

数理工学実験レポート

第 4 章（モンテカルロシミュレーション）

学籍番号 1029366161 中塚一瑛

2025 年 12 月 2 日

概要

目次

1	課題 1：疫病の確率的な伝染モデル	2
1.1	原理と方法	2
1.2	実装上の工夫	2
1.3	小問 1	3
1.4	小問 2	4
2	課題 2：モンテカルロ法による円周率推定	6
2.1	原理と方法	6
2.2	実装上の工夫	7
2.3	結果	7
3	課題 3：熱よく法とメトロポリス法のつり合い条件の証明	8
3.1	1. メトロポリス法が詳細つり合い条件を満たすことの証明	8
3.2	2. 熱浴法が詳細つり合い条件を満たすことの証明	9
3.3	3. 詳細つり合い条件 (5.19) からつり合い条件 (5.17) が従うことの証明	10
4	結論	10
付録 A	使用コード一覧	10

はじめに

今回は

1 課題 1：疫病の確率的な伝染モデル

1.1 原理と方法

本課題では、配布資料に記載の一次元確率的伝染モデルに従い、格子点 $i = 1, \dots, 64$ に状態変数 $s_i(t) \in \{0, 1\}$ を配置した。左右近傍の感染者数を

$$n_i(t) = s_{i-1}(t) + s_{i+1}(t)$$

とし、次時刻の状態は資料式 (5.2) に従う確率 $P(1 \mid s_i(t), n_i(t))$ に基づいて決定する。周期境界条件は

$$s_0(t) = s_{64}(t), \quad s_{65}(t) = s_1(t)$$

により与えた。

モンテカルロシミュレーションでは、時刻 t の状態列 $\{s_i(t)\}$ を初期状態 $t = 0$ から順に確率的に生成する。各格子点で一様乱数 $r \in [0, 1]$ を生成し、

$$s_i(t+1) = \begin{cases} 1 & (r \leq P(1 \mid s_i(t), n_i(t))) \\ 0 & (r > P(1 \mid s_i(t), n_i(t))) \end{cases}$$

により次状態を決定した。

時刻 t の病人数は

$$m(t) = \sum_{i=1}^{64} s_i(t)$$

で定義し、独立な M 個のサンプルに対してサンプル平均

$$\overline{m(t)} = \frac{1}{M} \sum_{k=1}^M m^{(k)}(t)$$

を計算することで、期待値の近似を得た。

本課題では、初期条件として中央のみ感染 ($s_{32}(0) = 1$) とし、シミュレーションの最終時刻は $T = 256$ 、サンプル数は M とした。

1.2 実装上の工夫

Python と NumPy を用いて実装を行った。Python で各格子点を for ループで逐次更新すると計算が極端に遅くなるため、本研究では全サイトをベクトル化により一括更新する方法を採用した。

まず、 M 個の試行を同時に扱うために、状態を

$$s(t) \in \{0, 1\}^{M \times 64}$$

という 2 次元配列として保持した。周期境界条件に基づく左右近傍は NumPy の `roll` により

$$\text{right} = \text{roll}(s, -1), \quad \text{left} = \text{roll}(s, 1)$$

として同時に生成し, $n(t) = \text{right} + \text{left}$ を一括して計算した。

資料式 (5.2) の遷移確率 $P(1 | s, n)$ は 6 通りの組 $(s, n) = (0, 0), (0, 1), (0, 2), (1, 0), (1, 1), (1, 2)$ に対応する値をあらかじめ一次元配列

$$\text{vals} = (\alpha_{00}, \alpha_{01}, \alpha_{02}, \alpha_{10}, \alpha_{11}, \alpha_{12})$$

として保持した。各格子点の組 (s_i, n_i) は

$$\text{index} = 3s + n$$

により 0–5 の整数に写像され,

$$\alpha = \text{vals}[\text{index}]$$

として全要素の遷移確率を分岐なしで取得した。

次状態の決定も同様にベクトル化し, 乱数配列 $r \in [0, 1]^{M \times 64}$ を生成して

$$s(t+1) = (r < \alpha)$$

という比較演算により全要素を同時更新した。病人数は `sum(axis=1)` を用いて

$$m^{(k)}(t) = \sum_i s_i^{(k)}(t)$$

を一括計算した。

以上のベクトル化により, 明示的な `for` 文を用いず高速に M 試行 \times 256 ステップのシミュレーションを実行できた。

1.3 小問 1

1.3.1 結果

パラメータ $p = 0.7$, サンプル数 $M = 10$ に対してシミュレーションを実行し, 各サンプルに対する各時刻での $m(t)$ を同時にプロットした (図 1)。

