パーソナルコンピュータによる疑似乱数の発生とその検定

村　田　茂　昭

はじめに

現在のパソコン〈パーソナルコンピュータ〉の隆盛は，非数値計算分野（ワープロ，データベース等）への応用をぬきにしては考えられない。つまり，パソコンにおいては，相対的に，数値計算の比重は低下したと考えられる。しかし，現在のパソコンは，数値計算の分野でも充分に強力である。

さらに，近年，FORTRAN 77に準拠したコンパイラが，出現した。説明書を，額面通りに，受け取れば，大型コンピュータで動作したプログラムを，そのまま，パソコン上で使用できることになる。

本論文は，ごくありふれたパソコンによって，確率的問題を扱うための，基礎的検討，試行を行なった結果の報告である。

使用したパソコンは，NEC PC9801RX2（16ビット），コンパイラは，LIFE BOAT 社の Pro FORTRAN77　Ver. 1．27 である。

1．乱数と疑似乱数

乱数（乱数列）とは，ある指定された確率分布を持つ数列のことであり，システムシミュレーションや，システムアイデンティフィケーション等に利用される。このうち，シミュレーション法は，モンテカルロ法と呼ばれ，コンピュータの確率論的応用分野として重要である。システムアイデンティフィケーションにおいては，乱数列は，不規則信号として利用される。

実は，コンピュータで発生できるものは，因果関係の定まった数列であり，有限の周期を持っている。したがって，これは，疑似乱数と呼ばれる。しかし，その周期が充分に長く，かつ，見かけ上の統計的性質が良好であれば，前記の応用計算において有効である。本論文においては，簡単のため，以後，疑似乱数を，単に，乱数と呼ぶ。また，物理乱数（後出）と区別するときは，算術乱数と呼ぶ。

コンピュータの発生する乱数の基本は，見かけ上一様な確率で出現する，非負の整数の列である。整数列の最大数をX_Mとするとき，X_M（またはX_M+1）で，この整数列を割ってやると，区間0～1に分布する，いわゆる，実数型の一様乱数が得られる。（X_M+1で割ったときは，1を含まない）

連続型の確率分布（正規分布，ポアッソン分布等）をする乱数は，この区間0～1に分布する一様乱数から，数値計算によって求められることが多い。本論文においては，この，一様乱数について論ずる。

一様な整数乱数列を発生するためには，現在有効と思われる方法がふたつある。ひとつは，Lehmer に，源を発する線形合同法[1], [2], [3], [4]であり，もうひとつは，M 系列法[5]である。本論文においては，数値計算法上単純な前者についてのみ論ずる。

乱数発生法の，満たすべき条件としては，一般に下記の四つがあげられる。[6]

(1) 高速であること。
(2) 乱数に，もし周期があるならば，その周期は，充分に長いものであること。
(3) 再現性があること。
(4) 発生した乱数は，良好な統計的性質を持つこと。

このうち，(2)と(3)は，物理乱数[7]（電子管の熱雑音や，放射性物質の崩壊雑音等を利用した乱数）を考慮してあげられたものと思われる。物理乱数には，一般に周期はなく，再現性もないが，乱数源をコンピュータシステムに組み込むとオンラインで計算できる利点がある。ただし，物理乱数を，磁気テープ等に記憶しておけば，再現性は得られる。もっぱら，ハードウェアの問題となるので，ここでは，物理乱数に関してはこれ以上は論じない。

前出の算術乱数という表現は，ソフトウェア的に発生する乱数を，物理乱数と区別するためのものである。算術乱数は，現在の発生法によっては，必ず周期を持つ。また，再現性があるのは当然である。現在のディジタルコンピュータによる数値計算で，再現性のない計算法は正しい計算法とはいえない。

