論　文

整数ロジスティック写像と撹拌演算による乱数生成

董　　際国[†a)]　　森田　啓義[†b)]

Random Numbers Generation by means of Integer Logistic Map and Mixing Operation

Jiguo DONG[†a)] and Hiroyoshi MORITA[†b)]

あらまし　整数演算を用いて計算精度 n ビットのロジスティック写像 $x_{t+1} = 4x_t(1-x_t)$ を計算し，計算過程に存在する $2n$ ビットの内部状態を利用した撹拌方法に基づく擬似乱数生成法を提案した．また，生成した乱数列に対し，初期値敏感性をハミング距離の収束の速さで提案法と MT と比較を行い，提案法の効果を示した．そして，計算精度 $n = 128$ ビットで生成される 2 進数列に対し，統計的検定を行い，一様性をもつとの結論を得た．

キーワード　整数演算，ロジスティック写像，撹拌，初期値敏感性，MT

1. まえがき

様々な応用分野で乱数が役立つ．特に，インターネット環境の発達で，通信や情報セキュリティ分野でより多くの擬似乱数を必要としている [1]．

今日，最も広く普及している乱数生成法は線形合同法と線形フィードバックシフトレジスタである．特に最近，GFSR を改良した MT [2] は長周期，良質な乱数を高速に生成できることで，著名である．しかし，MT は他の線形漸化式に基づく生成法と同じく，系列の一部から他の部分を推測できる．情報セキュリティ用には不向きである [3]．

一方，簡単な構成で複雑な数列を生成することで非線形関数（式 (1)）による乱数の生成の試みが 1947 年にもなされた [4]．しかし，式 (1) を計算して得られる x_t の系列は，一様分布にならず，U 字型の分布をもっている [5]．それに対し，しきい値法 (Kozak) [6] や多重化 (Arneodo) [7] など，ロジスティック写像から一様分布の擬似乱数を生成する方法が提案され，更に香田らにより，カオスを用いた 2 値系列生成方法及びその検定方法，多重化による生成法も発展し，報告された [1], [8]〜[11]．

$$x_{t+1} = 4x_t(1-x_t) \quad (0 < x_t < 1,\ t = 0, 1, \ldots) \tag{1}$$

これまでのカオスによる乱数生成の研究は多くの成果を得ているが，それらの乱数生成方法はコンピュータで計算し，産業技術として利用するとき，いくつかの問題が残されている．

まず，しきい値法による 2 値系列生成法は式 (1) を 1 回計算して，1 ビットしか出力しない．多重化による方法も，多重化回数を大きくしないと品質の良い乱数列が得られないことから，乱数生成の効率が良くない．

次に，一般に，カオスの計算は科学計算として浮動小数点演算が用いられる．しかし，浮動小数点演算はコンピュータシステムによって複数の規格が存在し，異なった規格の浮動小数点演算でカオスを計算すると，初期値を同じ値で入力しても同じ計算結果を得るには，何らかの補正が必要である．

そして，安価な産業用マイクロコントローラは浮動小数点演算をサポートしていない．使用できるメモリも少ない．したがって，カオスの計算以外に複雑な計算をしないシステムでもカオスを高速に生成するために，高価な汎用 CPU あるいは専用ハードウェアを導入しないといけない．すなわち，カオスシステムは安価な身近な産業技術になりにくい．

更に，広く産業技術への応用を目的としたとき，応

† 電気通信大学大学院情報システム学研究科，調布市
Graduate School of Information Systems, The University of Electro-Communications, 1–5–1 Chofugaoka, Chofu-shi, 182–8585 Japan
a) E-mail: jiguo@appnet.is.uec.ac.jp
b) E-mail: morita@is.uec.ac.jp

用目的に応じた性能–コストの組合せの選択を可能とした生成器が求められる．すなわち，性能–コストのニーズに柔軟に対応できることである．

本研究は幅広い産業技術へ応用できる擬似乱数生成法を提案する．提案法は整数の分割演算を用いた式 (1) の計算と，式 (1) の n ビットでの整数演算過程に存在する $2n$ ビットの内部状態を利用した撹拌方法からなる．

以下，**2.** に整数を用いる式 (1) の計算方法を示し，**3.** に撹拌方法（乱数生成方法）を提案する．**4.** は提案法と MT が生成する擬似乱数にハミング距離を用いた初期値敏感性の検定の比較など，提案法の有効性を確認する．**5.** に 8 項目の統計的検定の結果を示し，**6.** に結論を述べる．

2. 整数を用いたロジスティック写像の計算

式 (1) の主な計算は相補する x_t と $1-x_t$ の乗算である．式 (1) を計算精度 2 進小数 n ビットの固定小数点演算でするとき，x_t が $\frac{1}{2^n}$ に離散化され，閉区間 $[\frac{1}{2^n}, 1-\frac{1}{2^n}]$ の中で振舞いをする．したがって，式 (1) の計算は初期値 $x_0 = 0.25,\ 0.5,\ 0.75$ を除けば，固定小数点演算でも，不動点に陥ることやけた落ちがなく計算できる．

2.1 整数を用いたロジスティック写像の計算

式 (1) の固定小数点演算は整数で（式 (2)）実現できる [12]．ここに，n は計算精度のビット数，$\lfloor\ \rfloor$ は小数点以下を捨てる．

$$\begin{aligned} &X_{t+1} = \lfloor 4X_t(2^n - X_t)/2^n \rfloor \\ &(1 \le X_t \le 2^n - 1,\ t = 0, 1, \ldots) \end{aligned} \tag{2}$$

