ロジスティック写像による擬似乱数発生法†

5E－6

渡辺 裕明 (東京大学大学院理学系研究科情報科学専攻) ‡
金田 康正 (東京大学大型計算機センター) ††

1 はじめに

計算機による研究テーマの1つであるシミュレーションの分野において，モンテカルロ計算は非常に多くの乱数を必要とする．乱数発生法の一つとして，ロジスティック写像 による擬似乱数発生手法が提案されている [1][2]. 本稿では，ロジスティック写像により得られた値をIEEE 倍精度型実数表現のビット列として見なしたとき，その系列の一様性と乱雑さについて述べ，高速に発生可能な一様乱数発生法としての検討を行う．その結果，RISC ワークステーション上で，標準ライブラリ関数である `rand()` と `random()` と比べて高速に擬似乱数系列を発生できることがわかった．

2 ロジスティック写像の性質

ロジスティック写像は，カオスが存在する非線形写像として知られている．[3]. ロジスティック写像の写像関数$\tau_a(x)$ は，式 (1) で与えられる．

$$\tau_a(x) = ax(1-x) \quad (0 \le x \le 1) \qquad (1)$$

この写像に繰り返すことにより得られる系列の性質は，式 (1) における a の値により変わってくるが，特に $a = 4.0$ のとき，写像はエルゴード的であることが知られており [3]，区間 $(0,1)$ 内の $\{1/4, 1/2, 3/4\}$ を除く全ての有理数の初期値 x_0 から出発した系列は，区間 $(0,1)$ の間を非周期的かつランダムにふるまう．したがって，本稿では写像関数τ_aは $a = 4$ の場合のみを扱うことにする．

3 一様乱数の発生方法

式 (1) により発生させた系列の分布は，近似的にロジスティック分布 $1/\sqrt{x(1-x)}$ に従うことが知られている [2].

本稿では，一様乱数の発生方法としてロジスティック写像で得られた値をIEEE 倍精度型実数で表現した時に，その仮数部のビット列が一様かつランダムにふるまうと仮定した．その根拠は，この系列が非周期的であるため，数値表現のビットパターンが一様に現れると期待されることに起因している．

3.1 系列値の仮数部の一様性

前述の仮定における仮数部のビット列の一様性を検証するために，図1に示すIEEE 倍精度型実数の仮数部ビット f における，仮数部の 1 ビットから，連続した 2, 4, 8 ビットのそれぞれについて分布の一様性の検定を行った．

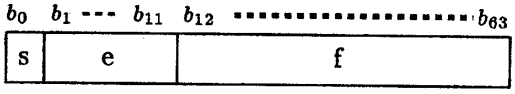

図 1: 倍精度実数型の数値表現 (IEEE 754 形式)

1 ビットの一様性を調べる場合，一定の初期値から系列を発生させ，それぞれの値について図1における仮数部の先頭ビット $\{b_{12}\}$ について $\{0,1\}$ の出現頻度の分布をとる．同様にして最下位ビット $\{b_{63}\}$ まで分布をとり，それぞれの分布について χ^2検定を行うことで一様性を検証する．

2 ビット長ごとの場合も 1 ビットの場合と同様に，仮数部の先頭 2 ビット $\{b_{12}b_{13}\}$ について $\{00, 01, 10, 11\}$ の出現頻度の分布をとり，次に 1 ビット左にずらした $\{b_{13}b_{14}\}$ の分布をとる．以下同様にして，$\{b_{62}b_{63}\}$ まで得られた分布の一様性を順次調べていく．他の 2, 4, 8 ビット長に対する方法も同様とする．

表 1に発生数を $10^4, 10^5, 10^6$ とした場合について，1, 2, 4, 8 ビット長の一様分布の検定をした結果を示す．表中の n は発生数，m はビット長，棄却されたビット位置は，自由度 $2^m - 1$，有意水準 5% の χ^2検定によって棄却されたビットの先頭位置を示している．

(なお，系列の初期値は先述の通り，区間 $(0,1)$ 内で $\{1/4, 1/2, 3/4\}$ 以外の全ての有理数をとること

†Pseudorandom Numbers Generater using Logistic Map

‡Hiroaki Watanabe, Department of Information Science, Graduate School of Science, University of Tokyo.

