研究ノート

一様乱数のプログラム

木　村　　等
井　上　知　子

I　はじめに

香川大学計算センターの「FACOM 230—45 S 科学用サブルーチン・ライブラリー（SSL）使用方法解説書」には，乱数に対して implicit に一様乱数を用いていても，表面上は正規乱数とポアソン乱数の CALL の方法の説明があるだけで他のものは見当たらない。しかし，シミュレーション等に使われる乱数の基本は一様乱数で，他の乱数は一様乱数から作られるのが普通なので，ここでは単精度と倍精度の一様乱数発生のプログラムを紹介し，なぜそれで良いのかを解説する。たいていのシミュレーション関係の教科書は乱数について少し触れてあるだけなので，計算機との関係を少し詳しく書いてみたい。ただし，次の節以降で説明するプログラムは FACOM 230—45 S 用であり，他の計算機にかける時は少し変更を要する事もある。

II　一様乱数発生のプログラムとその使用例

単精度の場合のプログラム

```
0001            FUNCTION  RANUNI(IRAND)
0002            IRAND=IRAND*5+6917
0003            IF(IRAND) 20,30,30
0004         20 IRAND=IRAND+32768
0005         30 RANUNI=IRAND/32768.0
0006            RETURN
0007            END
```

倍精度の場合のプログラム

```
0001            DOUBLE PRECISION EUNCTION DRANYU(IDD)
0002            DOUBLE INTEGER IDD
0003            IDD=IDD*5+453816811
```

```
0004            IF(IDD) 20,30,30
0005         20 IDD=IDD+2147483647+1
0006         30 DRANYU=IDD/2147483648.0
0007            RETURN
0008            END
```

単精度，倍精度いずれの場合も上記の通りFUNCTION タイプである。入力パラメーターとして一様乱数を作る時必要な初期値を与えて呼び出すと，それをもとにして乱数を1個発生する。2度目の呼び出しからは，最初に呼び出した時の入力パラメーターが FUNCTION 内で値の変更を受けて次の乱数発生に必要な初期値となる。従って一度初期値を設定した後は繰り返し呼び出すことにより，一連の一様乱数列が得られる。関数名 RANUNI と DRANYU に〔0, 1〕区間の一様乱数が，パラメーター IRAND または IDD に整数の一様乱数が発生させられると考えて良い。

倍精度の場合について，その使用例と結果の一部を載せておく。

```
0001            DOUBLE INTEGER II
0002            DOUBLE PRECISION RANSU, DRANYU
0003            WRITE(6, 10)
0004         10 FORMAT ('SEISU RANSUU' ,10X, '(0,1)RANSUU'/)
0005            II=1
0006            DO 100 J=1,200
0007            RANSU=DRANYU (II)
0008            WRITE(6,20) II, RANSU
0909         20 FORMAT(1H0, I11, 10X, D25.17)
0010        100 CONTINUE
0011            STOP
0012            END
```

```
SEISU RANSUU        (0,1) RANSUU
  453816816          0.21132492274045940D+00
  575417243          0.26794953411445020D+00
 1183419378          0.55107259098440410D+00
 2075946405          0.96668787533417340D+00
   96130596          0.44764297083020210D-01
  934469791          0.43514640582725410D+00
  831198470          0.38705694954842330D+00
  314841865          0.14660966815426950D+00
 2028026136          0.94437326118350030D+00
 2004012899          0.93319122632965450D+00
 1883946714          0.87728105206042530D+00
 1283615789          0.59773018071427940D+00
  429444812          0.19997582398355010D+00
  453557223          0.21120404032990340D+00
```

なお，前記 FUNCTION 内で用いている定数には後で説明するようにそれぞれ意味があり，理論的裏付けのない変更はしない方が良い。単精度の場合のプログラムを用いて，¥ JOB のコントロール・カードに DOUBLE，IDOUBLE の指定をすれば，乱数も長い周期のものを発生する事ができるように考えている人もいるようであるが，この場合は倍精度用の関数 DRANYU を用いなければならない。なぜなら，DOUBLE，IDOUBLE の指定をしても 32768.0 を自動的に 2147483648.0 にコンパイルしてくれることはあり得ないからである。