しかし，式 (2) をそのまま計算機で計算すると，計算過程にけた落ちが発生しないように，計算機が計算できる整数の半分の計算精度しかできない．本研究は整数の分割計算で，任意の計算精度の式 (2) の計算方法を示す．

2.2 ロジスティック写像の整数分割計算

$2m$ ビットの整数演算をサポートするシステムであれば，式 (2) はけた落ちがなく，m ビットの整数演算で行える．ここにいう整数演算は整数の乗算，加算，ビットシフト，ビットごとの論理演算を指す．これを利用して，式 (2) に対し計算精度 n ビットの固定小数点演算を $\lfloor((n-1)/m)\rfloor + 1$ けたの m ビット（2^m 進数）の整数に変換して，10 進数の多けた演算と同じ原理で，2^m 進数の整数の多けた演算を用いて高速に計算できる．

したがって，式 (2) の計算は容易に計算精度を拡張できる．この方法を用いた整数演算は異なる種類の計算機システムでも，その計算能力（サポートしている整数の乗算（器），加算，論理演算などのビット数 m）に応じた整数の分割計算だけで，同じ入力に対し，同じ出力が得られる．PC とマイコンによる乱数生成速度の例を **4.2** に示す．

式 (1) を計算して，対称性を利用したしきい値法や多重化による乱数生成法が知られている [8]．しかし，それらの方法は乱数生成効率が良くないという問題がある．本研究は式 (2) の整数演算過程に $2n$ ビットの内部状態が存在していることに注目し，それを有効に利用して，より効率の良い乱数生成方法を提案する．

3. 乱数の生成：撹拌

式 (2) を，式 (3) から (7) の演算過程に分解して考える．

$$A_t = X_t \tag{3}$$

$$B_t = 2^n - X_t \tag{4}$$

$$C_t = A_t B_t \tag{5}$$

$$D_t = 4C_t = 4A_t B_t \tag{6}$$

$$X_{t+1} = \lfloor D_t/2^n \rfloor \tag{7}$$

提案方法では，式 (6) における C_t を 4 倍する演算を，通常の左への 2 ビットシフト演算ではなく，式 (6) の 4 倍算を左への 2 ビット巡回シフト演算とする実装を考える．従来の式 (2) の計算は，D_t の下位 n ビットが切り捨てられるが，本研究では D_t の $2n$ ビットを利用して乱数を生成するので，左への 2 ビットシフト演算で空いた D_t の最下位 2 ビットの状態を決める必要がある．

整数 A_t，B_t，C_t，D_t の 2 進展開を式 (8) から (11) を用いて表す $((a_i, b_i \in \{0,1\},\ 0 \le i \le n-1)$, $(c_i, d_i \in \{0,1\},\ 0 \le i \le 2n-1))$.

$$A_t = a_0 2^{n-1} + \cdots + a_i 2^{n-1-i} + \cdots + a_{n-1} \tag{8}$$

$$B_t = b_0 2^{n-1} + \cdots + b_i 2^{n-1-i} + \cdots + b_{n-1} \tag{9}$$

$$C_t = c_0 2^{2n-1} + \cdots + c_i 2^{2n-1-i} + \cdots + c_{2n-1} \tag{10}$$

$$D_t = d_0 2^{2n-1} + \cdots + d_i 2^{2n-1-i} + \cdots + d_{2n-1} \tag{11}$$

更に，出力する擬似乱数 R_t も同様に 2 進展開により

$$R_t = r_0 2^{n-1} + \cdots + r_i 2^{n-1-i} + \cdots + r_{n-1} \quad (r_i \in \{0,1\},\ 0 \le i \le n-1) \tag{12}$$

と表す．

式 (2) の整数演算過程中の $2n$ ビットの整数 D_t の 2 進展開の係数 $(d_0 d_1 \ldots d_{2n-1})$ を前半分と後半分に分けて，両者の排他的論理和 (XOR) 演算をとることにより，$(r_0 r_1 \ldots r_{n-1})$ を求める．すなわち，式 (13) の乱数生成（撹拌）法を提案する．ここに，$\oplus$ はビットごとの排他的論理和演算である．

$$\begin{pmatrix} r_0 \\ r_1 \\ \vdots \\ r_{n-2} \\ r_{n-1} \end{pmatrix} = \begin{pmatrix} d_0 \\ d_1 \\ \vdots \\ d_{n-2} \\ d_{n-1} \end{pmatrix} \oplus \begin{pmatrix} d_n \\ d_{n+1} \\ \vdots \\ d_{2n-2} \\ d_{2n-1} \end{pmatrix} \tag{13}$$

以下では，式 (13) で生成した n ビットの R_t が，一様分布となる可能性を，式 (2) の演算過程を 2 進数に展開して考察する．

式 (5) は図 1 のように 2 進展開して計算される．

式 (8) と式 (9) を用いて式 (5) を展開し，整理すると，式 (14) が得られる．

$$\begin{aligned} C_t &= (a_0 2^{n-1} + \cdots + a_{n-1})(b_0 2^{n-1} + \cdots + b_{n-1}) \\ &= b_0 2^{n-1}(a_0 2^{n-1} + \cdots + a_{n-1}) + \cdots \\ &\quad + b_{n-1}(a_0 2^{n-1} + \cdots + a_{n-1}) \\ &= a_0 b_0 2^{2n-2} + (a_0 b_1 + a_1 b_0) 2^{2n-3} + \cdots \\ &\quad + (a_0 b_{n-1} + a_1 b_{n-2} + \cdots + a_{n-1} b_0) 2^{n-1} \\ &\quad + (a_1 b_{n-1} + a_2 b_{n-2} + \cdots + a_{n-1} b_1) 2^{n-2} + \cdots \\ &\quad + (a_{n-2} b_{n-1} + a_{n-1} b_{n-2}) 2 + (a_{n-1} b_{n-1}) \\ &= c'_0 2^{2n-1} + \cdots + c'_i 2^{2n-1-i} + \cdots + c'_{2n-1} \end{aligned} \tag{14}$$