2．線形合同法乱数

線形合同法とは，Lehmerが，1948年頃発表した方法であり，次のような漸化式を用いて，非負の整数列$<X_n>$を発生する。

$$X_n = a X_{n-1} + c \quad (\mathrm{mod}\ M) \qquad (2.1)$$

$c = 0$の場合を乗算型合同法，$c \neq 0$の場合を混合合同法と呼ぶ。

区間（0～1）上（0は，含む場合と含まない場合がある。1は，含まない）の実数型乱数$<x_n>$が必要な場合は，

$$x_n = X_n / M \qquad (2.2)$$

を用いる。ただし，Mは，法（modulus）である。

式（2.1）によって発生する整数列の周期は，原理的にMより長くなり得ない。従って，Mは，充分に大きくとる必要がある。周期がMの場合，この整数列は0～M－1の整数のすべてを網羅する。（ただし乗算式合同法の場合は，0は含まない。すなわち，周期は，M－1が最大である。）周期が，M（乗算式合同法の場合はM－1）でない場合は，一般には，初期値X_0（種（たね）と呼ぶ）によって周期が異なる。X_0が特定の条件を満たすとき，周期が最大となるが，これを最長周期と呼ぶ。

法M，乗数a（$0<a<M$），および加数c（$0<c<M$）は，$<X_n>$の周期がなるべく長く，かつ$<X_n>$が充分ランダムと見なせるように選ぶ必要がある。

特殊な場合についての定理は，いろいろあるが，一般的な定理は，次のふたつである。[8]

定理1．

（2.1）式によって生成される$<X_n>$が，最長周期Mを持つ（周期が，法Mに等しくなる）ための必要充分条件は，次の三条件がすべて成り立つことである。

i）cがMと互いに素である。
ii）$b = a - 1$が，Mを割り切るすべての素数の倍数である。
iii）Mが，4の倍数であれば，bも4の倍数である。

この定理は，$c = 0$の乗算合同法については何も言っていない。上記ii），iii）を計算することは，Mが，比較的少数の素数のべき乗であるとき，簡単になる。（たとえば，kを整数として，$M = 2^k$の場合，2のみを考慮すればよい。）

定理2． 乗算型合同法数列の周期に関しては，次のことが成り立つ。

i）$M = 2^k$（$k \geq 4$）の場合，可能な最長周期はM／4であり，これを達成できる

のは，a（mod 8）＝3，または5で，X_0が奇数のときに限られる。

ii）M が2よりも大きい素数pに等しい場合可能な最長周期はp－1であり，これを達成できるのは，$X_0 \neq 0$で，かつ，p－1の任意の素因数qに対して

$$a^{(p-1)/q} \neq 1 \pmod{p} \quad (2.3)$$

が成り立つ場合に限られる。

この定理i）は使用し易いが，ii）は，大変な手間がかかる。実用的な例では，pは，非常に大きな素数であり，

$$a^{(p-1)/q}$$

は，途方もない大きい数となる。また，qも多数存在することが多いので，この種の解析を多数回繰り返しなければならない。

3．ソフトウェアシステムによる制約

コンピュータによる乱数発生法を調べると，かつて，プログラム作成に，機械語ないしアセンブラ言語を用いた頃の名残がみられる。例えば，「2進法の計算機であれば，法を2^kにとると，乗算，加算の際の桁あふれを無視するだけで合同計算ができるので都合がよい。」etc.

演算の際に，オーバフロウした場合の処置は，FORTRAN 77の言語仕様には含まれていない。使用したコンパイラの説明書等[9], [10]を読んでみても判然としない。そこで，二三計算をしてみた結果，このコンピュータシステムにおいては，次のようになっていることが判明した。

(1) 整数型変数は，最大で一語4バイト（32ビット）で表現される。負数は2の補数表示である。

(2) 加算は，32ビット（31ビットではない）の整数として扱われ，オーバフロウした分は単に捨て去られる。ただし，結果が2^{31}以上になると（補数表示のため）以後の計算では，負数として扱われる。

(3) 2，4，8の乗算は，オーバーフロウに関しては，加算と同様に扱われる。それ以外の乗算で，結果が$2^{31}-1$をこえた場合は，実行時エラーとなる。エラーの際ランタイムサポートプログラムが，演算を続行するかどうか聞いてくるが，続行したときの結果は，何も保証されない。（結果の，下位ビットを保存してくれると都合がよいのであるが，そのようなことはない。）

