigma_k V_{kj}^\dagger. \quad (3.1)$$

ここで、 σ_k は k 番目の特異値であり、降順 ($\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min(m,n)}$) で並んでいるものとする。特異値の打ち切り次元 χ について、 $\chi = \min(m, n)$ とすれば、式(3.1)の右辺は元の行列 A を厳密に再現するが、特異値の減衰が十分に早ければ $\chi < \min(m, n)$ でも元の行列 A を十分な精度で復元することができる。このことを利用して、モノクロ画像データを圧縮してみよう。手続きは以下の通りである。

1. 画像ファイル（ここでは.jpg ファイルとする）をモノクロ化し、モノクロ化された画像を行列とみなす。
2. この行列に対し、特異値分解による低ランク近似を実行する。
3. 低ランク近似で得られた行列をモノクロ画像として再構成する。

この手続きに従って、lecture10/report4/img の中にある画像を圧縮し、どの程度の χ で元の画像がどのくらい復元されるか Python を使って調べてみよう。ただし、以下の設問では、次のソースコード 1 のように各種ライブラリが import されていることを前提としている。

ソースコード 1 本課題で必要なライブラリの import

```
1 import numpy as np
```

```
2 import matplotlib.pyplot as plt
3 from scipy import linalg
4 from PIL import Image
```

(A) ソースコード 2 は `filename` という名前の画像ファイルをモノクロ化し、モノクロ化された画像を行列へと変換するための関数である。この関数を使って `image0.jpg` を行列へ変換した上で特異値分解を実行し、得られた特異値 σ_i について、横軸を i 、縦軸を σ_i とする図を作成せよ。

ソースコード 2 画像ファイルのモノクロ化とその行列への変換

```
1 def matrix_grayscale(filename):
2     image = Image.open(filename)
3     grayscale_image = image.convert("L")
4     return np.array(grayscale_image)
```

(B) 元々の `image0.jpg` に対応する行列および、 $\chi = 5, 25, 50, 100$ で圧縮された行列をモノクロ画像として再構成するコードを作成し、得られたモノクロ画像をレポートに掲載せよ。^{*2} なお、ソースコード 3 は与えられた行列 `matrix_image` をモノクロ画像として再構成するための関数である（引数の `matrix_image` は NumPy 配列にも対応している）。必要に応じて自由に使用してもらって構わない。

ソースコード 3 行列をモノクロ画像として再構成

```
1 def plot_single_image(matrix_image):
2     plt.figure(figsize=(8, 6))
3     plt.plot()
4     plt.title("Name")
5     plt.imshow(matrix_image, cmap="gray")
6     plt.axis("off")
7     plt.savefig("visualized_image.pdf")
```

(C) `lecture10/report4/img` の中にある `image0.jpg` 以外の画像、^{*3} あるいは自分で好きに用意した.jpg ファイルに対して、上の設問 (A), (B) を行ってみよ。ただし、圧縮を行う際の χ の値は適宜変更しても構わない。

4

この課題では、プログラミングにおける生成 AI の活用方法について考えてみたい。生成 AI は、デバッグ（プログラムに混入したバグを特定・修正する作業）やリファクタリング（出力結果を変えずに、コードの可読性や保守性を向上させる書き換え）において、有用な支援ツールとなり得る。

^{*2} LATEX で複数の画像を見栄えよく掲載するには `minipage` 環境を使うと良い。興味のある人は `minipage` 環境の使い方を調べてみよ。

^{*3} `lecture10/report4/img` の中に格納されている `image5.jpg` は一様乱数を使って生成されたものである。

実際に、生成 AI を使って自身が書いたコードのリファクタリングをしてみよう。ここでは、OpenAI が開発した対話型生成 AI サービスである ChatGPT (<https://chatgpt.com/>) を使ってみよう。URL からアクセスすると、図 2 のような画面になる。テキスト入力ウィンドウから生成 AI への質問を入力し、Enter キーを押すことで生成 AI からの回答が得られる。テキスト入力ウィンドウに図 3 のような入力をし、それに続いて、自身が作成したコードを貼り付けることで、生成 AI にリファクタリングをさせることができる。

