数理計算演習: 第 0 課題 レポートの書き方

李 乾雨 (イ・ゴンウ) 工学部情報学科数理工学コース 1029-28-8029

提出日: 2020 年 04 月 08 日

1 はじめに

決定性有限オートマトンである計算機が生成した乱数列は真の乱数列だと言えないため擬似乱数列と呼ばれる. アルゴリズムによって生成された擬似乱数はその規則性と再現性のため, 生成法と初期値がわかれば予測可能である. しかし, いくつかの擬似乱数列生成アルゴリズムは統計的に十分な質が保証され, 自然現象のノイズなどを観測して得られる真の乱数列 (CloudFlare 社の LavaRand[1] など) より演算速度が速いというメリットがある. また, 同じシードであれば同じ結果が得られるという擬似乱数のみのメリットがあるため擬似乱数列は日常生活の様々なところで用いられている.

本レポートは擬似乱数列の生成法の 1 つである線形合同法によって生成された乱数列の特徴を実験を通して考察する. 本稿の残りの構成は以下のとおりである. 第 2 節では背景として実験環境と乱数生成方法を述べる. 第 3 節では実験の具体的な方法を説明する. 第 4 節では実験結果を報告する. 第 5 節では実験結果を考察する. そして第 6 節では本稿の内容をまとめる.

2 背景

2.1 実験環境

実験機 京都大学情報環境機構の仮想端末サービス

使用言語 C 言語

コンパイラー gcc 4.2.1

エディター nvim 0.4.3

乱数生成関数 stdlib.h の rand()

グラフの作成 Apple 社 Numbers

2.2 線形合同法

線形合同法は擬似乱数列の生成式の 1 つであり, 式 (1) の漸化式によって与えられる. A, B, M は自然数の定数である. 線形合同法で生成された乱数列は周期性を持ち, 周期の最大値は M である.

$$X_{n+1} = (A \times X_n + B) \mod M \quad M > A, M > B, A > 0, B \geq 0 \tag{1}$$

C 言語の stdlib.h ヘッダーファイルの rand() 関数は線形合同方による擬似乱数列を生成する. C 言語のレファレンスによると rand() 関数の定数は $A = 1103515245, B = 12345, M = 2^{31}$ である. 乱数の種は srand(time(null)) を用いて初期化した.

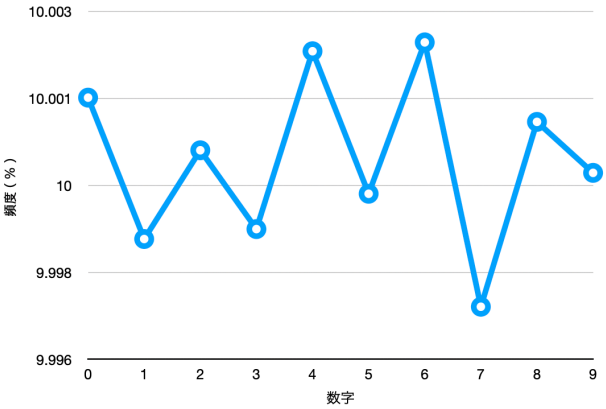

Fig. 1 各数字が出た頻度

3 実験方法

本稿では線形合同法により生成された乱数列の特徴を調べるため, rand() 関数から得られた int 形の乱数を 10 で割った余りの分布を解析する. 3 分間 0 から 9 までの乱数を生成し, 各数字が出てくる頻度を記録する. また, 同じ数字が連続して出る頻度を利録する.

4 実験結果

3 分間 413,329,999 個の乱数が生成された. Table 1 は各数字が出た頻度および同じ数字が連続して出てきた頻度の記録結果である. 本実験では 10 回以上連続して同じ数字が出たケースが観測されなかったため, 9 回まで表示する.

Fig. 1 は各数字が出た頻度を表したグラフである. Fig. 2 は連続して同じ数字が出てくる頻度の和の変化を表している. 例えばグラフの左から 3 番目の点は各数字が 3 回連続して出た回数の和を各数字が 4 回連続して出た回数の和で割ったものである.