ここに 2^{2n-1-i} 項の係数 c'_i（c'_i は 0 以上の整数）は式 (15) で与えられる．

$$\begin{array}{cccccc} & & & a_0 & \ldots & a_{n-1} \\ & & \times & b_0 & \ldots & b_{n-1} \\ \hline & & & a_0 b_{n-1} & \ldots & a_{n-1} b_{n-1} \\ & & \vdots & \vdots & & \\ + & a_0 b_0 & \ldots & a_{n-1} b_0 & & \\ \hline c_0 & c_1 & \ldots & c_n & \ldots & c_{2n-1} \end{array}$$

図 1　整数乗算の 2 進展開
Fig. 1　Binary progressing of integer multiplication.

$$c'_i = \begin{cases} 0, & (i = 0) \\ \displaystyle\sum_{(j,l):\, j+l=i-1} a_j b_l, & (1 \le i \le 2n-1) \end{cases} \tag{15}$$

更に，$i+1$ けたから i けたへのけた上がり数を k_i（k_i は 0 以上の整数）とおくと，k_i は式 (16) で与えられる．

$$k_i = \begin{cases} \lfloor (c'_{i+1} + k_{i+1})/2 \rfloor, & (0 \le i \le 2n-2) \\ 0, & (i = 2n-1) \end{cases} \tag{16}$$

式 (10)，(14)，(16) から，式 (17) が成り立つ．

$$c_i = (c'_i + k_i) \bmod 2, \quad (0 \le i \le 2n-1) \tag{17}$$

したがって，D_t の 2 進展開の係数 d_i は，式 (18) で計算される．

$$d_i = \begin{cases} c_{i+2}, & (0 \le i \le 2n-3) \\ c_{i-2n+2}, & (2n-2 \le i \le 2n-1) \end{cases} \tag{18}$$

式 (18) を行列表現すると，式 (19) となる．

$$\begin{pmatrix} d_0 \\ \vdots \\ d_{n-2} \\ d_{n-1} \\ \vdots \\ d_{2n-3} \\ d_{2n-2} \\ d_{2n-1} \end{pmatrix} = \left(\begin{pmatrix} a_1 & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ a_{n-1} & \ddots & \ddots & a_0 \\ 0 & \ddots & \ddots & a_1 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & a_{n-1} \\ 0 & \ddots & \ddots & 0 \\ a_0 & \cdots & \cdots & 0 \end{pmatrix} \begin{pmatrix} b_0 \\ \vdots \\ b_{n-1} \end{pmatrix} + \begin{pmatrix} k_2 \\ \vdots \\ k_n \\ k_{n+1} \\ \vdots \\ k_{2n-1} \\ k_0 \\ k_1 \end{pmatrix} \right) \bmod 2 \tag{19}$$

式 (19) から，D_t の各ビットと入力 A_t の各ビット

の関わり方を見ると，d_{n-2} が A_t の全てのビットと関っていることと，d_{n-2} を中心として両側に離れるほど d_i が依存する A_t のビットの数が減少していくことが分かる．

この特徴は式 (1) を浮動小数点倍精度で計算し，その仮数部の乱雑性を調べて得た仮数部の下位ビットほど乱雑である結果 [13] に矛盾しない．よって，我々は，出力される擬似乱数の各ビットに関わる入力 A_t のビット数をより多くすることで，より乱雑な乱数列が得られると考える．

式 (19) より，式 (13) は式 (20) となる．

$$\begin{aligned} R &= ((A_1B + K_1) \bmod 2) \oplus ((A_2B + K_2) \bmod 2) \\ &= ((A_1B \bmod 2 + K_1 \bmod 2) \bmod 2) \\ &\quad \oplus ((A_2B \bmod 2 + K_2 \bmod 2) \bmod 2) \\ &= (A_1B \bmod 2) \oplus (K_1 \bmod 2) \\ &\quad \oplus (A_2B \bmod 2) \oplus (K_2 \bmod 2) \\ &= ((A_1 \oplus A_2)B \bmod 2) \oplus (K_1 \bmod 2) \\ &\quad \oplus (K_2 \bmod 2) \\ &= (AB \bmod 2) \oplus (K_1 \bmod 2) \oplus (K_2 \bmod 2) \end{aligned} \tag{20}$$

ここに，R，A_1，B，K_1，A_2，K_2，A はそれぞれ，

$$R = \begin{pmatrix} r_0 \\ r_1 \\ \vdots \\ r_{n-2} \\ r_{n-1} \end{pmatrix}, A_1 = \begin{pmatrix} a_1 & a_0 & \cdots & 0 & 0 \\ a_2 & a_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-1} & a_{n-2} & \cdots & a_1 & a_0 \\ 0 & a_{n-1} & \cdots & a_2 & a_1 \end{pmatrix},$$

$$B = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-2} \\ b_{n-1} \end{pmatrix}, K_1 = \begin{pmatrix} k_2 \\ k_3 \\ \vdots \\ k_n \\ k_{n+1} \end{pmatrix},$$

$$A_2 = \begin{pmatrix} 0 & 0 & \cdots & a_3 & a_2 \\ 0 & 0 & \cdots & a_4 & a_3 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \\ a_0 & 0 & \cdots & 0 & 0 \end{pmatrix}, K_2 = \begin{pmatrix} k_{n+2} \\ k_{n+3} \\ \vdots \\ k_0 \\ k_1 \end{pmatrix},$$

$$A = A_1 \oplus A_2 = \begin{pmatrix} a_1 & a_0 & \cdots & a_3 & a_2 \\ a_2 & a_1 & \cdots & a_4 & a_3 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n-1} & a_{n-2} & \cdots & a_1 & a_0 \\ a_0 & a_{n-1} & \cdots & a_2 & a_1 \end{pmatrix}$$