III　プログラムの理論的根拠—混合型合同法

数列のうち，一様乱数列というのは

（i）　周期のないこと

（ii）　各数値が等確率であらわれること（等確率性）

（iii）　無規則に数値が並ぶこと（独立性または無相関性）

が保証されているものをいう。数式を根拠にコンピューターで生成する乱数列の系列は，ある間隔で必ず周期があらわれるので，これを擬似乱数列と呼ぶが，実用に耐えるようにするためには十分大きな周期が得られるように工夫を要する。

II節に掲げたプログラムの理論的な根拠は混合型合同法[(1)]と呼ばれる次の再帰関係式である。

$$\begin{cases} x_i \equiv a x_{i-1} + c \pmod{m}, & (i=1, 2, 3, \cdots) \\ x_0 = b \end{cases}$$

ここに，a，b，c，x_i は0から $m-1$ までの整数値をとり得る。この式に従って数列 x_1，x_2，x_3，……を作ってゆくと明らかに最大周期は m であるが，定数の選び方によっては一般に周期はもっと短くなる。整数論において次の命題の成り立つことがわかっている。

「（1）　c と m は互いに素である。

（1）　乱数には，次式で示される乗算型合同法によるものがある。

$$\begin{cases} x_i \equiv a x_{i-1} \pmod{m} \\ x_0 = b \end{cases}$$

これは性質として $m=2^k$ のとき

$$\begin{cases} a \equiv 3 \pmod{8} \\ b：\text{奇数} \end{cases}$$

とすれば最大周期 2^{k-2} を与えることがわかっている。従って計算機の制約内でなるべく大きい周期を持つようにということで，混合型合同法をプログラムした。

（2） m のあらゆる素因数 p に対して

$$a\equiv 1 \pmod{p}$$

（3） m が4の倍数であれば，

$$a\equiv 1 \pmod{4}$$

を満足するように，a，c，m を選べば，初期値 b の値に無関係に数列 x_1，x_2，x_3 ……は最大周期 m を持つ。」

計算機を考慮にいれると $m=2^{15}$ または $m=2^{31}$ であるから，上記の（1），（2）の条件は，

（1）′ $c\equiv 1 \pmod{2}$

（2）′ $a\equiv 1 \pmod{2}$，

すなわち，a も c も少なくとも奇数でなければならない。最大周期として m が得られるということは，m 個の数値すべてが1回ずつ現われることを意味するが，(ii) の等確率性が保証されたわけではない。厳密には具体的な数列について，別の検討を必要とする事柄である。

ここでは (iii) の無規則性について，考えてみよう。前記計算法で得られる数列に関して，相続く2つの乱数 x_i と x_{i+1} との相関係数 ρ は

$$\frac{1}{a}-\frac{6c}{am}\left(1-\frac{c}{m}\right)-\frac{a}{m}<\rho(x_i, x_{i+1})<\frac{1}{a}-\frac{6c}{am}\left(1-\frac{c}{m}\right)+\frac{a}{m}$$

であることが解っており，無規則性の立場からは $\rho(x_i, x_{i+1})$ の絶対値がなるべく0に近くなるように a，c，m を選べということになる。このことをもう少し考えてみると

（イ） $\frac{1}{a}-\frac{6c}{am}\left(1-\frac{c}{m}\right)\pm\frac{a}{m}$ が0に近いこと

（ロ） ρ の範囲を小さくするために，a/m が小さいこと

が要請される。（イ）の要請を考えてみる。

$$\frac{1}{a}-\frac{6c}{am}\left(1-\frac{c}{m}\right)=\frac{a}{m} \qquad (\alpha)$$

と置く。a について解くと

$$a=\pm\sqrt{m-6c\left(1-\frac{c}{m}\right)}$$

系列相関を小さくするという立場からは

$$a\fallingdotseq\sqrt{m}$$

であることが望ましいと書かれている教科書もある。これはたとえば $c=1$ として上式

の根号内の $c\left(1-\frac{c}{m}\right)$ を十分小さいとみて無視する立場をとったものかも知れない。しかしここではもう一度等号の成立する時を考えてみる。上記の等式 (α) を変形すると

$$\frac{\left(c-\frac{m}{2}\right)^2}{\frac{m^2}{12}}-\frac{a^2}{\frac{m}{2}}=1$$

