統計数理研究所彙報
第32巻 第2号（1984）

有限体での演算と多次元擬似乱数

統計数理研究所 仁 木 直 人

（1984年7月 受付）

1. はじめに

モンテカルロ法やシミュレーションでは，しばしば大量の多次元一様乱数が必要とされる．多次元超立方体での多重モンテカルロ積分や，固定複数個の一様乱数からある特定の分布に従う乱数に変換して使用する場合（精度の高いWalker (1974) の方法，Niki (1979) など）は，その典型例である．また，常に固定次元のベクトルとして使われるのではなくとも，多次元一様性が保証されている乱数列については，より低い次元での独立性がある程度保証されていることになる．

擬似乱数の発生法として最もよく使われている乗算合同法は，連続した数個の乱数列を1個の多次元乱数とみなしたとき，乗数をうまく選べば，ある程度高い次元までの良い一様性を実現できる（一様性の評価についてはNiederreiter (1978a, 1978b) を参照）．しかし，乗算合同法は周期が比較的短く（法として用いた数の大きさ以下），また擬周期の問題を考慮すると，大量の多次元乱数として用いることには問題に応じた工夫が必要であろう．

乗算合同法の欠点を克服するために考案されたM系列に基づく発生法（例えば，伏見・手塚 (1981) 参照）は，周期全体を見るときには，ほとんど理想的な乱数列を発生するが，その部分列の統計的性質は必ずしも良くないとの報告が多い（Lindholm (1968)，仁木 (1983)，栗田 (1983) など）．その改良のため，M系列の特性多項式を3項式から5項式以上とする提案もあるが，発生速度の大幅な犠牲を伴う上，Lewis and Payne (1973) 型の周期分割を行う際に必要な初期値設定の問題が残ろう．

ここで議論するのは，多次元一様擬似乱数を直接発生する原理とその実現法である．発生法自体は，乗算合同法とM系列発生法をその特殊ケースとして含む，一般的な方法といえよう．同様な一般的発生法として，線型結合による方法（Knuth (1981), 3.2.2）があるが，有限体の理論に基づく点は同じでも，基本的な発想と具体的な発生アルゴリズムが異なる．

2. 多次元擬似乱数の発生原理

いま n 次元単位超立方体（$n \geqq 1$）上の擬似的な一様乱数列

$$U = \{\boldsymbol{u}_0, \boldsymbol{u}_1, \boldsymbol{u}_2, \ldots\}$$

を発生させる場合を考える．すなわち，各 $\boldsymbol{u}$ は n 次元ベクトルで，

$$\boldsymbol{u} \in [0, 1]^n$$

とする．このためには，必要な有効桁数に応じて充分大きい素数 p を選び，p 法とする剰余類体を

$$\boldsymbol{Z}_p = \{0, 1, \ldots, p-1\} \approx GF(p)$$

とするとき，

$$(\boldsymbol{Z}_p)^n \approx GF(p^n)$$

で一様に分布する（あるいは一様分布にごく近い分布をする）ベクトル列

$$V=\{\boldsymbol{v}_0, \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots\}$$

を生成し，

$$\boldsymbol{u}=(1/p)\boldsymbol{v}$$

とすればよいであろう．

多項式環 $\boldsymbol{Z}_p[x]$ のある n 次既約多項式 $(n\geqq 1)$

$$g(x)=x^n-a_{n-1}x^{n-1}\cdots-a_0$$
$$=x^n+(p-a_{n-1})x^{n-1}+\cdots+(p-a_0)\in \boldsymbol{Z}_p[x]$$

に対して，$\boldsymbol{Z}_p[x]$ の $g(x)$ を法とする剰余類体 $F=\boldsymbol{Z}_p[x]/[g(x)]$ を考えると，$(\boldsymbol{Z}_p)^n$ は F に同形であり，F に属す多項式の係数からなる n 次元ベクトル（各係数は $\boldsymbol{Z}_p$ の要素）の集合として表現できる．すなわち，

$$(\boldsymbol{Z}_p)^n \ni (b_0, b_1, \cdots, b_{n-1}) \to b_{n-1}x^{n-1}+\cdots+b_0\in F$$