である．

入力 A_t は n ビットであることから，式 (20) より，R_t の各ビットは異なった形で，A_t の全てのビットを含んでいることが分かる．提案法は 1 回の XOR 演算で，計算精度と同じビット数を出力するので，最も効率の良い（撹拌）生成方法といえる．

4. 提案法の効果

4.1 多重化しなくても良質な擬似乱数を生成可能

4.1.1 香田らの乱数検定実験を追試

香田ら [8] によれば，ロジスティック写像（式 (1)）に多重化を行えば，良質な擬似乱数を生成できる．そのことを連検定と組合せ検定の結果で示した．本研究は同じ方法で，この二つの検定を追試し，同等な結果を得られ，確認できた．

4.1.2 提案法と MT を同方法で検定

MT と提案法（精度 64 ビット）で生成した整数の擬似乱数列を 10^5 個の 0〜1 の実数数列に変換し，香田ら [8] と同様な方法で連検定と組合せ検定を行った（表 1）．

検定は変換された実数系列に対し，しきい値 c 未満なら 0，以上なら 1 を対応させ，生成された 2 値系列に対して行う．

連検定は 2 値系列に連（同種の数値が続く長さ）の分布を調べ，頻度の少ない連を一つのカテゴリーにまとめ，カテゴリー数 $\nu+1$ の（等確率でない）ベルヌーイ試行における連の発生頻度（理論値）との近さを χ^2 検定を用いて比較する．得られる自由度 ν の χ^2 値 χ_ν が自由度 ν の χ^2 分布の有意水準 5%のときの χ^2 値 χ_0 と比較し，その比 χ_ν/χ_0 が < 1 であれば，生成した 2 値系列はベルヌーイ試行に十分近い系列とみなす．

組合せ検定は 2 値系列に対し，長さ 20 を一組にし

表 1 提案法と MT の連検定と組合せ検定の結果
Table 1 Result of run test and combination test by proposal method and MT.

		Run test					Combination test				
c		.30	.40	.50	.60	.70	.30	.40	.50	.60	.70
ν		19	14	12	14	19	12	13	13	13	12
$\frac{\chi_\nu}{\chi_0}$	MT	.47	.84	.77	.63	.55	.53	.41	.56	.63	.48
	提案法	.58	.42	.92	.34	.55	.19	.36	.30	.30	.36

て，1 が現れる回数を調べ，同じようにカテゴリー数 $\nu+1$ の（等確率でない）ベルヌーイ試行におけるその発生頻度（理論値）との近さを χ^2 検定を用いて比較する．得られる自由度 ν の χ^2 値 χ_ν が自由度 ν の χ^2 分布の有意水準 5%のときの χ^2 値 χ_0 と比較し，その比 χ_ν/χ_0 が <1 であれば，生成した 2 値系列はベルヌーイ試行に十分近い系列とする．

4.1.3 結　　果

表 1 は，MT と提案法をそれぞれ用いて生成した整数系列を 0〜1 の実数数列に変換し，しきい値 c で生成した 2 値系列に対し行った連検定と組合せ検定の結果 χ_ν/χ_0 である．ν は χ^2 検定における自由度である．提案法は MT と同様，多重化することがなくても，検定に合格する良好な擬似乱数を生成できることを示している．

4.2 生 成 速 度

計算速度は，プロセッサの種類とプログラミング方法やコンパイラの種類によるが，ここにノート PC と 8 ビットマイコンでの計算例を表 2 に示す．

提案法の乱数生成は式 (2) の計算以外に数回の XOR 演算だけである．したがって，2 値系列生成法に比べ，ほぼ計算精度 (n) 倍の乱数生成速度が得られた．また，計算能力の低い 8 ビットマイコンでの乱数生成が可能であることを確認できた．

4.3 初期状態の微小な差に対する敏感な反応

カオスは初期値敏感性を有する [5]．当然，擬似乱数生成も同じように，初期状態に敏感に反応するべきである．本研究は提案法と MT に対し，微小の差をもつ二つの初期状態から生成した系列の間のハミング距離を計測して，初期状態の微小な差に対する反応の強さを測る実験を行う．

4.3.1 方　　法

二つの系列の間が無関係であればそのハミング距離 h を計測に使われるビット数 n で割った値 H ($H=h/n$) が n の数が大きくなると，0.5 に近づく．したがって，二つの微小の差しかもたない初期状態から生成した系列の間から得られた H の値が 0.5 の近傍に収束されるまでの n の値を観測することにより，初期状態の微小な差に対する反応の強さを見ることができる．

表 2　計算精度と乱数生成速度
Table 2　Calculation accuracy and random numbers generation speed of the logistic map. C is calculation accuracy (bit) and S is generation speed. (PC: By Genuine Intel(R) 1.50 GHz, MC: By ATmega168V 8 MHz)

C		32	64	96	128	160	192	224	256
S	PC (Gbit/s)	1.9	1.0	.71	.56	.51	.43	.36	.31
	MC (Kbit/s)	262	166	125	101	80	67	60	54

MT は 624 ワードの内部状態を有し，内部状態の更新はワード（32 ビット）単位で行われている [2]．ランダムに決まった内部状態①の任意のワードにある 1 ビットを反転した内部状態②と，乱数を生成しながら，ハミング距離 h を計算し，n と H を出力する．