をうる。これは図1に示すような双曲線となる。条件(1)′,(2)′,(3)を満たす a は $4k+1$ (k は0または0より大きい整数), c は $2l+1$ (l は0または0より大きい整数)の形で，これを満たしつつ図の双曲線上の点になるべく近いものを選ぶことが考えられる。さらに，m は計算機の制約で決ってくる定数であるから，(ロ)の要請は a をなるべく小さく選ぶべきであることを意味する。そこで $a=1$ とすると

$$x_i \equiv x_{i-1}+c \pmod{m}$$

となる。$m=2^{15}$ のとき式 (α) を近似的に満たす奇数として $c=6923$ または 25843 をうる。乗算がないので一見計算機向きであり，相関係数も下表で示すとうり小さい。しかしながら $c=6923$ にとると m に比べて値が小さく，等差数列がいくつかずつ組になって現われてしまう。$c=25843$ を採用すると $c=6923$ の場合のようにはならないが，〔0,1〕を10等分して一様性とみるとやはり等差数列的な動きを示す。ゆえにどういう c であっても $a=1$ としない方が良いように思われる。次に小さな a，ということで $k=1$ にとって，

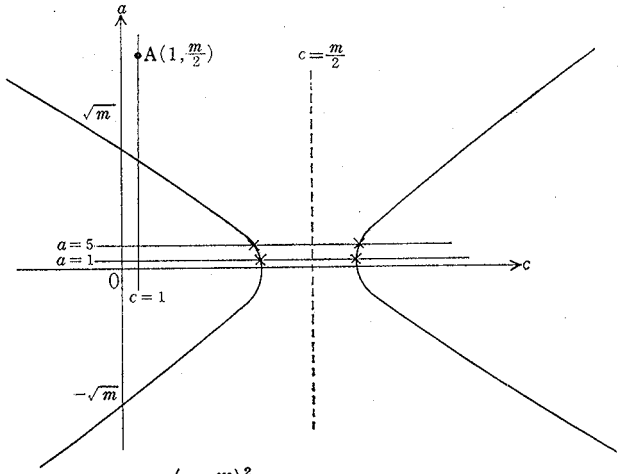

図1　$\frac{\left(c-\frac{m}{2}\right)^2}{\frac{m^2}{12}}-\frac{a^2}{\frac{m}{2}}=1$

$a=5$ としてプログラムを組んである。このときは $c=6917$ とした。$c=25851$ を採用しない根拠はない。

以上の考えによる m, a, c の値と相関係数の表を次にかかげておく。

相関係数 ρ の範囲

$m=2^{15}$			$m=2^{31}$		
$a=1$	$c=6923$	$-0.1\times10^{-6}<\rho<0.6\times10^{-4}$	$a=1$	$c=453816693$	$-0.6\times10^{-16}<\rho<0.9\times10^{-9}$
	$c=25843$	〃		$c=1693666955$	$0.2\times10^{-16}<\rho<0.9\times10^{-9}$
$a=5$	$c=6917$	$0.0<\rho<0.3\times10^{-3}$	$a=5$	$c=453816685$	$-0.1\times10^{-16}<\rho<0.5\times10^{-8}$
	$c=25851$	$0.5\times10^{-7}<\rho<0.3\times10^{-3}$		$c=1693666963$	〃
$a=9$	$c=6901$	$0.0<\rho<0.5\times10^{-3}$	$a=9$	$c=453816669$	$0.0<\rho<0.8\times10^{-8}$
	$c=25867$	$0.6\times10^{-7}<\rho<0.5\times10^{-3}$		$c=1693666979$	$0.1\times10^{-16}<\rho<0.8\times10^{-8}$
$a=181$	$c=1$	$0.2\times10^{-6}<\rho<0.1\times10^{-1}$	$a=46341$	$c=1$	$0.5\times10^{-10}<\rho<0.4\times10^{-4}$
$a=16385$	$c=1$	$-0.5<\rho<0.5$	$a=1073741825$	$c=1$	$-0.5<\rho<0.5$

表から二つのことがわかる。一つは，(1)′，(2)′ および (3) を満たすように a, c を選び，双曲線に近い点ならばたいていのものは相関係数が小さいということ。もう一つは，表の最後の行で示されているように，a, c を双曲線から離れて採ると（たとえば図1におけるA点で，$a\fallingdotseq\frac{m}{2}$, $c=1$）相関係数が大きくなることがあるということである。

ただしこれまで述べてきたことは，相続く x_i と x_{i+1} の間の相関を小さくするという立場からのもので，x_i と x_{i+n} ($n=2, 3, \cdots\cdots$) の相関についてはまだ何も考えていない段階である点を注意しておく。