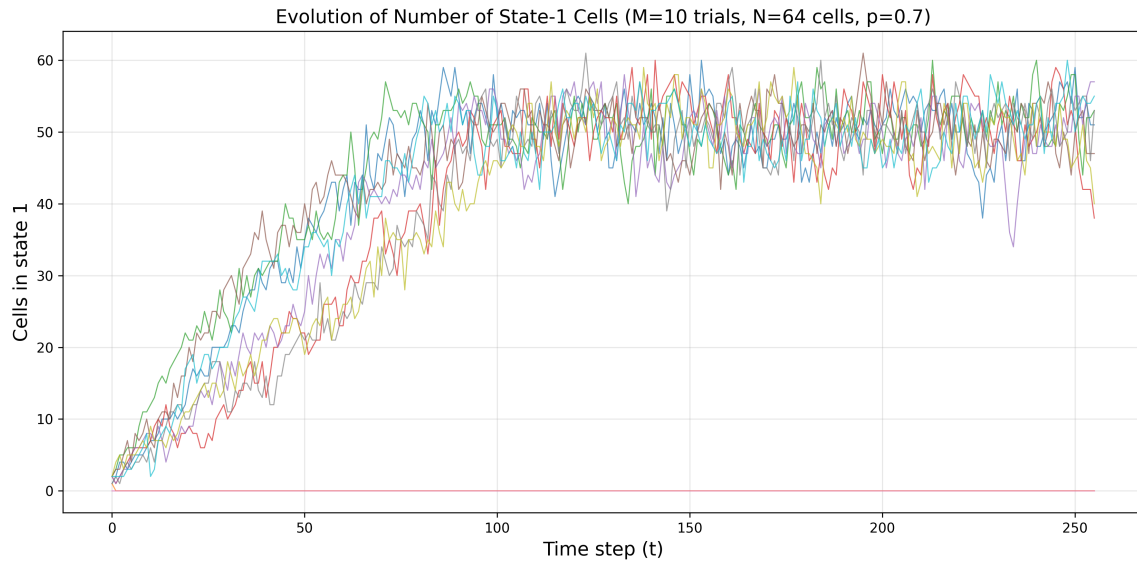


図 1: 各サンプルに対する 各時刻での $m(t)$ ($p = 0.7$, $M = 10$)

1.3.2 考察

図 1 に示すように、各サンプルに対する 各時刻での $m(t)$ は、時間とともに増減を繰り返しながら増加し、最終的には 50 前後を上下するようになっている。これは、感染が広がるにつれて感染者数が増加する一方で、回復によって感染者数が減少するつり合いの結果であると考えられる。また、ごく一部のサンプルでは、開始直後に感染者が 0 となり、そのまま感染が広がらない場合も見られた。これは、感染者が 0 となった時点で、その後の感染拡大が起こり得ないためであると考えられる。

1.4 小問 2

1.4.1 結果

パラメータ $p = 0.64, 0.66, 0.68, 0.7$ 、サンプル数 $M = 10000$ に対してシミュレーションを実行し、各時刻での $m(t)$ のサンプル平均をプロットした (図 2)。