以上の結論として，整数の乗算は，常にオーバーフロウによる実行時エラーの恐れがある。これを避けるためには，整数を上位16ビットと，下位16ビットに分けて計算することが考えられる。さらにいうと，16ビットの整数どうしの乗算においても，結果が32ビットの整数となり得るので，これも実行時エラーをひきおこす可能性がある。これについては，何か特別なしかけを考えるか，一歩退いて，上位15ビット下位15ビット合計30ビットで我慢しなければならない。乗数aは一般には，非常に大きい場合もあり得るから，aとX_{n-1}の両方についてこのような扱いが必要である。しかし，今回は，第一着手であるので，この方面には立ち入らなかった。

4．乱数発生プログラム

今回は，次の三つの乱数発生プログラムにつき検討した。

4．1　乱数1…半語〈16 ビット 〉乱数

$$X_n = aX_{n-1} + c \pmod{M} \quad (4.1)$$

において

$$\left.\begin{array}{l} a = 12869\ (17 \times 757) \\ c = 6925\ (5^2 \times 277) \\ M = 32768\ (2^{15}) \end{array}\right\} \quad (4.2)$$

とする。[11]（付録Ⅰ参照 〉

これは，定理1により最長周期 M をもつ。すなわち，M が，4の倍数である場合に当てはまる。（b＝12868＝2^2×3217）X_{n-1}は，最大16ビットで，aもcも，2進法で表現したとき15ビットよりも小さいから（4.1）式の演算において，オーバーフロウする事はない。

実数型の乱数＜x_n＞は，

$$x_n = X_n / M \quad (4.3)$$

で求める。きわめて，周期の短い乱数であるが，一応検討した。なお，簡単なプログラムで周期を確認した。（所要時間は8秒であった。）

4．2 乱数2

（2.1）式において

$$\left.\begin{array}{l} a = 1229 \quad (\text{素数}) \\ c = 351750 \quad (2 \times 3 \times 5^3 \times 7 \times 67) \\ M = 1664501 \quad (\text{素数}) \end{array}\right\} \quad (4.4)$$

とする。[12]（付録Ⅱ参照 〉

X の最大値は，1664500であるから，

$$MAX(aX_{n-1}+c) = 2046022250 < 2147483647 = 2^{31}-1 \quad (4.5)$$

であって，これも，整数演算中にオーバーフロウするおそれはない。

周期は，このプログラムの著者は（プログラム中のコメント文中で）M（＝ 1664501）であると明記している。しかし，今回簡単なプログラムで，周期を測定したところ，832250（（M －1）／2）であった。（所要時間 6分27秒 〉式の形から考えて，周期 832250と 832251のふたつの数列があると考えられるが，後者は確認できていない。

なお，定理1において，法 M が素数の場合は何といっいない。この定理に対する誤解が，こういう結果になったのではないかと思う。この乱数については，種種のテストが行なわれたはずであるが，周期は確認しなかったらしい。

4．3 乱数3… システム内蔵関数

Pro FORTRAN 77には，単精度の一様乱数発生関数 RANDOM が内蔵されている。

X ＝RANDOM（IR）

としてIR に種（第2章のX_0）を入れると，X に，区間（0～1）の実数型乱数が得られる。IR が0以下の場合は，乱数列の次の乱数が得られる。

この乱数の発生機構が，マニュアルに明記されていないので，数値実験的に解明した。まず，このプログラムは，アセンブラ言語等によって書かれ，オーバーフロウの処理がきちんと行われているものと推定した。FORTRAN のプログラムで

X 1 ＝RANDOM(1)

X 2 ＝RANDOM(2)