128 ビットの整数で式 (1) を計算する ($L128$)．ランダムに決まった初期状態①と，1 ビット反転された内部状態②と，MT と同じように，(式 (19) の d_0〜d_{127} の 128 ビット）ハミング距離 h を計算し，n と H を出力する．同時に，提案法（式 (13)）により，①と②から生成した擬似乱数 R_t についても，ハミング距離 h を計算し，n と H を出力する．

以上の実験を 100 回行い，H の推移状態などを観察し，比較する．

4.3.2 手　　順

S1．ランダムに初期状態①を決め．

S2．状態①の任意の位置にある 1 ビットを反転して状態②に．

S3．初期状態①と②からそれぞれ乱数を出力して，h を計算し，適当なところに n と H を出力する．

S4．H が 0.5 の近傍に収束したら，終了する（本研究は $n=3.2\times10^9$ まで）．

S5．S1〜S4 を 100 回行う．

4.3.3 結　　果

図 2 (a)，(b)，(d) にそれぞれ $L128$ と MT と提案法の計測結果を示す．横軸は n に底 10 の対数をとった値で，縦軸は H の値である．

図 2 (a) の h の計測に使われる系列は計算精度 128 ビットで計算したロジスティック写像（式 (1)）の x_t の小数点以下の部分，すなわち，式 (19) の d_0〜d_{127} (式 (2) の X_t）である．図 2 (a) から，初期状態によって，H が 0.5 に近づくにつれ n の値が大きく異なっていることを観測できる．初期状態に異なるビットの位置が上位であるほど，H が 0.5 に近づくにつれ n の値が小さくなる．この特徴をなす原因は式 (19) から観測できる．A_t のより上位ビットほど，d_0〜d_{127} の中のより多くのビットに関与していることである．

一方，図 2 (b) の MT のそれは，どんな初期状態でも 10^6 ビットほど，二つの状態から近い系列が続いて生成されていることが観測される．H が 0.5 に近づく

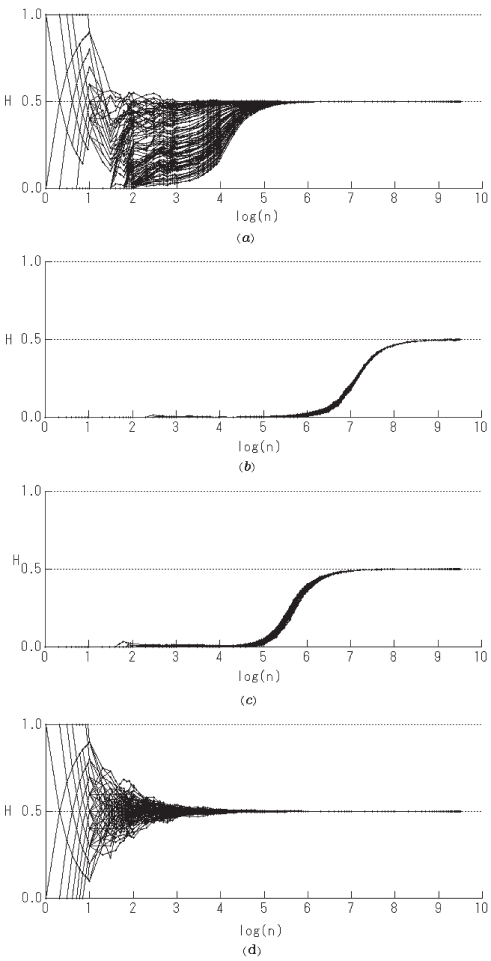

図 2　ハミング距離の推移の比較
Fig. 2　Comparison of transitions of Hamming Distance.

につれ n の値は 10^8 ビットほど必要である．

そして，提案法である図 2 (d) からは初期状態の異なる位置に関係せず，その違いを敏感に反応し，速く H の値を 0.5 に収束させていることが観測できる．このことは，式 (13) で提案法の出力である R_t の全てのビットが A_t の全てのビットに影響されていることと一致している．

なお，MT に生成速度と初期値依存性を改良した SFMT [14] が知られている．本研究は SFMT19937 に対し，MT と同様な実験を行った．その結果を図 2 (c) に示す．その H の推移が図 2 (b) の MT と同じ形であるが，MT よりも速く H が 0.5 に収束していることを確認できる．それは，SFMT が MT よりも速く内部状態の微小の違いを拡散していることを意味する．

また，任意の二つの初期状態で生成した MT の系列からは図 2 (d) と同等な計測結果が得られることを確認した．

以上の結果から，式 (1) を整数で計算する式 (2) から得た U 字型の分布をもつ X_t の系列に対し，従来切り捨てられている D_t の下位ビットを有効に利用した式 (13) による撹拌が，一様分布の R_t の系列を出力すると考えられる．また，提案法は初期状態の微小な違いに最も敏感に反応していることが分かる．なお，より厳しい 8 項目の統計的検定による擬似乱数列の一様性に対する検証を **5.** で行う．

5.　統計的検定

この章は計算精度 128 ビットとしたとき，提案法により生成した R_t の 2 進系列に対し，7 項目の χ^2 検定に加え，シリアル相関検定と計 8 項目の統計的検定を行う．そして，その結果について検討する．