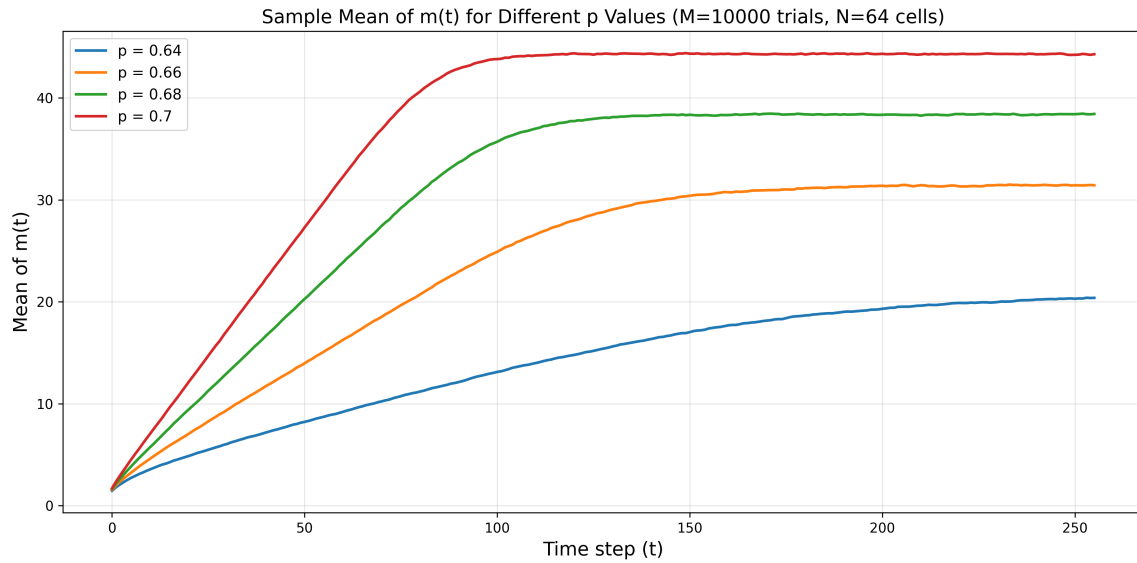


図 2: 各時刻での $m(t)$ のサンプル平均 ($p = 0.64, 0.66, 0.68, 0.7, M = 10000$)

1.4.2 考察

図 2 に示すように、各時刻での $m(t)$ のサンプル平均は、時間とともに増加し、最終的には一定の値に収束する傾向が見られた。また、パラメータ p が大きくなるにつれて、最終的な感染者数の平均値も増加する傾向が見られた。これは、感染確率が高いほど感染が広がりやすくなり、より多くの人々が感染するためであると考えられる。また、問題では最終時刻 256 としているが、 $T=100000$ までシミュレーションを行った結果が以下の図である。