として，（X 2 －X 1）＊ 2^{31}，X 1 ＊ 2^{31}を計算したところ，いずれも8189となった。

これは，（2.1）式において

$$a = 8189\ (=19 \times 431)$$

$$c = 0 \quad (4.6)$$

とした場合に対応する。すなわち，乗算合同法による乱数である。しかし，法（modulus）が，2^{31}か，$2^{31}-1$（素数）かは，単精度演算ではわからない。(RANDOM自身は単精度であるが，(2.2)式の右辺の計算を，最下位ビットまで確認するには，倍精度演算が必要である。)これを確認するため，簡単な数値計算を行なった（付録Ⅲ，Ⅳ参照）なお，$2^{31}=2147483648$である。

$$2^{31}/8189 = 262240.0352$$
$$(2^{31}-1)/8189 = 262240.0350 \qquad (4.7)$$

であるから，

$$X_n = a X_{n-1} (\mathrm{mod}\ M)$$
$$x_n = X_n / M \qquad (4.8)$$

と仮定したとき，合同演算が行なわれる境目は，IR＝262240とIR＝262241である。
すなわち，前者では，そのまま乗算されるが，後者では，乗算後，Mが差し引かれる。予備計算によって，実数型に直すときの除算が，法Mでなく，M＋1である可能性がでてきたのでこれも検討することとした。結果は，付録Ⅳに示すように，付録Ⅲのプログラム中の変数でいうと，Z2とZ3Dがゼロとなった。

Z2がゼロとなることは，整数列から，実数型の乱数$<x_n>$を求める際

$$x_n = X_n / 2^{31} \qquad (4.9)$$

としていることを示す。Z3Dがゼロになることは，IR＝262241の時に乗算後$2^{31}-1$が引かれていることを示す。すなわち，法Mは，$2^{31}-1$である。したがって，この内蔵関数では，(4.8)式のかわりに

$$x_n = X_n / (M+1) \qquad (4.10)$$

と計算している。これは，理論的には，若干問題であると思うが，この方が計算速度は，早いのであろう。

前述のように，c＝0とした場合は，Mを$2^{31}-1$（素数）としても，最長周期が，Mになることは保証されない。しかし，数値計算によれば，この内蔵関数を使用して，50000個の乱数を発生するのに，9秒を要した。これによれば，10^8個の乱数を発生するのに，5時間かかる計算になる。実験によれば，約25時間計算を行なったが，同じ乱数を得られなかった。したがって，この乱数列の周期は，5×10^8以上と推定される。

5．乱数列の検定

5．1　等出現性の検定

区間（0～1）の乱数を，幅0.1で，出現頻度を数えた。実際の計算は

$$L = [10X] \qquad (5.1)$$

とした。ここで，[ ]は，ガウスの記号である。発生した乱数から，整数Lを計算し，0～9の各々について，出現回数を積算した。乱数の総数をNとするとき，乱数が完全にランダムであるときの理論度数をN_Tとすると

$$N_T = N/10 \qquad (5.2)$$

である。実際の出現同数をN_Lとして

$$S = \sum_{L=0}^{9} \frac{(N_L - N_T)^2}{N_T} \qquad (5.3)$$

を計算した。帰無仮説を，「N個の乱数が，区間（0，1）の中に等確率で現れる。」とすると，このSのとる値は，N_Tがある程度大きいとき，自由度9の，χ^2分布をすることが知られている。[13] 有意水準5％の棄却域は，χ^2分布表より，区間（16.92, ∞）である。[14], [15]

5．2 無規則性の検定

n＋k個の

$$x_1,\ x_2,\ x_3,\ \cdots,\ x_n,\ \cdots,\ x_{n+k}$$

においてk個ずらせた数の組

$$(x_1,\ x_{1+k}),\ (x_2,\ x_{2+k}),\ \cdots,\ (x_n,\ x_{n+k})$$

における相関係数$\rho^{(k)}$は，次式で与えられる。

$$\rho^{(k)}=\frac{\frac{1}{n}\sum_{i=1}^{n}x_i x_{i+k}-\bar{x}^2}{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^2}$$

$$=\frac{\frac{1}{n}\sum_{i=1}^{n}x_i x_{i+k}-\bar{x}^2}{\frac{1}{n}\sum_{i=1}^{n}x_i^2-\bar{x}^2} \qquad (5.4)$$