††Yasumasa Kanada, Computer Centre, University of Tokyo.

$n = 10^4$	
m	棄却されたビット列の先頭位置
1	12 – 20, 32, 37, 39, 53, 58
2	12 – 20, 41
4	12 – 19, 33, 35, 58
8	12 – 21

$n = 10^5$	
m	棄却されたビット列の先頭位置
1	12 – 20, 22, 23, 31, 39, 41, 48
2	12 – 23, 43
4	12 – 21, 36, 51
8	12 – 19

$n = 10^6$	
m	棄却されたビット列の先頭位置
1	12 – 24, 26, 54
2	12 – 25, 27, 41, 45, 49
4	12 – 25, 36
8	12 - 24, 32

表 1: 仮数部ビットの一様性についての検定 (系列の初期値 = 1/3)

が可能であるが, 本稿では系列の初期値をすべて 1/3 として実験した.)

また, 系列の仮数部の下位部分 16 ビットと 32 ビットのそれぞれを用いて擬似実数値乱数系列を 10^6個発生させたところ, 両系列とも有意水準 5%の検定をパスしている. これらの結果より, 下位ビットのについては, ほぼ一様な分布をしていることが分かる.

3.2 系列値の仮数部の乱雑さ

次に系列の乱雑さを調べるために, 前節と同様に系列の仮数部の下位 16 ビットと 32 ビットのそれぞれを用いて擬似実数値乱数を 10^6個発生させ, 系列の乱雑さについての検定法の一つである t 個の数の最大値検定 [4] を $t = 2, 4, 8, 16, 32, 64$ の場合に適用させた. その結果, 両系列ともこれらの t の値における有意水準 5%の検定では, すべての t でパスしていた.

4 他方式との発生速度の比較

DEC AlphaAXP21064(175MHz) 上において, 本手法とシステムの標準ライブラリ関数 `rand()`, `random()` との乱数の発生速度を比較するために, 1 秒当たりの乱数発生速度を求め, その結果を表 2 に示す. ここで, `rand()` は線形合同法, `random()` は非線形なフィードバック加算による乱数発生法*である.

	発生速度 [M 個/s]
本手法	10.680
rand	4.706
random	3.254

表 2: 乱数の発生速度の比較

この結果より, 本手法は発生速度の点で他の 2 つの発生法と比較して高速な発生方法であることが分かる. `rand()` より速い理由としては, 1 つの乱数を発生させるのに, 本手法では乗算 2 回, 減算 1 回, ビット演算 1 回の計算を行うが, 剰余算を行う必要が無いためであると推測される.

5 おわりに

本稿では, 一様乱数発生の一手法としてロジスティック分布に基づく擬似乱数発生法を提案し, その一様性と高速性について検討した. その結果, 発生速度に関して他の発生法に対して優位になることがあることを示した. しかし, 乱雑についての検定は十分になされていないので, さらなる検討を行うことが今後の課題である.

参考文献

[1] 香田 徹, 柿本 厚志: 擬似乱数とカオス. 情報処理学会論文誌 Vol. 27, No. 3, pp. 289 – 296(1986).

[2] S. C. Phatak, S. Suresh Rao: Logistic map: A possible radom-number generator, *Phys. Rev.* E, Vol. 51, No. 4, pp. 3670 – 3678(1995).

[3] 伊藤 俊次. 一次元カオス. 別冊 数理科学「現象にひそむ非線形」, サイエンス社, pp. 15 - 24, 1989.

[4] Knuth, D. E. : *The Art of Computer Programming*, Vol. 2 , Seminumerical Algorithrms, 2nd ed. , Addison-Wesley, (1981). ; 渋谷政昭 (訳) : 準数値算法/乱数, サイエンス社, 東京, (1981).

[5] S. K. Park and K. W. Miller(訳:西村 恕彦): 乱数生成系で良質なものはほとんどない, *bit*, Vol. 25, No. 4 – No. 5, (1993).

*DEC OSF/1 オンラインマニュアルより