なる対応を考えればよい。いま，

$$T=p^n-1$$

とし，F の任意の原始元を $\xi=f(x)$ すると

$$F^*=F-\{0\}$$
$$=\{f(x), (f(x))^2, \ldots, (f(x))^T=1\}$$

であるから，多項式 $\xi=f(x)$ のべき乗列に対応して $(\boldsymbol{Z}_p)^n$ から零ベクトルを除いた集合上で一様分布する n 次元ベクトル列を生成することができ，その周期は T となる．

よって，もし $\boldsymbol{Z}_p[x]$ のある n 次既約多項式 $g(x)$ を見つけ，かつ

$$F=\boldsymbol{Z}_p[x]/[g(x)]$$

のある1個の原始元

$$\xi=f(x)$$

が見出されれば，F 上のべき計算，あるいは，適当な0でない初期値 $h_0\in F$ から

$$h_{i+1}(x)=f(x)\ h_i(x)\ \ [\mathrm{mod}\ p \text{ and } g(x)]$$

なる漸化式を用いて $h_i(x)(i=0, 1, 2, \ldots)$ の係数として目的とする

$$V=\{\boldsymbol{v}_0, \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots\}$$

を生成することができる．

素数 p を法とする乗算合同法は

$$n=1,\ g(x)=x,\ \xi=c\in \boldsymbol{Z}_p$$

の場合と考えればよい．ただし，c は $\boldsymbol{Z}_p$ の原始根である．

また，$\boldsymbol{Z}_2$ 上の原始既約3項式

$$x^n+x^m+1\quad (n>m)$$

に従うM系列発生法が

$$p=2,\ g(x)=x^n+x^{n-m}+1,\ \xi=x$$

に対応していることは，ただちに示すことができる．ただし，p が小さいので必要な有効桁にするには別な工夫（Tausworthe (1965), Lewis and Payne (1973)）が必要である．

3. 既約多項式と原始元の発見法

前節で述べた発生法を実現するためには，F の構成に必要な既約多項式 $g(x)$ および F の原始元 $\xi=f(x)$ を見つけなければならない．

まず，$Z_p[x]$ 内の n 次既約多項式を見出す問題については，Rabin (1980) が実際的な方法を提示している．簡単に述べれば，『勝手な $Z_p[x]$ 内の n 次多項式をとってくれば，それが既約である確率はほぼ 1/n であるから，(1) ランダムに n 個の整数 ($\in Z_p$) を選んで，それらを x の 0 次から $n-1$ 次までの項の係数とする n 次モニック多項式をつくり，既約性の判定を行う．(2) 既約であればそれを $g(x)$ として採用する．(3) 可約であれば，既約なものが見つかるまで (1) から繰返す．』ということである．

$Z_p[x]$ の内の n 次多項式 $g(x)$ が既約であるための必要十分条件は，

(1) $g(x)|(x^{p^n}-x)$,

(2) $l_1, \ldots, l_k$ を n の全ての因数とするとき

$$GCD[g(x), (x^{p^{l_i}}-x)]=1,\ 1\leqq i\leqq k$$

で与えられる．または，法 p のもとで多項式の因数分解を行う Berlekamp のアルゴリズム (Knuth (1981) 4.6.2 参照) の前半を利用して，因数多項式の数が 1 であるかどうかを確かめてもよい．電子計算機による数式処理システム（例えば，Hearn (1983)）を使用すれば，その判定はさほど困難ではない．

次に，原始元 $\xi=f(x)$ を見出す方法であるが，これも既約多項式をみつける場合と同じ戦略をとる．F° の要素（T だけある）のうち原始元は

$$r(T)=\varphi(T)/T=(1-1/p_1)(1-1/p_2)\cdots\cdots$$

の割合で含まれている．ここに $\varphi(m)$ はオイラーの関数であり，$p_1, p_2, \cdots$ は T を割り切る相異なる素数を表わす．$r(T)$ の値は比較的大きく，1/6 を下回ることは稀である．そこで同様に，ランダムに選んだ n 個の整数 ($\in Z_p$) により ξ をひとつ定め，それが F の原始元であるかどうか判定する．原始元でなければ，原始元が現れるまで新たな ξ を作って判定することを繰返せばよい．