ここで

$$\bar{x}=\frac{1}{n}\sum_{i=1}^{n}x_k \qquad (5.5)$$

である。

一方，区間（0，1）上の乱数について，帰無仮説を，「乱数が無規則に並んでいる。」とする。この帰無仮説のもとで

$$Z^{(k)}=\frac{\sqrt{n}\ \rho^{(k)}}{\sqrt{13}} \qquad (5.6)$$

は，Nが充分大きいとき，平均値0，分散が，1の正規分布にしたがうことが知られている。[15], [16] すなわち，$Z^{(K)}$は，正規化した相関係数となる。

有意水準を，95%とすると

$$f(x)=\frac{1}{\sqrt{2\pi}}\int_0^x e^{-t^2/2}\,dt \qquad (5.7)$$

として

$$f\ (1.96)\fallingdotseq 0.4750 \qquad (5.8)$$

により，[17] 棄却域は，次のふたつの区間となる。

$$(-\infty,\ -1.96),\ (1.96,\ \infty)$$

5．3 数値計算例

この論文の範囲では，発生した乱数は，配列中に記憶しておき，それから各種の統計量を計算している。この程度のプログラムでは，配列の総量としては，単精度実数型（4バイト）換算で，90000個の領域をとることができる。（コンピュータのRAMは，640kバイトである。）各々の乱数発生プログラムについて計算した例を表1，表2に示す。ＩＲは，乱数の種（たね）の値である。

なお，10000個の乱数の発生時間は，乱数1と乱数3は2秒，乱数2は3秒であった。また，統計量計算時間は，同じく10000個について，乱数1と乱数3は27秒，乱数2は44秒であった。同じ計算式であるのに，乱数2に関して，時間がかかるのは，乱数列を，倍精度の配列に格納しているからである。（付録Ⅱ参照）念のため，まったく同じコンピュータで，数値データプロセッ

I R＝137

N＼L	0	1	2	3	4	5	6	7	8	9	S
1000	92	109	107	89	107	113	96	86	100	101	7.46
10000	976	1037	975	1006	990	1032	937	988	1025	1034	9.62

(1)　乱数 1

I R＝137

N＼L	0	1	2	3	4	5	6	7	8	9	S
1000	110	92	97	112	88	101	101	81	104	114	10.36
10000	1023	1048	1004	1025	904	968	1005	1002	1023	998	14.28

(2)　乱数 2

I R＝137

N＼L	0	1	2	3	4	5	6	7	8	9	S
1000	91	94	98	108	100	118	88	103	105	95	7.12
10000	1020	972	937	1053	1036	1032	993	962	1024	971	13.19

(3)　乱数 3

表 1　等出現性の検定（N_LとS）

N＝10000
I R＝137

k	1	2	3	4	5	6	7	8
$\rho^{(K)}$	0.003	0.014	−0.004	−0.014	0.003	0.008	−0.010	0.002
$Z^{(K)}$	0.092	0.385	−0.102	−0.394	0.084	0.233	−0.290	0.049

(1)　乱数 1

N＝10000
I R＝137

k	1	2	3	4	5	6	7	8
$\rho^{(K)}$	0.012	−0.016	0.012	−0.001	−0.017	−0.010	0.030	−0.006
$Z^{(K)}$	0.334	−0.454	0.324	−0.038	−0.473	−0.291	0.822	−0.157

(2)　乱数 2

N＝10000
I R＝137

k	1	2	3	4	5	6	7	8
$\rho^{(K)}$	−0.008	−0.007	0.015	0.005	0.003	0.011	−0.005	−0.002
$Z^{(K)}$	−0.215	−0.182	0.424	0.126	0.090	0.301	−0.138	−0.061

(3)　乱数 3

表 2　無規則性の検定（相関係数）

サ（80287）を増設したシステムで計算実行したところ，諸統計量計算時間は，乱数１と乱数３は27秒，乱数２は28秒であった。すなわち，倍精度演算のスピードアップには，数値データプロセッサの増設が有効である。