5.1　検 定 方 法

数列のランダム性を検査するために様々な手法が提案されているが，ここでは①等分布検定，②シリアル検定，③ポーカー検定，④クーポン券収集検定，⑤連検定，⑥衝突検定，⑦誕生日間隔検定の χ^2 検定と，⑧シリアル相関検定の計 8 項目の検定方法を扱う．検定方法は，評価指標に対して，与えられた数列がランダム，すなわち数列の成分が独立かつ一様な分布に従って発生しているものと仮定したときの理論分布に対し，①～⑦は数列から得られる経験分布の適合度を表す χ^2 値を求めることで検定する方法であり，⑧は数列から得られた相関係数を求めることで検定する方法である．本節ではこれらの検定方法を用いて，計算精度 128 ビットのとき，提案法によって生成された 128 ビット数列のランダム性について検討する．

具体的な検討方法を，8 ビット整数列の等分布検定 (**5.2.1**) を例に説明する．まず提案法によって生成された 128 ビット数列を逐次連結することで，十分な長さの 0, 1 ビット列（生成 0, 1 ビット列）を作成する．生成 0, 1 ビット列から 8 ビットずつ順次切り出すことで，サンプル数 S の 8 ビット整数 (0～255) 列を得る．この整数列に対し等分布性，すなわち 0～255 の離散一様分布に対する適合度を表す χ^2 値を求める．同じ実験を 1000 回行い，1000 個の χ^2 値が得られるので，この χ^2 値の経験分布と χ^2 分布を比較する．生成 0, 1 ビット列が独立なベルヌーイ分布列であれば，経験分布は χ^2 分布と一致する．ここでは 1000 個の χ^2 値の経験分布を，χ^2

分布の $P < 1\%, 5\%, 25\%, 50\%, 75\%, 90\%, 95\%$値で集計することで χ^2 分布との比較を行う．この手法はサンプル数 S が十分大きいときだけ有効であるが，サンプル数をあまり大きくしすぎると「局所的にランダムでない挙動を平滑化してしまう」[15] こともある．そこで複数の S を設定し，実験を行う．検定方法②～⑦についても，切り出すビット数やサンプル数 S が異なるが同様の実験を行う．なお，各検定方法の具体的な手順は [15] に示されている．

5.2 検定結果

5.2.1 等分布検定

等分布検定は生成された乱数列を 8 ビットごとに切り出すことで，(0～255) の整数系列を構成する．この整数系列の等分布性に対する χ^2 値を求め，自由度 255 の χ^2 分布との比較を行った．その度数分布を表 3 に示す．その結果は乱数列から求めた χ^2 値が，実際に χ^2 分布に従っていることを示唆している．

5.2.2 シリアル検定

二次元の等分布検定は 8 ビットの整数数列を生成して，二つの 8 ビット整数を一組にしたサンプルで等分布性に対する χ^2 値を求め，自由度 65535 の χ^2 分布と比較した．それぞれの検定から得られた χ^2 値の各区間にある頻度分布を表 4 に示す．乱数列から求めた χ^2 値が，実際に χ^2 分布に従っていることを示唆している．

5.2.3 ポーカー検定

ポーカー検定は 3 ビットの整数数列 (0～7) 5 組に対して各組内での整数値の一致/不一致に関する 5 種類のカテゴリーのポーカー検定（[15]，**3.2**）に相当する χ^2 値を求める．これを自由度 4 の χ^2 分布と比較する．得られたχ^2 値の区間ごとの頻度分布を表 5 に示す．乱数列から求めた χ^2 値が，実際に χ^2 分布に従っていることが分かる．

サンプル数 2^{10} の場合の χ^2 値の上側の $P > 99\%$に入る数が 28 個 $(1000 - 972)$ で，理論値の 1%に対し 2.8%との結果を得た．理論値に比べると，やや多いが，サンプル数を増やした場合の χ^2 検定値の分布はそれぞれの理論値の近傍にある．したがって，サンプル数 2^{10} のときの結果はサンプルの数が少ないことが原因と考えられる．

5.2.4 クーポン券収集検定

クーポン券収集検定は 3 ビットの整数数列 (0～7) に対し，38 回打切りのクーポン券収集検定に相当する χ^2 値を求め，自由度 30 の χ^2 分布と比較する．それぞれの検定で得られた χ^2 値の各区間にある頻度分布を表 6 に示す．乱数列から求めた χ^2 値が，実際に χ^2 分布に従っていることを示唆している．

5.2.5 連検定

連の検定は 32 ビットの整数の長さ S の数列 $(0 \sim 2^{32} - 1)$ に対し，長さ 6 までの連検定に相当する χ^2 値を求め，自由度 6 の χ^2 分布と比較する．それぞれの検定で得られた χ^2 値の各区間にある頻度分布の上昇連の検定結果を表 7 に示す．その結果も乱数列から求めた χ^2 値が，実際に χ^2 分布に従っていることを示唆している．なお，下降連の検定も行ったがほぼ同様な結果が得られている．

5.2.6 衝突検定

衝突検定は 20 ビットの整数 $(0 \sim 2^{20} - 1)$ の，長さ 2^{14} の数列に対して [15]，**3.2** にある衝突回数 101，

表 3　等分布検定の結果
Table 3　Result of the equidistribution test.

	CHI-SQUARE DISTRIBUTION ($P <$)						
S	1%	5%	25%	50%	75%	95%	99%
2^{16}	10	48	260	521	754	927	986
$2^{16} \times 10$	13	53	262	510	747	945	991
$2^{16} \times 100$	6	46	255	501	749	942	990
$2^{16} \times 1000$	8	52	256	504	757	965	995

表 4　シリアル検定の結果
Table 4　Result of the serial test.

	CHI-SQUARE DISTRIBUTION ($P <$)						
S	1%	5%	25%	50%	75%	95%	99%
2^{20}	12	47	240	499	758	959	990
$2^{20} \times 10$	7	45	234	508	755	951	989
$2^{20} \times 100$	7	38	228	492	739	951	991
$2^{20} \times 1000$	11	61	248	475	723	935	985

表 5　ポーカー検定の結果
Table 5　Result of the poker test.