原始元であるための必要十分条件は，$(n-1)$ 次以下の多項式 $f(x)$ について有限体 F における演算を行い，全ての $p_1, p_2, \cdots$ について，

$$\{f(x)\}^{T/p_i}\neq 1,\ (i=1, 2, \cdots)$$

が成立することである（原始元の定義から明らか）．この判定も，数式処理システムを用いて（F における演算を行う環境設定で），2 乗計算の繰返しにより

$$\{f(x)\}^{2^0}, \{f(x)\}^{2^1}, \cdots, \{f(x)\}^{2^i}, \cdots$$

をまず求め，次いで $f(x)$ のべきのビット・パターンに従って乗算を行っていけば，比較的容易に実行できる（なお，このテクニックの採用により，周期分割も簡単に行えることを付記しておく）．

但し，実用的な擬似乱数の発生法という観点からは，発生に要する計算が比較的簡単である必要があろう．そのためには，原始元 $f(x)$ として低次の多項式が見つかれば好都合である．実際，ξ が 1 次式

$$f(x)=cx+d$$

であれば，既約多項式 $g(x)$ を法とする演算を行うことは，

$$x^n\rightarrow a_{n-1}x^{n-1}+a_{n-2}x^{n-2}+\cdots+a_0$$

なる置換則が設定されていることと同等であるから，

$$h_i(x)=b_{i,n-1}x^{n-1}+b_{i,n-2}x^{n-2}+\cdots+b_{i,0}$$

とすれば，

$$h_{i+1}(x)=f(x)\ h_i(x)$$

は，

$$b_{i+1,j}=ca_jb_{i,n-1}+cb_{i,j-1}+db_{i,j}\ (1\leqq j\leqq n-1)$$
$$b_{i+1,0}=ca_0b_{i,n-1}+db_{i,0}$$

を法 p のもとで計算することにより求めることができる．あらかじめ定数 ca_j $(0\leqq j\leqq n-1)$ を求めておいて，j の大きい方から $b_{i+1,j}$ を計算するような手続きとすれば，1個の乱数（n 次元乱数を n 個からなる乱数列と見なすと）は3回の乗算と2回の加算および1回（場合によってはそれ以上）の剰余計算により発生できることになる．

1次式という形の元の集合に限定したときにも，上記の確率的な原始元探索を実行してみると，かなりの高い頻度で1次式の原始元が見出されることが，経験の上から言うことはできる．

発生に要する演算を減らすという観点からは，$f(x)$ が単に1次式というばかりではなく，$c=1$ あるいは $d=0$ または1であれば，乗算・加算の回数がさらに減ることになる．しかし，計算の簡単化を図ることが，生成される数列の性質に不都合な影響を及ぼす恐れもあり，必ずしも得策であるとは断言できない．

なお，1次式の原始元 $f(x)(=x$ の場合を除く$)$が見つかれば，それをもととして，$g(x)$ 以外の n 次既約多項式のひとつを簡単に見つけることができる．$\boldsymbol{Z}_p$ の拡大体 L における $g(x)=0$ の根を α とし，有限体 $GF(p^n)$ の原始元 $f(\alpha)$ の $\boldsymbol{Z}_p$ 上の最小多項式を $g_1(x)$ とすれば，$GF(p^n)$ が $\boldsymbol{Z}_p[x]/[g_1(x)]$ に同形となることを証明できるから，$g_1(x)$ は真の（n 次の項が0でない）n 次式である．よって，L 上で $f(\alpha)$ を根としてもつ n 次式

$$g_1(x)=g(f(x)) \qquad (\mathrm{mod}\ p)$$

により，新しい n 次の既約多項式が得られる．

4. 実　　例

固定小数点数（整数）を32ビット（符号のための1ビットを含む）で表現するIBM系の計算機では，法となる素数 p としては

$$2^{31}-1=2,147,483,647$$

または

$$2^{15}-19=32,749$$

などを用いるのが有利と考えられる．

ここでは，12次元（すなわち $n=12$）擬似乱数を，法 $p=32,749$ のもとで発生するひとつのアルゴリズムを導いてみよう．12次元一様性（周期に関する一様性，以下同様）が保証されているということは，12の要素をより少ない個数単位で用いる場合にも，12の約数である{1, 2, 3, 4, 6} の各次元一様性も同様に保証されていることになる．

まず，周期 T であるが，その値は

$$\begin{aligned}T&=32749^{12}-1\\&=1,521,866,364,883,767,922,745,072,354,026,397,261,801,111,691,569,982,000\\&=2^4\cdot 3^2\cdot 5^3\cdot 7\cdot 11\cdot 13\cdot 19\cdot 131\cdot 1,279\cdot 2,729\cdot 19,501\cdot 279,523\cdot 1,625,461\\&\quad\cdot 2,791,357\cdot 8,063,614\cdot 536,248,501\end{aligned}$$