IV 計算機での mod 計算

$$x_i\equiv ax_{i-1}+c \pmod m$$

とは，$ax_{i-1}+c$ の値を m で割ったときの余りを x_i にするということであるのに，II節で紹介したプログラムは割り算を用いていない。これは固定小数点演算（整数型演算）における桁あふれ（overflow）現象が丁度整数論の mod 計算に対応することを用いたものである。

単精度の場合の整数値の内部表現がどのようになるか次に掲げるようなプログラムで調べてみた。

```
      C
      C       タンーセイド
      C
0002          ＩＡ＝32767
0003          ＩＢ＝32768
0004          ＩＣ＝８
0005          ＩＤ＝ＩＣ*(2**12+1)
0006          WRITE (6, 10)ＩＤ,ＩＤ
0007       10 FORMAT(‘ *** タンーセイド *** ’//‘ ＩＤ＝’, Ｉ6, 8X, Z8)
0008          ＩＦ(ＩＤ) 20, 30, 30
0009       20 ＩＸ＝ＩＤ+32767+1
0010          ＩＹ＝ＩＤ+32768
0011       30 ＷＲＩＴＥ(6,40) ＩＡ,ＩＡ,ＩＢ,ＩＢ,ＩＣ,ＩＣ,ＩＸ,ＩＸ,
              ＩＹ,ＩＹ
0012       40 ＦＯＲＭＡＴ(‘ＩＡ＝’,Ｉ6,8X,Z8/‘ＩＢ＝’,Ｉ6,8X,Z8/‘ＩＣ
              ＝’,Ｉ6,8X,Z8/‘ＩＸ＝’,Ｉ6, 8X,Z8/‘ＩＹ＝’,Ｉ6,8X,Z8/)
```

FORMAT 文中 Z は内部表現を知るために16進数で表現することを指定したものである。結果は次のようである。

```
*** タンーセイド ***
ＩＤ＝−32760          ８００８
ＩＡ＝ 32767          ７ＦＦＦ
ＩＢ＝−32768          ８０００
ＩＣ＝     8          ０００８
ＩＸ＝     8          ０００８
ＩＹ＝     8          ０００８
```

単精度の整数は16ビットの情報を用いて表現される。ビット・パターン

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
	A_1				A_2				A_3				A_4		

（a_i は0または1）
（16進法表示）

によって，数

$$a=a_1\times2^{14}+a_2\times2^{13}+a_3\times2^{12}+\cdots\cdots+a_{14}\times2+a_{15}$$
$$=A_1\times16^3+A_2\times16^2+A_3\times16+A_4$$

をあらわす。また

1	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}	b_{13}	b_{14}	b_{15}

によって，数

$$-2^{15}+(b_1\times2^{14}+b_2\times2^{13}+\cdots\cdots+b_{14}\times2+b_{15})$$
$$=-2^{15}+b$$

をあらわす，$b<2^{15}$ であるから，この数は負数である。例えば，

1 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1
F	F	F	F

は，

$$-2^{15}+(2^{14}+2^{13}+\cdots+2+1)=-32768+32767=-1$$

をあらわしている。このような負数の表示を補数表示といい，この場合は $2^{15}=32768$ の補数をもちいているわけである。これに対して，-1 のように絶対値の前に符号 $-$ をつけて負数を表示する方法を絶対値表示という。2進法による計算機では補数表示をとるのが通常である。補数表示の場合，0ビットが0の場合は正数を，1の場合には負数をあらわすことになるから，これを符号ビットとよぶ。このような数の表示法をとっているので，単精度整数が表現しうる最大の数は，

0 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1
7	F	F	F

すなわち，

$$2^{14}+2^{13}+\cdots+1=2^{15}-1=32767=7\,\mathrm{F}\,\mathrm{F}\,\mathrm{F}\ (16\text{進法表現})$$

である。また最小の数は，

1 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
8	0	0	0

すなわち，

$$-2^{15}=-32768=-8000\ (16\text{進法表現})$$

である。

補数表示の負数の絶対値をもとめる方法を考えてみる。補数表示の負数は

1 b_1 b_2 b_3	b_4 b_5 b_6 b_7	b_8 b_9 b_{10} b_{11}	b_{12} b_{13} b_{14} b_{15}

というパターンを示す。これによって表現されている数はつぎのとうりである。すなわち，

$$-2^{15}+b_1\times2^{14}+b_2\times2^{13}+\cdots+b_{15}=-2^{15}+b$$
$$=-(2^{15}-1+1)+b=-(2^{-15}-1+1-b)$$
$$=-[\{(2^{15}-1)-b\}+1]$$