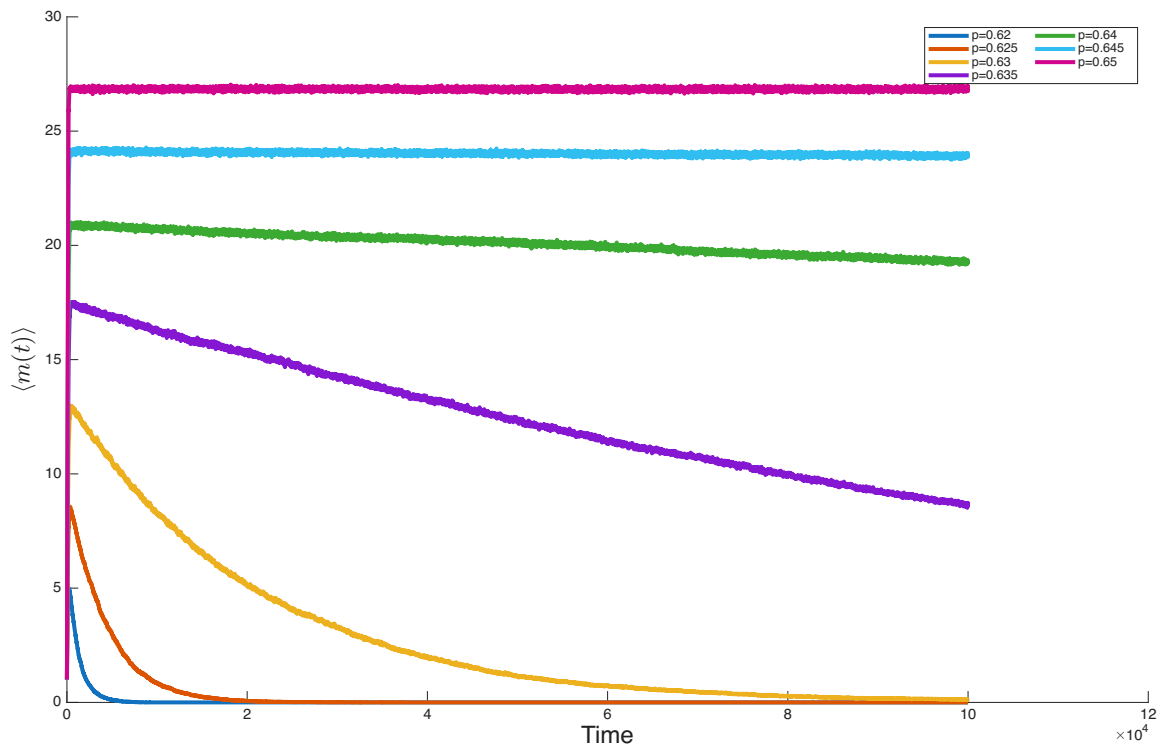


図 3: 各時刻での $m(t)$ のサンプル平均 ($p = 0.64, 0.66, 0.68, 0.7, M = 10000, T = 100000$)

図 3 が示すのは、 $p = 0.65$ では一旦は増加した感染者数が一定の値を保っているのに対し、 $p < 0.65$ では一旦は増加した感染者数が減少する方向に向かう。これは、 $p = 0.65$ が臨界点であり、これを境に感染が広がるか収束するかが決定されるためであると考えられる。図 2 において $p = 0.64$ だけ収束が見えなかったのは、収束が遅いというよりも収束先が 0 であるためであるとも考えられる。

2 課題 2：モンテカルロ法による円周率推定

2.1 原理と方法

本節では、モンテカルロ法を用いて円周率 π を推定する原理と、実際に用いる計算手順をまとめる。

まず、次の定積分が成り立つことを用いる。

$$\int_0^1 \frac{4}{1+x^2} dx = \pi. \quad (1)$$

区間 $[0, 1]$ 上の一様分布に従う確率変数 X を考えると、その確率密度関数は

$$p_X(x) = \begin{cases} 1 & (0 \leq x \leq 1), \\ 0 & (\text{それ以外}) \end{cases}$$

である。このとき、任意の可積分関数 $g(x)$ に対して

$$\mathbb{E}[g(X)] = \int_0^1 g(x) p_X(x) dx = \int_0^1 g(x) dx$$

が成り立つ。 $g(x) = 4/(1+x^2)$ とおけば、式 (1) は

$$\pi = \int_0^1 \frac{4}{1+x^2} dx = \mathbb{E}\left[\frac{4}{1+X^2}\right] \quad (2)$$

と書き換えられる。したがって、 π は確率変数 $Y = 4/(1+X^2)$ の期待値として表される。

ここで、 X の独立な標本 X_1, \dots, X_N を一様分布 $\mathcal{U}(0, 1)$ に従って生成し、対応する

$$Y_k = \frac{4}{1+X_k^2} \quad (k = 1, \dots, N)$$

を計算する。このとき、 Y_k は互いに独立同分布であり、その期待値は式 (2) より $\mathbb{E}[Y_k] = \pi$ である。よって、標本平均

$$\hat{\pi}_N = \frac{1}{N} \sum_{k=1}^N Y_k = \frac{1}{N} \sum_{k=1}^N \frac{4}{1+X_k^2} \quad (3)$$

は π の推定量として用いることができる。大数の法則により、 $N \rightarrow \infty$ の極限で

$$\hat{\pi}_N \rightarrow \pi \quad (\text{ほとんど確実に})$$

が成り立つため、式 (3) はモンテカルロ法による円周率の一貫した推定量になっている。さらに中心極限定理から、 N が十分大きいとき $\hat{\pi}_N$ は平均 π 、分散が $O(N^{-1})$ の正規分布に近づき、推定誤差の大きさは概ね $N^{-1/2}$ のオーダーで減少する。

本実験では、サンプル数 N を

$$N \in \{10^2, 10^3, 10^5, 10^7\}$$

とし、各 N について独立な試行を 10 回行う。各試行で一様乱数列 $\{X_k\}_{k=1}^N$ を生成し、式 (3) により推定値 $\hat{\pi}_N$ を求める。得られた 10 個の $\hat{\pi}_N$ から、 N ごとに