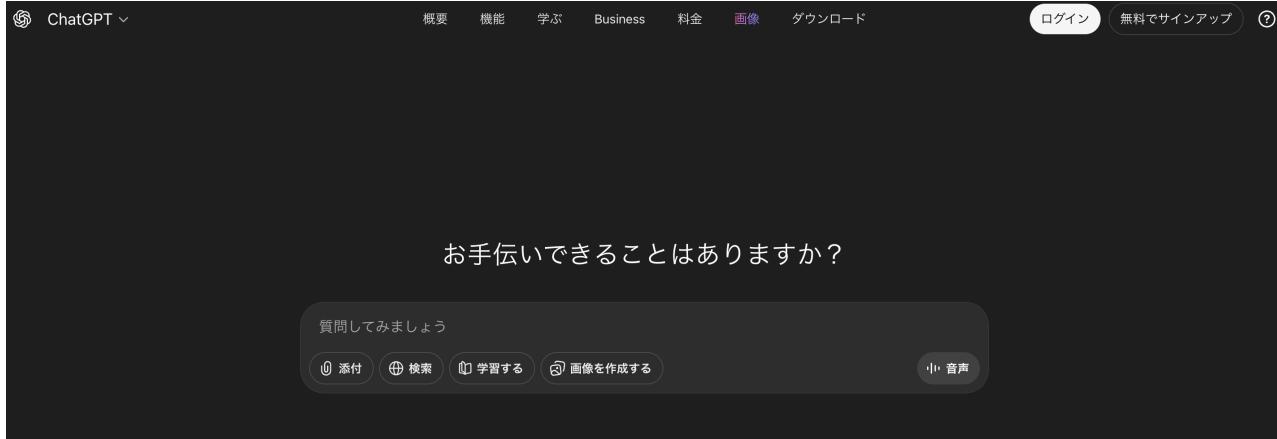


図 2 ChatGPT の画面

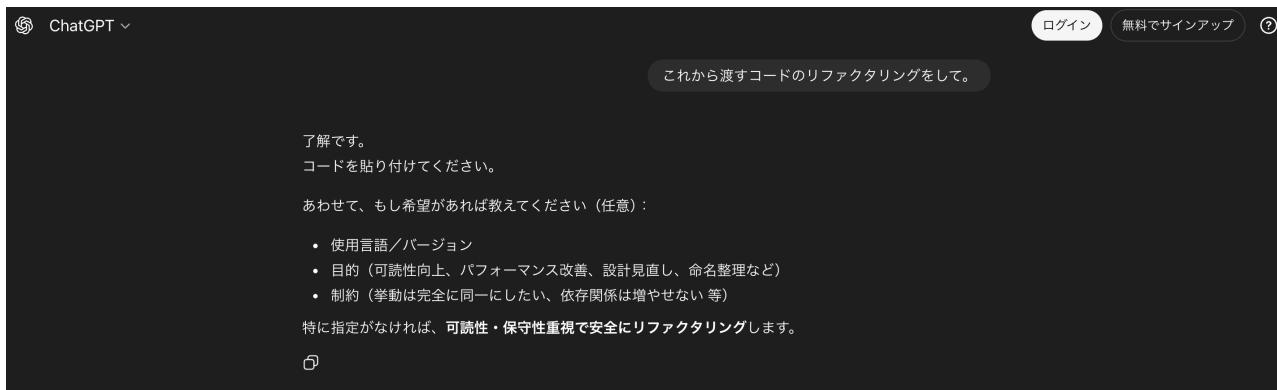


図 3 ChatGPT にリファクタリングをさせる

図 3 のような入力に続いて、第 2 回レポート課題 4 で自身が作成したソースコードを貼り付け、そのコードのリファクタリングを生成 AI によって行ってみてよ。レポートには以下の二点について簡潔にまとめてもらえれば良い。

- 生成 AI が出力したコードが実行できたか。また、そのコードの出力結果は妥当であったか。
- 自身の書いたコードと生成 AI が書いたコードを比較し、どのような差異（改善または改悪）が見られたか。

なお、この課題のソースコードについては、生成 AI によってリファクタリングされたソースコードのみを提出すれば良い。

アンケート

この項目は成績とは無関係です。こちら（クリック）の Google フォームよりアンケートにご協力ください。