各々の計算例を見ると，等出現性の検定（χ^2検定）は，いずれも合格している（$S<16.92$)。すなわち，有意水準５％で，「乱数は区間（０，１）の中に等確率で現われる。」と言う帰無仮説は成立する。周期の，きわめて短い乱数１は，$N=10000$のとき，Ｓが一番小さい。これは，$N=32768$（周期）とすると，$S=0$となるので，当然である。（（５.１）式の計算が整数どうしの計算であるので，数値計算では，完全にゼロとはならないが）

無規則性の検定においても，すべてのｋについて$|Z^{(k)}|<1.96$となり，各々の乱数について，有意水準95％で「乱数が無規則に並んでいる。」という帰無仮説が成立する。

６．まとめ

三つの既成の乱数について，古典的検定を行ったが，何れも合格であった。これらは，以前から使用されている乱数であるので，その統計的性質が，極端に悪いとは考えられず，当然と言える。

乱数２については，その周期が，従来信じられていた値の半分であることが確かめられた。

また，Pro FORTRAN 77のシステム内蔵関数による，乱数３については，その発生数式を明らかにすることができた。

さらに，FORTRAN 77で書かれた，大型コンピュータ向けのソフトウェアが，そのまま，パソコン上を走ることが確認された。計算機言語の趣旨から言うと，これは当然のことに見えるが，著者は，重大なことと考える。すなわち，時代の趨勢が，大型コンピュータの，TSS 利用から，パソコンの使用へと流れていると感ずるからである。（勿論，コンピュータネットワークのことを考慮にいれている。）また，パソコンには，CPU Time を余り気にせず計算できるというメリットがある。

乱数発生法の文献を読むと，何れも，発生速度を非常に重視している。しかし，今回の，簡単な統計計算においても，何れも，その計算時間が，乱数発生時間の10倍以上となっている。従って，余り，曲芸的な方法で，乱数発生の速度を上げることは無意味である。著者は，それは，機械語やアセンブラ言語でプログラムを書いた時代の名残のように思われる。本論文の一つの結論として，「良い乱数発生法であれば，多少時間がかかっても良い。」ことをあげておく。

今後の方向としては，

(1) 整数計算のオーバーフロウに対する対策をたて，乱数発生プログラムの自由度を増す。
(2) 乱数の近代的検定法につき検討する。
(3) 不規則信号としての乱数の応用につき検討する。

等が考えられるが，(3)に重点をおきたいと考えている。

［結言］この研究は，平成 2年度札幌大学個人研究助成費によるものであり，その関係者に厚く感謝いたします。

参考文献

(1) 伏見政則「乱数」東京大学出版会　P 1
(2) 津田孝夫「モンテカルロ法とシミュレーション」改定版　培風館　P11
(3) 宮武修，脇本和晶　「乱数とモンテカルロ法」　森北出版　P 4
(4) 脇本和晶　「乱数の知識」　森北出版　P26
(5) 伏見　P12
(6) 津田　P 9
(7) 宮武，脇本　P16
(8) 伏見　P 2
(9) 「Pro FORTRAN－77 Compiler user's manual」LIFEBOAT社
(10) 山本和明「パーソナル　FORTRAN－77　トレーニングマニュアル」ラジオ技術社
(11) 浦昭二「FORTRAN－77入門」培風館　P152
(12) 渡辺力，名取享，小国力「FORTRAN－77による数値計算ソフトウェア」P316　（プログラムは，フロッピーディスクで供給されている。）
(13) 脇本　P41
(14) たとえば薩摩順吉「確率，統計」岩波書店　P214
(15) 脇本　P53
(16) 津田　P30
(17) 脇本　P142

C 乱数 半語

```
      FUNCTION URANH (I)
      INTEGER L,C
      REAL MU
      PARAMETER (L=12869, C=6925, MU=2.0**15)
      I=MOD (L*I+C, 2**15)
      URANH=REAL (I)/MU
      RETURN
      END
```