- 推定値の平均（系統誤差の確認）、
- 推定値の標準偏差（統計誤差の大きさ）

を算出し、サンプル数 N の増加に伴う精度向上の様子を評価する。

2.2 実装上の工夫

本実験ではサンプル数 M を $10, 10^3, 10^5, 10^7$ と増加させながら、式 (5.10) に基づくモンテカルロ積分を行う。計算負荷が大きくなるため、以下の点に留意して実装した。

- **完全ベクトル化による高速化** Python の for ループは極端に遅いため、乱数生成から積分評価

$$\frac{4}{1+X^2}$$

の計算までを NumPy の配列演算として一度に実行した。これにより Python レベルのループを排除し、C 実装による高速な要素演算が可能になる。

- **形状 (10, M) の乱数生成** 10 回の独立試行を行うため、毎回乱数を生成するのではなく、あらかじめ形状 (10, M) の乱数行列を生成した。これにより、乱数生成コストを 1 回にまとめつつ、同じ処理を 10 行に対して並列に実行できる。
- **in-place 演算の利用** NumPy の in-place 演算（例：`s *= s, np.reciprocal(out=s)`）を積極的に利用することで、中間配列の生成を避け、メモリ消費とメモリアロケーション時間を削減した。特に $M = 10^7$ のような大規模計算では、配列の再生成がボトルネックになるため有効である。
- **標本平均の同時計算** 10 行分の計算結果に対して、NumPy の `mean(axis=1)` を用いて一度に標本平均を求めることで、試行ごとのループ処理を回避した。

上記の工夫により、大規模サンプルに対しても純粋 Python 実装と比べて大幅に高速な実行が可能となった。

2.3 結果

本実験では、各サンプル数 M について 10 回の独立試行により推定値 $\hat{\pi}$ を求め、その平均と分散を算出した。得られた結果を以下に示す。

以上の結果より、サンプル数 M の増加に伴い推定値が真値 π に収束し、分散が概ね $O(M^{-1})$ で減少している様子が確認できる。

表 1: モンテカルロ法による円周率推定結果

サンプル数 M	推定値の平均	分散	相対誤差
10	3.1589	1.3593×10^{-2}	0.55%
10^3	3.1332	4.3925×10^{-4}	0.27%
10^5	3.1410	3.6470×10^{-6}	0.02%
10^7	3.1415	2.6389×10^{-8}	0.003%

2.3.1 考察

表 tab:pi-estimation に示す通り、分散が概ね $O(M^{-1})$ で減少していることから、分散が $\frac{1}{M} [\langle f^2(X) \rangle - \langle f(X) \rangle^2]$ であることは裏付けられるが、 $[\langle f^2(X) \rangle - \langle f(X) \rangle^2]$ の値を求めるため、 $f(X) = \frac{4}{1+X^2}$ の 2 乗の平均値を求める必要がある。そこで、追加のコードを実装し、関数 pi_pi_monte_carlo_integration を実装し、 $M=10,000,000$ で 2 乗の平均値を求めた。その結果、10.2826 であり、真値 $\pi^2 \approx 9.8696$ との差は 0.4130 であった。シミュレーションによる $[\langle f^2(X) \rangle - \langle f(X) \rangle^2]$ の値は 0.3 0.5 であったことから、確かに分散が $\frac{1}{M} [\langle f^2(X) \rangle - \langle f(X) \rangle^2]$ であることが確かめられた。