	CHI-SQUARE DISTRIBUTION ($P <$)						
S	1%	5%	25%	50%	75%	95%	99%
2^{10}	2	54	284	553	799	940	972
$2^{10} \times 10$	8	48	244	501	772	959	989
$2^{10} \times 100$	11	44	224	454	730	939	989
$2^{10} \times 1000$	10	47	241	508	751	963	994

表 6　クーポン券収集検定の結果
Table 6　Result of the coupon collector's test.

	CHI-SQUARE DISTRIBUTION ($P <$)						
S	1%	5%	25%	50%	75%	95%	99%
2^{10}	12	58	273	505	750	938	989
$2^{10} \times 10$	11	46	245	493	725	946	988
$2^{10} \times 100$	8	47	268	511	751	936	988
$2^{10} \times 1000$	8	63	279	528	766	946	993

表 7　連検定の結果
Table 7　Result of the run test (up).

	CHI-SQUARE DISTRIBUTION ($P <$)						
S	1%	5%	25%	50%	75%	95%	99%
2^{16}	9	52	247	484	749	948	988
$2^{16} \times 10$	9	60	278	523	758	956	988
$2^{16} \times 100$	11	48	249	517	767	958	990
$2^{16} \times 1000$	8	49	261	507	732	953	990

表 8　衝突検定の結果
Table 8　Result of the collision test.

	CHI-SQUARE DISTRIBUTION ($P <$)						
S	1%	5%	25%	50%	75%	95%	99%
10^3	181	394	737	892	969	993	997
10^4	165	368	717	873	956	997	1000
10^5	73	156	453	697	858	962	992

表 9　誕生日間隔検定の結果
Table 9　Result of the birthday spacings test.

	CHI-SQUARE DISTRIBUTION ($P <$)						
S	1%	5%	25%	50%	75%	95%	99%
10^3	12	50	253	494	765	947	993
10^4	15	49	244	526	757	956	991
10^5	13	52	271	515	743	952	991
10^6	14	55	268	488	757	963	992

108，119，126，134，145，153 の衝突検定に相当する χ^2 値を求め，自由度 7 の χ^2 分布と比較する．それぞれの検定で得られた χ^2 値の各区間にある頻度分布を表 8 に示す．

各サンプル数において，検定値である χ^2 値の各区間にある頻度がそれぞれの理論値よりも全体的に高い値になっている．例えば，$P < 1\%$に対し，$S = 10^3$ のとき，計測された χ^2 値の数は 181，すなわち，18.1%もあった．これは検定で得られた衝突回数の分布が理論値により近い振舞いをしていて，得られた χ^2 値が小さいものが多いことを意味する．

5.2.7　誕生日間隔検定

誕生日間隔検定は 25 ビット長さ S の整数 (0〜$2^{25} - 1$) 数列を生成して，$m = 2^{25}$，$n = 2^9$ の自由度 3 の χ^2 検定を行う [15]．それぞれの検定で得られた χ^2 値の各区間にある頻度分布を表 9 に示す．乱数列から求めた χ^2 値が，実際に χ^2 分布に従っていることを示している．

5.2.8　シリアル相関検定

シリアル相関検定は 4 ビットの整数 (0〜15) の数列を生成して行う．相関係数 C が 95%の信頼区間に入る数を計数する．良質の乱数であれば，95%の信頼区間では，1000 回の検定で，C が $\mu - 2\sigma \leq C \leq \mu + 2\sigma$ に入る数は 950 以上となる [15]．それぞれの結果を表 10 に示す．良質といえる結果である．

表 10　シリアル相関検定結果
Table 10　Result of the serial correlation test.

S	$C < \mu - 2\sigma$	$C > \mu + 2\sigma$	$\mu - 2\sigma \leq C \leq \mu + 2\sigma$
2^{10}	18	19	963
$2^{10} \times 10$	33	23	944
$2^{10} \times 10^2$	30	17	953
$2^{10} \times 10^3$	21	23	956

表 11　MT と物理乱数 IDQ の衝突検定の結果
Table 11　Result of the collision test of MT and IDQ.

		CHI-SQUARE DISTRIBUTION ($P <$)						
	S	1%	5%	25%	50%	75%	95%	99%
MT	10^3	158	350	701	854	951	995	1000
	10^4	163	375	702	879	952	989	1000
	10^5	155	367	7043	871	948	991	997
IDQ	10^3	176	374	700	860	949	989	998
	10^4	181	380	695	856	943	987	997

5.3　検定結果について

8 項目の検定結果から，良好な結果を得た．ただ，衝突検定で得られた χ^2 値の振舞いは検定で得られた衝突回数の分布が理論値に近く，小さい χ^2 値の頻度が χ^2 分布よりも高い．検定で得られた χ^2 値が小さすぎることは非ランダム的な振舞いの可能性を示す [15] こともあるから，そこで，本研究は MT と物理乱数生成器 IDQ [16] に対して同様な衝突検定を行った．その結果を表 11 に示す．提案法とほぼ同等な結果であることを確認できる．なお，物理乱数の生成に多くの時間を要するため，$S = 10^5$ については行わなかった．

MT は生成した乱数列が高次の均等分布性を有し，良質な乱数性を有することが証明されていることと，物理乱数は周期がないことから，衝突検定で得られた χ^2 値の振舞いが理論分布よりやや小さいことは検定に使われる系列の非ランダム的な振舞いによるものではないと判断できる．