というように素因数分解できる．12次元一様性からのずれに関しては，周期 T ごとに零元を挿入すれば12次元一様分布列となることから，修正すべき個数の全体の個数に対する比を誤差の大きさと考えれば，

$$(1/T)\sim 10^{-54}$$

と実用上全く無視できるオーダである．

次に，法 $p=32,749$ に関して既約な12次多項式 $g(x)$ の探索を行う．12個の係数をランダム

にきめ，それを用いてできるモニック多項式が既約かどうかの判定を行っていけばよかった．このとき，多次元での一様性が疑わしいような算術擬似乱数を用いるのは，特定の係数パターンをもつ多項式の集合のみをわたる危れがあり，いささか抵抗がある．物理的に発生した乱数が利用できれば理想的であるが，簡単には，手元にある乱数表（対数表の下1桁でもよい）から，あまり厳密な手続きを踏む必要なく，とにかく"13"個の数（$\in Z_p$）を選び，一般にモニックではない12次多項式を作ってもよいと思う．その既約性が判定されたのち，同等のモニック多項式に変形すれば済むことである．なお，1個だけ余計に選ぶのは，『厳密な手続き』を踏まないことの補償のつもりである．

数式処理のためのREDUCEシステム（Hearn（1983））には，Berlekampのアルゴリズム（Knuth（1981），4.6.2）がPFACTORIZEという関数としてすでに組み込まれているので（ただし，32,749以下の素数について），既約性の判定だけには少々もったいないが，便利に使うことができる．その結果，著者の場合は幸運にも4番目の候補で，

$$\begin{aligned} g(x)&=x^{12}+9593x^{11}+6890x^{10}+19751x^9+7371x^8+3677x^7+7642x^6\\ &\quad+9404x^5+8034x^4+4431x^3+9627x^2+15802x+22208\\ &=x^{12}-23156x^{11}-25859x^{10}-12998x^9-25378x^8-29072x^7-25107x^6\\ &\quad-23345x^5-24715x^4-28318x^3-23122x^2-16947x-10541 \end{aligned}$$

なる既約多項式を得ることができた．関数PFACTORIZEは，因数多項式としてモニックなものを結果として返すから，あらためて最高次の係数を1に変換する必要がないのも便利な点である．

既約多項式$g(x)$が見出されたので，続いてFの原始元の探索を行う．数式処理システムを使う場合には，Fにおける演算が実行できるような環境設定を行ったのち，

$$p_1=2,\ p_2=3,\ p_3=5,\ \cdots,\ p_{16}=536{,}248{,}501$$

について，原始元候補$f(x)$に関して

$$\{f(x)\}^{T/p_i}\neq 1 \qquad (i=1,\ 2\ \cdots,\ 16)$$

を確かめればよい．具体的な計算方法は前節で述べた通りである．なおREDUCEを用いる場合，有限体での演算環境は

```
LET  X**N=G1;  （代入則の設定．但し，G1=x^n−g(x)とする）
SETMOD  P;     （『法』の設定）
ON  MODULAR;   （自動的剰余計算の開始）
```

で簡単に設定され，後は普通の演算と同様の容易さで有限体での演算が行える．また，大きい数のビット・パターンを求める際，REDUCEのalgebraic modeでは整数除算および剰余関数が定義されていないので，symbolic mode（すなわちLISPレベル）の関数QUOTIENTとREMAINDER（またはDIVIDEとCARおよびCDR）を用いる．そのためには，

```
SYMBOLIC  OPERATOR  QUOTIENT, REMAINDER;
```

とすれば，以後algebraic mode（関数定義を含む）でもこれらの関数を使うことができる．当然のことながら，ビット・パターンの計算時には

```
OFF  MODULAR;  （自動的剰余計算の停止）
```