3 課題 3：熱よく法とメトロポリス法のつり合い条件の証明

3.1 1. メトロポリス法が詳細つり合い条件を満たすことの証明

メトロポリス法では、現状態 X から次状態候補 Y を対称な確率分布 $Q(X, Y) = Q(Y, X)$ に従って提案し、受理確率

$$A(X \rightarrow Y) = \min \left(1, \frac{W(Y)}{W(X)} \right)$$

で遷移する。したがって遷移確率は

$$P(X \rightarrow Y) = Q(X, Y)A(X \rightarrow Y)$$

である。

詳細つり合い条件

$$P(X \rightarrow Y)W(X) = P(Y \rightarrow X)W(Y) \quad (5.19)$$

を示すため、 $W(Y) \geq W(X)$ と $W(Y) < W(X)$ の場合に分ける。

■(i) $W(Y) \geq W(X)$ の場合 このとき

$$A(X \rightarrow Y) = 1, \quad A(Y \rightarrow X) = \frac{W(X)}{W(Y)}.$$

よって

$$P(X \rightarrow Y)W(X) = Q(X, Y) \cdot 1 \cdot W(X),$$

$$P(Y \rightarrow X)W(Y) = Q(Y, X) \frac{W(X)}{W(Y)} W(Y) = Q(X, Y)W(X),$$

となり (5.19) が成り立つ。

■(ii) $W(Y) < W(X)$ の場合 このとき

$$A(X \rightarrow Y) = \frac{W(Y)}{W(X)}, \quad A(Y \rightarrow X) = 1.$$

したがって

$$P(X \rightarrow Y)W(X) = Q(X, Y) \frac{W(Y)}{W(X)} W(X) = Q(X, Y)W(Y),$$

$$P(Y \rightarrow X)W(Y) = Q(Y, X) \cdot 1 \cdot W(Y) = Q(X, Y)W(Y),$$

となる.

以上より任意の X, Y について

$$P(X \rightarrow Y)W(X) = P(Y \rightarrow X)W(Y)$$

が成立し、メトロポリス法は詳細つり合い条件を満たす.

3.2 2. 熱浴法が詳細つり合い条件を満たすことの証明

状態を $X = (x_1, x_2, \dots)$ とし、時刻 t において添字 i が状態に依存しない確率 p_i で選ばれるとする. 熱浴更新では i 番目の成分だけを更新し、それ以外は不変である.

X と Y が 1 箇所だけ異なる場合

$$X = (x_1, \dots, x_i, \dots), \quad Y = (x_1, \dots, y_i, \dots)$$

となる. このとき熱浴法の条件付き確率は

$$P(x'_i = y_i) = \frac{W(Y)}{\sum_{x_i} W(x_1, \dots, x_i, \dots)} \equiv \frac{W(Y)}{Z_i},$$

ここで Z_i は i 番目のみを動かしたときの重みの総和で、 X と Y に対して共通である.

よって遷移確率は

$$P(X \rightarrow Y) = p_i \frac{W(Y)}{Z_i}, \quad P(Y \rightarrow X) = p_i \frac{W(X)}{Z_i}.$$

したがって

$$P(X \rightarrow Y)W(X) = p_i \frac{W(Y)}{Z_i} W(X),$$

$$P(Y \rightarrow X)W(Y) = p_i \frac{W(X)}{Z_i} W(Y),$$

より両辺は一致する.

X と Y が 2 箇所以上異なる場合には熱浴法では 1 ステップで遷移できないため

$$P(X \rightarrow Y) = P(Y \rightarrow X) = 0$$

となり、この場合も自明に詳細つり合いが成り立つ.

以上より熱浴法の遷移確率は詳細つり合い条件

$$P(X \rightarrow Y)W(X) = P(Y \rightarrow X)W(Y)$$

を満たす.

3.3 3. 詳細つり合い条件 (5.19) からつり合い条件 (5.17) が従うことの証明

詳細つり合い条件

$$P(X \rightarrow Y)W(X) = P(Y \rightarrow X)W(Y) \quad (5.19)$$

がすべての X, Y について成り立つと仮定する.

このとき任意の X について

$$\sum_Y P(Y \rightarrow X)W(Y)$$

を計算する:

$$\sum_Y P(Y \rightarrow X)W(Y) = \sum_Y P(X \rightarrow Y)W(X),$$

ここで右辺の等号は詳細つり合い (5.19) より従う.

また遷移確率の和は $\sum_Y P(X \rightarrow Y) = 1$ であるから,

$$\sum_Y P(Y \rightarrow X)W(Y) = W(X) \sum_Y P(X \rightarrow Y) = W(X).$$

よってつり合い条件

$$W(X) = \sum_Y P(Y \rightarrow X)W(Y) \quad (5.17)$$

が示された. 詳細つり合いが成立すれば, W はマルコフ過程の定常分布となる.

4 結論

本章での結論と今後の課題を箇条書きでまとめる。

付録 A 使用コード一覧

主要スクリプトと入手先を列挙する。

参考文献

参考文献

[1] 数理工学実験 (2025 年度配布資料) .