したがって，8 項目の検定結果から提案法で生成される 2 進数列がランダムな振舞いをする数列であることを示唆しているといえる．

本論文で述べた 8 項目の検定においては，一度のロジスティック写像の計算で 128 ビットの 2 進数列を生成し，複数の 2 進整数（サンプル）を出力する．そのため，一度のロジスティック写像の計算ごとに生成される 128 ビットの乱数の間に存在するランダムでない挙動を平滑化してしまう可能性がある．そこで更に一

度のロジスティック写像の計算に一つの 2 進ブロック（サンプル）しか出力しない（例えば：等分布検定において，一度のロジスティック写像の計算に生成される 128 ビットの中，先頭の 8 ビットだけをサンプルデータにする）方法で，同じ 8 項目の検定を行った．同等な結果を得られることが確認された．

6. むすび

本研究は式 (1) に対し，コンピュータの上に規格化されたビット長の整数演算機能を利用した計算方法を示し，PC と 8 ビットマイクロコントローラでの二つの計算速度の例を示した．また，乱数生成の高速化と良い統計的な乱数性をもつ乱数を得るため，一度の整数演算によるロジスティック写像の計算で，計算精度と同じビット数の 2 進乱数列を出力する撹拌方法を提案した．この撹拌方法は従来切り捨てられた内部状態の下位ビットを利用している．また，提案法と MT に初期状態の微小の違いに対する反応の敏感さ（初期値敏感性）の比較実験を行った．提案法は MT よりも強い初期値敏感性を有することを確認した．そして，計算精度 128 ビットで生成される 2 進数列に対し，8 項目の統計的検定を行った．その結果，厳しい統計的な検定に耐え得る乱雑な 2 進数列であることが分かった．計算精度の拡張による乱数の生成速度の低下を抑えることができた．

また，本研究が提案する生成方法は整数の整数加算，乗算，ビットの論理演算だけで実現できる．そのため整数の加算，乗算，ビットの論理演算ができるシステムであれば，ソフトウェア上でも，ハードウェア上でも，同じ計算結果が得られる．したがって，産業の幅広い分野での応用が可能となり，期待される．

なお，ロジスティック写像の周期長は計算精度 n ビットのとき $2^{n/2}$ に近い [17] と報告したが，周期に陥るメカニズムなどの周期性や初期値選びなどに関する問題の解明は，今後の課題として，まだ残されている．

文　献

[1] 香田　徹，離散力学系のカオス，コロナ社，東京，1998.

[2] M. Matsumoto and T. Nishimura, “Mersenne twister: A 623-dimensionally equidistributed uniform pseudo random number generator,” ACM Trans. Model. Comput. Simul., vol.8, pp.3–30, 1998.

[3] 岡本栄司，暗号理論入門，共立出版，東京，1993.

[4] S.M. Ulam and J. Von Neumann, “On combination of stocastic deterministic processes,” Bull. AMS., vol.53, p.1120, 1947.

[5] 長島弘幸，馬場良和，カオス入門，培風館，東京，1992.

[6] J.J. Kozak, M.K. Musho, and M.D. Hatlee, “Chaos, periodic chaos, and the random-walk problem,” Phys. Rev. Lett., vol.49, pp.1801–1804, 1982.

[7] A. Arneodo and D. Sornette, “Monte Carlo random-walk experiments as a test of chaotic orbits of maps of the interval,” Phys. Rev. Lett., vol.52, pp.1857–1860, 1984.

[8] 香田　徹，柿本厚志，“擬似乱数とカオス，” 情処学論（A），vol.27, no.3, pp.289–296, 1986.

[9] 香田　徹，梶原聖司，福重雅志，“M 系列およびチェビシェフ写像で生成されるカオスの乱雑さ，” 信学技報，NLP88-11, 1988.

[10] 香田　徹，梶原聖司，“チェビシェフ乱数列の理論的検定，” 信学技報，NLP88-53, 1988.

[11] 香田　徹，梶原聖司，“疑似乱数やカオスの理論的検定法，” 信学技報，NLP88-63, 1989.

[12] 石田邦昭，常田明夫，井上高宏，“有限ビット演算によるカオス的系列の性質，” 信学技報，CAS98-16, 1998.

[13] 渡辺裕明，金田康正，“ロジスティック写像による擬似乱数発生法，” 情処学第 53 回全大，1–65, SE–6, 1996.

[14] M. Saito and M. Matsumoto, “SIMD-oriented fast Mersenne twister: A 128-bit pseudorandom number generator,” in Monte Carlo and Quasi-Monte Carlo Methods 2006, pp.607–622, Springer, 2008.

[15] D.E. Knuth, The Art of Computer Programming vol.2 Seminumerical Algorithms, Third Ed., Adisson-Wesley, 1997.

[16] http://www.idquantique.com/

[17] 董　際国，山田孝子，庄野克房，“3 次の写像関数が生成するカオスの非線形量子化解析と予測可能性，” 信学技報，CAS2007-53, 2007.

（平成 23 年 1 月 23 日受付，6 月 14 日再受付）

董　際国　（学生員）

1984 上海科学技術大・精密機械卒．1997 山形大・人文・経済卒．1999 同大大学院社会文化システム研究科修士課程了．現在電気通信大学情報システム研究科博士後期 3 年在学．

森田　啓義　（正員：シニア会員）

1983 大阪大学大学院博士後期課程了．同年豊橋技術科学大学助手．1990 電気通信大学電気通信学部講師を経て，現在同大大学院情報システム学研究科教授．3D 画像処理，MPEG 応用，データ圧縮，誤り訂正など，情報理論とその応用に関する研究に従事．工博．情報処理学会，情報理論とその応用学会，IEEE, ACM 各会員．