とし，異なる環境下での演算に備えて，結果を配列に保存しておく必要がある．

ここの例では，有限体Fに含まれる原始元の比率は

$$r(T)=r(32749^{12}-1)=0.198\cdots$$

で約1/5である．そのうち1次式に対応する原始元を捜してみると，再び幸運なことに

$$f(x)=733x+2353$$

を3番目の候補として得ることができた．実際，

$$\{f(x)\}^{T/2}=32748\ (=-1)$$
$$\{f(x)\}^{T/3}=28394\ (=-4355=4354^{-1})$$
$$\{f(x)\}^{T/5}=15492x^{11}+13200x^{10}+13364x^{9}+22557x^{8}+26234x^{7}+29455x^{6}$$
$$+5405x^{5}+12579x^{4}+30708x^{3}+1454x^{2}+8300x+11700$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$
$$\{f(x)\}^{T/536248501}=18664x^{11}+23235x^{10}+20012x^{9}+32508x^{8}+21233x^{7}$$
$$+28367x^{6}+6233x^{5}+13090x^{4}+30234x^{3}+25962x^{2}+30391x+23197$$

であり，前節の判定基準により，$f(x)=733x+2353$ が原始元であることがわかる．これらを，2乗，3乗，5乗，…，536248501乗して1になることを確かめてみるのもチェックのひとつとなる．

実際に擬似乱数を発生するプログラムを作成してみよう．

$$b_{i,j}\leqq 32748$$
$$ca_j\leqq 32748 \qquad (j=1, 2, \cdots, 12)$$

であるから，次の乱数値を計算する途中の値は，

$$ca_jb_{i,n-1}+733b_{i,j-1}+2353b_{i,j}\leqq 35834\times 32748<2^{31}-1$$

から分かるように，IBMタイプの計算機で表現できる整数値の範囲を越えない．それゆえ，アセンブラによる記述あるいは特別な技巧を用いなくとも，1個あたり1回の剰余計算だけで済む．以下に掲げるのはFORTRANによるプログラム例である．

```
      SUBROUTINE  RNDN (U)
      INTEGER A(12), B(12), S(12), C, D, P
      DATA  A/9366, 25725, 30324, 642, 22926, 31242,
     &          16907, 5898, 26977, 17193, 10280, 30538/
      DATA  B/7365, 21936, 4693, 9594, 30474, 16360,
     &          19619, 5128, 14280, 21523, 27147, 27633/
      DATA  C, D, K, N, P, RP/733, 2353, 1, 12, 32749, 32749.0/
      IF (K, EQ. N)GO TO 1
      B(K)=MOD (A(K)*B(1)+C*B(K+1)+D*B(K), P)
      U=B(K)/RP
      K=K+1
      RETURN
    1 B(N)=MOD(A(N)*B(1)+D*B(N), P)
      U=B(N)/RP
      K=1
      RETURN
      ENTRY SEEDN (S)
      DO 10 I=1, N
   10 B(I)=S(I)
      K=1
      RETURN
      END
```

このサブルーチンを12回コールすることにより，12次元一様ランダム・ベクトルが得られる．もちろん，その要素単位あるいは数個ずつの組として使用しても構わない．プログラム中のA(J)が ca_{n-j} に，B(J)が $b_{i,n-j}$ に，それぞれ対応しているは直ちに分かるであろう．

ENTRY ポイント SEEDN は，乱数列の初期値を本体部の DATA 文で用意した以外の設定とする場合に用いる．

5. 今後の問題

多次元一様擬似乱数の発生法に関して，その原理と具体的な実現法を述べてきた．ただし，ここで取扱ったのは乱数列の1周期に亘る多次元一様性のみで，実際の使用形態に応じた理論的・実験的評価は全く残されている．

周期に亘る性質だけを考える上では全て同等であった既約多項式および原始元の組合わせにも，部分列の統計的性質の良し悪しによって，優劣がついてくることになる．『この既約多項式とこの原始元の組合わせを用いれば，性質の良い乱数列が得られる』と自信をもって言うことができるようになるまでには，もう少し経験を積まねばならない．

その際，考えるべきことのひとつに，発生アルゴリズムの簡素化の程度がある．乗算合同法およびM系列に基づく発生法は，これまで述べた発生法の最も簡単な形態といえる．その簡単さゆえに，非常に効率よく発生できるようになっていることは確かであるが，反面，『真の乱数列』の代替列としての性質が犠牲にされていることも事実であろう．『発生効率』と『乱数としての性質』とのバランスを考慮した，『適度な複雑さ』を求めていく必要が痛感される．

既約多項式および原始元の判定は，現在のところ，計算機との頻繁な会話を行いながら実行しているが，数式処理にあまり馴染みのない研究者でも容易に判定ができるように，REDUCE などの数式処理システムに必要な手続きを組込むことも考えている．数式処理システムを用いなくとも判定計算が可能ではあるが，関連する計算の多くを全く別に行う必要があることを考慮すると，FORTRAN などの数値計算用の汎用言語で手続きを記述するのは得策ではないように思う．

謝　　辞

適確なコメントをいただいた査読者の方々に感謝いたします．

参 考 文 献

伏見正則・手塚　集（1981）．多次元分布が一様な擬似乱数列の生成法，応用統計学，**10**，151-163.