ここで，$(2^{15}-1)$ のビット・パターンは，

0 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1

である。b の各ビットは0または1であるから，$\{(2^{15}-1)-b\}$ の各ビットは，$(1-b_i)$ すなわち，b_i が1ならば0，b_i が0ならば1となる。0ビットも，1から0にかわるのであるから，結局，すべてのビットについて，1を0に0を1に変換したものが $\{(2^{15}-1)-b\}$ である。その後に+1すなわち，15ビットに1を加えればよい。例えば，

1 0 0 0	0 0 0 0	0 0 0 0	0 0 0 1

は，

0 1 1 1	1 1 1 1	1 1 1 1	1 1 1 0
＋			1
0 1 1 1	1 1 1 1	1 1 1 1	1 1 1 1
7	F	F	F

＝32767

となり，絶対値表現として10進法表示−32767，16進法表示−7FFFをうる。

上に述べたように，単精度整数によって表現しうる整数は，−32768から+32767までの整数である。ところが上述のプログラムにおいて，ＩＢ＝32768，あるいは，ＩＹ＝ＩＤ+32768というように，この範囲をこえた整数32768がかかれていながら，OVER SIZED CONST. のエラー表示がない。これはつぎのような理由からである。すなわち，FACOMマニュアルに「32768以上は自動的に倍長定数となる」と書かれているように，32768はコンスタント・エリアに32ビットをとって，

0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	1 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
0	0	0	0	8	0	0	0

として確保される。これをＩＢ＝32768によってＩＢにうつすことを考えてみる。ＩＢは単精度整数タイプであるから，エリアは16ビットしかない。そこで右半分だけとって，8000が入ることになる。これを10進数として出力すれば，−32768となることは前に述べた通りである。

つぎにＩＤについて考えてみる。ＩＤの出力をみれば，ＩＤ＝$8\times(2^{12}+1)$ の計算を，付号ビットを無視して

$2^{12}+1 \equiv 1001 = 0001\ 0000\ 0000\ 0001$

と

$$8=2^3=1000$$

をかけて，

1000 0000 0000 1000＝8008（−32760）

としていることがわかる。乱数をつくるためには，x_{i+1} が m（この場合は $2^{15}=32768$）より大きくなったとき，mod (m) をとって，m より小さい数にしておきたい。すなわち，

$$ID=8\times(2^{12}+1)=2^{15}+2^3=32776$$

$$32776\equiv 8 \mod (32768)$$

であるから，上のＩＤの０ビットの１を０にかえたいのである。このために 32767＋１あるいは32768を加えて，０ビットに１を加え，１０として，１をあふれさせてしまえばよい。これが，ＩＸ，ＩＹであってどちらでもよいことがわかる。乱数発生のプログラムでは，IRANDが32768より大きくなったかどうかは，IRANDの付号ビットが１か０か，すなわち，IRANDが負であるか正であるかによって判定できることをもちいる。すなわち，IRANDが負と判定されたときは，IRANDは32768より大きな数になっているのであるから，32768を加えて符号ビットを０にしてmodをとっているのである。

倍精度の場合について，単精度のときとほぼ同じであるが，次に掲げるようなプログラムを考えてみる。

```
0001            DOUBLE INTEGER DA,DB,DC.DD,DE,DX,DY
                C
                C   バイ-セイド
                C
0013            DA=32768
0014            DB=2147483647
0015            DC=2147483648
                  ¥
F002W       OVERSIZED CONST.
0016            DD=8
0017            DE=DD*(2**28+1)
0018            WRITE(6,50)DE,DE
0019         50 FORMAT(///'*** バイ-セイド***'//'DE=',I11,3X,Z8)
0020            IF(DE)60, 70, 70
0021         60 DX=DE+2147483647+1
0022            DY=DE+2147483648
                     ¥
F002W       OVERSIZED CONST.
0023         70 WRITE(6,80)DA,DA,DB,DB,DC,DC,DD,DD,
                DX,DX,DY,DY
```

```
0024          80  FORMAT('D A =',I 11,3X, Z8/'D B =',I 11, 3X, Z8/'D
                  C =',I 11,3X, Z8/'D D =',I 11,3X, Z8/'D X =',I 11,3X,
                  Z8/'D Y =',I 11,3X, Z8)
0025              STOP
0026              END
```