5 考察

5.1 実験結果の考察

まず, Table 1 の第 2 列目のデータをみると, 0 から 9 までの一応に分布してるように見えるかもしれない. しかし, Fig. 1 をみるとデータの分布が強い規則性を持っていることが分かる. 偶数と奇数間の出現頻度が大きく異なり, 偶数の場合がより多い. パーセンテージで言うと偶数が 50.007% でかなり偏っている. すなわち, 本実験だけで偶数のデータが奇数のデータより約 5 万 6 千回多く出たことを意味する. これは乱数列の下位ビットが十分ランダムでないことに起因すると思う. それは偶数

Table 1　各数字の頻度および同じ数字が連続して出た頻度

数字	出た回数	1 回	2 回	3 回	4 回	5 回	6 回	7 回	8 回	9 回
0	41338227	33482618	3348365	333750	33206	3333	316	31	1	0
1	41326468	33474437	3346280	334445	33583	3304	336	29	3	0
2	41333849	33476218	3351768	335822	33510	3383	326	33	1	0
3	41327294	33474345	3348639	335273	33509	3438	322	31	2	0
4	41342087	33485769	3349027	334563	33353	3315	319	27	3	0
5	41330233	33477712	3349458	334232	33462	3302	337	24	4	1
6	41342824	33490458	3348993	334242	33458	3471	306	30	2	0
7	41320835	33472495	3344444	335476	33052	3277	344	29	4	0
8	41336211	33480730	3348982	334373	33211	3384	350	33	6	0
9	41331971	33480635	3347668	334932	33706	3336	341	29	3	0
総	413329999	334795417	33483624	3347108	334050	33543	3297	296	29	1

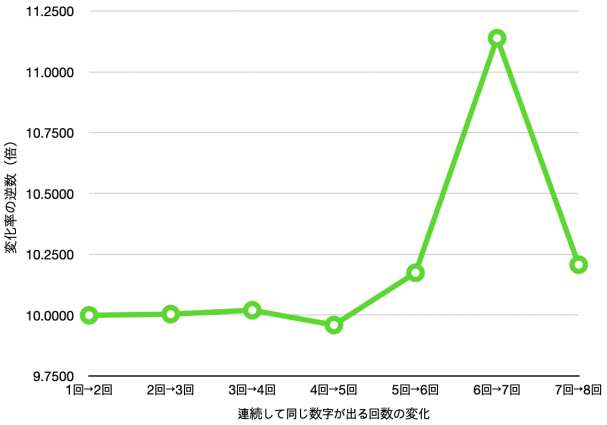

Fig. 2　連続して同じ数字が出てくる頻度の変化率

と奇数の違いが 2 進数で表したときの下 1 桁だからである.

5.2　理想的な乱数との比較

0 から 9 までの自然数が得られる理想的な一応乱数列を考えてみよう. 理想的な乱数列なら, 実行回数が十分大きければそれぞれの確率は 10% に収束するだろう. しかし, 前節で述べたように本実験で用いた擬似乱数列はそれぞれの分布に偏りがある. そのため, 完全なランダムとはの頻度比較はできない.

ここで, Fig. 2 をみると, 連続して同じ数字が出る回数が 1 回上がればその頻度が 10 倍下がることが観察できる. この 10 倍という数値は理想的な乱数列でも同じく出てくる. なぜならば, 0 から 9 までの理想的な乱数を M 個生成したとき, ある数字が n 回連続して出てくる期待値は $\frac{9M}{10^{n+1}}$ 回だからである. 理想的な乱数列の場合も M が十分大きければ, n が 1 1 がると期待値は 10 倍落ちる. ここで注目すべき言葉わ十分大きい実行回数 M である. Fig. 2 の連続 4 回から 5 回までの頻度は約 10 倍落ちたが, その後からは大きく外れる. 十分大きい実行回数があれば, 頻度の変化率が 10 倍にどんどん収束していくと思う.

6　結論

6.1　まとめ

本稿は擬似乱数生成アルゴリズムの 1 つである線形合同法によって 3 分間生成した乱数列を 10 で割った余りを観察し, 各数字が出てくる頻度や, 連続して同じ数字が出てくる頻度を記録することで, 得られた乱数列の特徴を調べてみた. 本実験から線形合同法で生成された乱数列の奇数, 偶数間の偏りと数字の連続出現の特徴がわかった. 乱数列の下位 1 ビットより, 偶数が奇数より 0.014% 多く得られた. また, 連続出現の確率は回数が 1 回上がると 10 倍落ちる.

6.2　今後の課題

本実験では擬似乱数列を 10 で割った余りの分布や連続した出現を考察したが, 16 や 32 のような 2 の乗数で割った余りを観察してもかなり意味のある課題になると思う. 本稿では線形合同法による擬似乱数列の二進数表記の下 1 桁の規則性しか探せなかったため, 擬似乱数列の規則性をより根本的に調べるため下位ビットに注目したい. また, 本当に下位ビットに強い相関関係, または規則性があるならば, 下位 n ビットをシフトして, なくなった上位 n ビットを適切に埋め込むことにより, もっとランダム性の強い擬似乱数生成法を作れるかもしれない. 切り捨てるビットの数と上位ビットを埋め込む適切な方法は今後の課題にしておきたい.

参考文献

[1] Joshua Liebow-Feeser. Lavarand in production: The nitty-gritty technical details. `blog.cloudflare.com/lavarand-in-production-the-nitty-gritty-technical-details/`, 2017.