Hearn, A.C. (ed.) (1983). *REDUCE user's manual,* Rand Corp., Santa Monica.

Knuth, D.E. (1981). *The art of computer programming,* Vol. **2** (2nd ed), Reading, Massachusettes, Addison-Wesley.

栗田良春(1983)．M 系列の L-tuple の weight distribution の偏りについて，数理解析研講究録 498, 153-171.

Lewis, T.G. and Payne, W.H. (1973). Generalized feedback shift register pseudorandom number algorithm, *J. ACM,* **20**, 456-468.

Lindholm, J.H. (1968). An analysis of the pseudo-randomness properties of subsequences of long m-sequences, *IEEE trans. Informa. Th.,* IT-**14**, 569-576.

Niederreiter, H. (1978a). The serial test for linear congruential pseudo-random numbers, *Bull. Amer. Math. Soc.,* **84**, 273-274.

Niederreiter, H. (1978b). Quasi-Monte Carlo methods and pseudo-random numbers, *Bull. Amer. Math, Soc.,* **84**, 957-1041.

Niki, N. (1979). Multi-folding the normal distribution and mutual transformation between uniform and normal random variables., *Ann. Inst. Statist. Math.,* **31**, 125-140.

仁木直人 (1983). パーソナル・コンピュータのための物理乱数発生器, 統数研彙報, **31**, 33-49.
Rabin, M.O. (1980). Probabilistic algorithms in finite fields, *SIAM J. Comput.*, **9**, 273-280.
Tausworthe, R.C. (1965). Random numbers generated by linear recurrence modulo two, *Math. Comput.*, **19**, 201-209.
Walker, A.J. (1974). New fast method for generating discrete random numbers with arbitrary frequancy distributions, *Electr. Letters*, **10**, 127-128.

Finite Field Arithmetics and Multi-dimensional Uniform Pseudo-random Numbers

Naoto Niki

(The Institute of Statistical Mathematics)

The main purpose of this article is to give, from finite field arithmetics, a general method for generating a sequence of n-dimensional vectors uniformly distributed in the n-dimensional unit cube with a very long period. In Monte-Carlo simulations we actually need a huge amount of multi-dimensional uniform random numbers. Furthermome, if a sequence of vectors uniformally distributed in the n-dimensional unit cube, the sequence of elements of the vectors may be expected to have good serial "independence".

For a given prime p and an integer $n \geqq 1$, we can find by probabilistic algorithms an irreducible polynomial $g(x)$ over $\boldsymbol{Z}_p$ of degree n and a primitive element $f(x)$ in the finite field $\mathrm{F}=\boldsymbol{Z}_p\,[x]/[g(x)]$. Each element $h_i(x)$ $(i=0, 1, 2\cdots, T-1;\ T=p^n-1)$ except 0 in F is generated by the recurrence formula

$$h_{i+1}(x)=f(x)\ h_i(x) \qquad (\mathrm{mod}\ p \text{ and } \mathrm{mod}\ g(x))$$

with an arbitrary non-zero polynomial $h_0(x)$ in F.

The above recurrence formula itself has a period of length $T=p^n-1$ and the n-dimensional vectors $\boldsymbol{v}_i(i=0, 1, 2\cdots)$ of which components comprise the coeficients in $h_i(x)$ are uniformly distributed over $(\boldsymbol{Z}_p)^n-\{\boldsymbol{0}\}$. Therefore, if the prime p is large, the sequence of vectors

$$((1/p)\boldsymbol{v}_0,\ (1/p)\boldsymbol{v}_1,\ (1/p)\boldsymbol{v}_2,\ \cdots)$$

may be usable as a sequence of "random" vectors in the n-dimensional unit cube.

Using a computer algebra system, REDUCE, an experiment was made to find an irreducible polynomial and a "linear" primitive element for $p=32749$ and $n=12$. The process and the results are shown in Section 4. A prototype of FORTRAN programmes for our generation method is also exhibited.