結果は次のようである。

```
*** バイ-セイド ***
D E =-2147483640     8 0 0 0 0 0 0 8
D A =       32768    0 0 0 0 8 0 0 0
D B =  2147483647    7 F F F F F F F
D C =           1    0 0 0 0 0 0 0 1
D D =           8    0 0 0 0 0 0 0 8
D X =           8    0 0 0 0 0 0 0 8
D Y =-2147483639     8 0 0 0 0 0 0 9
```

倍精度整数は，単精度2語分を使用するので32ビット。単精度のときの符号ビットがどう扱われるか見るためにＤＡに32768を入れてみる。内部表現は 00008000 で10進変換後はＤＡ＝32768 と印刷される。ゆえに単純に０～31ビットまで２語分つながれて，０ビットが符号ビットに，16ビッ目はふつうのビットの扱いになると考えて良い。マニュアルによれば「倍長整定数がとりうる最大値は$2^{31}-1$である。負の最小値-2147483648（$=-2^{31}$）はコンパイル時には指定できない」とあるが，これを調べてみる。ＤＢに$2^{31}-1=2147483647$をいれると16進で７ＦＦＦＦＦＦＦ，10進でＤＢ＝2147483647と印刷されこれは問題ない。次にＤＣに$2^{31}=2147483648$をいれてみると，前記プログラム中でわかるように，コンパイラーによりＦ002Ｗ OVERSIZED CONST. のエラー・メッセージが出される。実際にＤＣには16進で00000001すなわち１がはいっている。(2) これは，定数については単精度のとき符号ビットも含めて情報を入れてしまい，内部表現を外部表現にするときにマイナスの値と解釈したのと様子が異っている。

さらにＤＤ＝８としてＤＥ＝ＤＤ×$(2^{28}+1)=2^{31}+2^3=2147483648+8=2147483656$ の筈が実際は-2147483640，で内部表現が 80000008．単精度の時と同様に，ＤＥには2^{31}の mod をとった残り８だけがはいって欲しいので符号 bit の１を０にしたい。2^{31}を加えたいのであるが前記のことよりFORTRAN文ではこれを2147483647＋１を加えることによっ

（２）コンパイル時のエラー・メッセージ表には，このＦ002Ｗのとき処置として，「整数型は表わされる最大値に符号をつける」となっているが，調べてみると2147483649＝$2^{31}+2$も4294967297＝$2^{62}-2$も１となってしまい，マニュアルどうりではない。

て実現する。数学的に同じであるからと言って2147483648を加えてはならない。その実例を，DXとDYで示してあるので結果の違いに注意されたい。

以上でII節のプログラムの妥当であることが証明された。

V おわりに

$x_i \equiv a x_{i-1} + c \pmod{m}$ で $a \neq 1$ のときのプログラムは，たとえばIRAND=IRAND*5+6917として右辺に乗算があるので，2つのレジスターをつないで演算をするためにか，計算機は何のエラー・メッセージも出さない。しかし $a=1$ のときは，

$$x_i \equiv x_{i-1} + c \pmod{m}$$

で，プログラムはIRAND=IRAND+6923となって右辺に乗算を含まない。このときは単精度で32768，倍精度で2147483648を越えることがあるとF501W Fixed Point Overflowのメッセージが，この演算を行なうたびに印刷される。乱数としての結果には影響がないので放っておいてもかまわないが，気になるようなら ¥JEND カードの前でデーターの後，または ¥FD UIN=*のすぐ手前に

¥␣PARA␣MSGLEVEL=0

のカードをいれておくと，このメッセージは印刷されない。このコントロール・カードは「演算割り込みが起きてもエラーとみなさずメッセージを出力するな」の意味であり，他の所で0−Dividが起きてもエラーメッセージを出さないので注意を要する。勿論，演算割り込み以外はチェックされる。

乱数を発生させるのに

$$x_{i+1} \equiv x_i + x_{i-1} \pmod{m}$$

の式に基づくものがある。これは乗算を用いないという点で計算機向き，かつ2つの値が共に同じ状態にならないと周期が来ないので，かなり長い周期が期待されるが，上記F501Wのメッセージに必ずぶつかると思われるので，MSGLEVEL=0 の指定をしてプログラムを実行した方が良い。

参 考 文 献

1. 三根久著 “オペレーションズ・リサーチ” p. 72〜74
2. 井上勝人，高橋知子著 “事業部投資提案の態様” 香川大学経済論叢 Vol. 50, No. 1 p. 98—105