付録I 乱数1のプログラム

```
      SUBROUTINE URAND1 (N, X, IR) 〉
*********************************************************
*  UNIFORM RANDOM NUMBER GENERATOR (MIXED CONGRUENTIAL METHOD)   *
*        PORTABLE BUT SLOW. THE PERIOD IS ONLY 1664501.          *
*  PARAMETERS                                                    *
*    (1)  N     (I) THE NUMBER OF RANDOM NUMBERS TO BE GENERATED *
*                   (INPUT)                                      *
*    (2)  X     (D) UNIFORM RANDOM NUMBERS (OUTPUT)              *
*    (3)  IR    (I) THE INITIAL SEED (INPUT)                     *
*                   THE SEED FOR THE NEXT CALL (OUTPUT)          *
*  COPYRIGHT : Y. OYANAGI, JUNE 30, 1989 V. 1                    *
*********************************************************
*
      DOUBLE PRECISION X(N) , INVM
      PARAMETER (M =1664501, LAMBDA =1229,MU =351750)
      PARAMETER (INVM=1.0D0/M)
*PARAMETER CHECK
      IF (N. LE. 0) THEN
        WRITE (6,*) '  (SUBR.URAND1) PARMETER ERROR. N   =  ',N
        WRITE (6,*) 'RETURN WITH NO FURTHER CALCULATION.'
        RETURN
      END IF
      IF (IR  .LT.  0  .OR.  IR  .GE.  M) THEN
      WRITE (6,*) '(SUBR.URAND1) WARNING. IR  =  ',IR
      END IF
*MAIN LOOP
      DO 10  I = 1 , N
        IR=MOD (LAMBDA*IR +MU,M)
        X (I) =IR*INVM
 10   CONTINUE
      RETURN
      END
```

付録Ⅱ　乱数2のプログラム

```
C  システム内蔵乱数発生プログラムのテスト
      DOUBLE PRECISION P1, P2, X1, X2, R1, R2, R3D, R4, R4D
      PARAMETER (P1=2147483647. D0, P2=2147483648.D0)
   10 PRINT *
      PRINT *,'I ='
      READ *, I
      IF (I   .LT. 0) STOP
      RS=RANDOM (I)
      X1=DBLE (I)*8189.D0
      IF (X1 .LT.  P2) THEN
        R1=X1/P1
        R2=X1/P2
        Z1=RS-R1
        Z2=RS-R2
        PRINT *,'RS =', RS
        PRINT *, Z1, Z2
      ELSE
        W3=X1-P1
        W4=X1-P2
        R3=W3/P1
        R3D =W3/P2
        R4=W4/P1
        R4D =W4/P2
        Z3=RS-R3
        Z3D =RS-R3D
        Z4=RS-R4
        Z4D =RS- R4D
        PRINT *,'RS =', RS
        PRINT *, Z3, Z3D, Z4, Z4D
      END IF
      GO TO 10
      END
```

付録III　システム内蔵乱数発生プログラムのテスト

I＝1
RS＝　3.8133004E－06
－1.7757062E－15　0.0000000E＋00

I＝262240
RS＝　9.9999988E－01
1.4435499E－08　1.4901161E－08

I＝262241
RS＝　3.6796554E－06
－1.7134730E－15　0.0000000E＋00　4.6565955E－10　4.6566128E－10

I＝－1

STOP

付録Ⅳ　テストプログラムの入出力画面（下線は入力を示す）

付録V　使用したコンピュータシステムの概要

ハードウェア　コンピュータ　NEC PC9801 RX2　クロック　12MHz（80286 ）
RAM 640KB
増設 RAM 2MB
内蔵フロッピーディスク装置　1 MB× 2
増設ハードディスク装置　40MB

ソフトウェア　オペレーティングシステム　MS－DOS Ver. 3.3C　Micro Soft 社
エディタ　VZ editer Ver.1.5 （兵藤嘉彦 ）
コンパイラ　Pro FORTRAN 77 Ver. 1 .27　LIFE BOAT 社