

有理関数近似を利用した

正規乱数発生法

電試計算センター 戸田英雄

1. まえおき

コンピュータで正規擬似乱数を発生させる方法は、種々あつて一長一短<sup>ど</sup>がある。筆者は最近正規分布関数の逆関数の有理関数近似を種々作成して来たが、この中で擬似乱数発生に利用できるものがあるので報告する。

正規乱数発生法のうち8種をとり FACOM 230-50 の SAFF (アセンブラ) 語でプログラムを作り、計算時間の比較 (計算センターの島山高君による) を行なつた結果を次の表で示す。

方 法	名 称	10000個 発生時間 (単位秒)	精 度		メモリー スペース
			xの相対誤差	xの範囲	
(1)一様乱数の12個の和	RAN1	1785			30
(2)逆関数法					
(2.1) 山内のミニマックス近似	RAN6	10.10	$4.89 \cdot 10^{-4}$	4.74	48
(2.2) 戸田のミニマックス近似	RAN5	10.56	$1.46 \cdot 10^{-4}$	4.74	52
(2.3) 区分わけミニマックス多項式	RAN2	4.39	$1.9 \cdot 10^{-3}$ (約)	3.49	116
(2.4) 区分わけミニマックス有理式	RAN3	4.99	$4.90 \cdot 10^{-4}$	5.04	194
(3)直接法 (Box-Muller)	RAN4	11.30 <sup>*</sup>	$5.0 \cdot 10^{-7}$	6.0	40
(4)吉沢法	RAN7	3.28			129
(5)棄却法	RAN8	26.64	$5.0 \cdot 10^{-7}$	5.0 (とされた)	50

IBM 704 による実験結果 (4) の約 1/6 である。

この表から分かるように、計算速度を速くするためにはメモリスペースが大きくなる。一番手がるなのは(1)であろう。一番速いのは(4)か(2,4)である。何等の近似を用いていないのは(3)である。しかし一長一短があつてどのプログラムが一番よいのか評価は、このように問題を限定しても、使用者が何に重点をおくかで異なると思う。

一様乱数の発生には、FACOM 230-50 の RANDOM の手続きを用いた。また参考のため FACOM 230-50 の演算速度を次にあげる

演算速度:

演算 \ 数	固定小数点	浮動小数点	
		1倍精度	2倍精度
加算 / 減算	4.4	15.4	18.0
乗算	25.3	30.8	88.0
除算	35.2	39.6	107.8
ロード/ストア	4.4	4.4	6.6

(単位はマイクロセカンド)

RANDOMの手続き:

$$\text{混合型合同式 } u_{i+1} \equiv (2^7 + 1)u_i + 1 \pmod{2^{35}}$$

を用いてある。

## 2. 発生法の主なもの

区間  $(0, 1)$  の上で一様に分布する変数が、擬似乱数  $u$  で与えられたとする。  
 (この発生法についてはここでは取扱わない) この  $u$  を変換して正規擬似乱数  
 を作る方法は種々あつて次にその主なものを取り上げる。

## (1) 中心極限定理を利用する方法

独立な  $n$  個の  $u$  の和を  $\Delta_n$  とすれば、 $(\Delta_n - n/2) / \sqrt{n/12}$  は中心極限定理で  $n \rightarrow \infty$  において標準正規分布  $N(0, 1)$  に近づく。 $n \geq 5$  でもかなり実用には使われているが、 $n = 12$  とすると独立な  $u$  の 12 個の和から 6 をひく計算で済む。

## (2) 逆関数法

$$u(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2} dt$$

を満たす  $x = x(u)$  を求める。この計算のための近似式の精度までの正規乱数が作れる。従来よく見られたのは Hastings の近似式である。これより山内二郎【2】か筆者【3】の近似式の方が精度も計算時間も得をする。

(2.1) 山内のミニマックス近似式 (  $p$  が  $u$  に相当する近似式である )

$$y = -\log_e (4p \cdot (1-p)) , \quad \text{ここで}$$

$$0 \leq y \leq 10 \quad (1.135_{10} \cdot 10^{-5} \leq p \leq 0.5)$$

$$x(p) = \left\{ y \left( 2.0611786 - \frac{5.7262204}{y + 11.640595} \right) \right\}^{\frac{1}{2}} \quad (2.1)$$

相対誤差を  $E_R(y)$  とすれば  $|E_R(y)| \leq 4.89_{10} \cdot 10^{-4}$  である。

(2.2) 戸田のミニマックス近似式 (  $p$  は  $u$  に相当して )

$$y = -\log_e(4p \cdot (1-p)) \text{ ここで } 0 \leq y \leq 10.$$

$$x(p) = \left\{ y (t_0 + t_1 y + t_2 y^2 / (t_3 + t_1 y)) \right\}^{1/2} \quad (2.2)$$

ここで

$$t_0 = 3.702\ 993\ 4$$

$$t_1 = -0.029\ 489\ 901$$

$$t_2 = 1.956\ 129\ 4$$

$$t_3 = -0.917\ 227\ 58$$

相対誤差を  $E_R(y)$  とすれば  $|E_R(y)| \leq 1.46_{10}^{-4}$  である。

(2.3) 区分わけミニマックス多項式近似

$$u = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2} dt$$

から  $x=x(u)$  を求めるのに  $0.5 \leq u < 1$  の区間を図 2-1 のように区分

わけして、各区分で簡単な多項式で近似することを考えた。

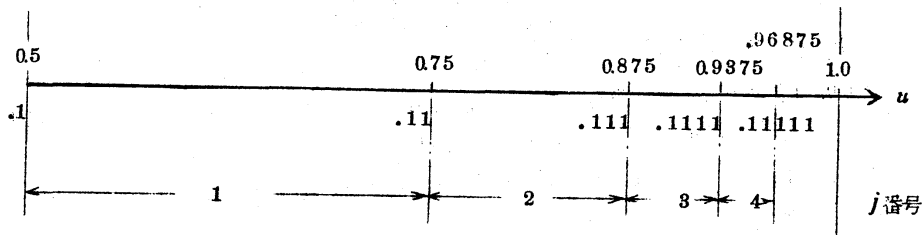


図 2-1 ミニマックス多項式近似 (2.3) 式のため  $u$  の  $[0.5, 1.0)$  区間をサブ区分にわけ

すなわち、

$$\frac{2^j - 1}{2^j} \leq u_j \leq \frac{2^{j+1} - 1}{2^{j+1}}, \quad (j = 1, 2, \dots, 11)$$

に区分にわけて、この範囲内の $u_j$ に対して

$$x_j = a_{0j} + u_j (a_{1j} + u_j a_{2j}) \quad (2.3)$$

である2次式で近似し、 $x_j$ の絶対誤差 $E(u_j)$ がミニマックスになるように $a_{0j}, a_{1j}, a_{2j}$ を定めた。

各 $j$ 区分における係数 $a_{0j}, a_{1j}, a_{2j}$ と $|E(u_j)|$ の値の表は省略した。これから $|E(u_1)| \leq .19_{10}^{-2}$ という $j=1$ の区分を除けば絶対誤差は $j=2, \dots, 11$ で $0.1_{10}^{-2}$ でそろっている。

(2.4) 区分わけミニマックス有理式近似

図2-2に示すようにの範囲 $[0.5, 1.0)$ を次のように区分にわけ  
る。0.5と0.75の間を更に2等分してある。

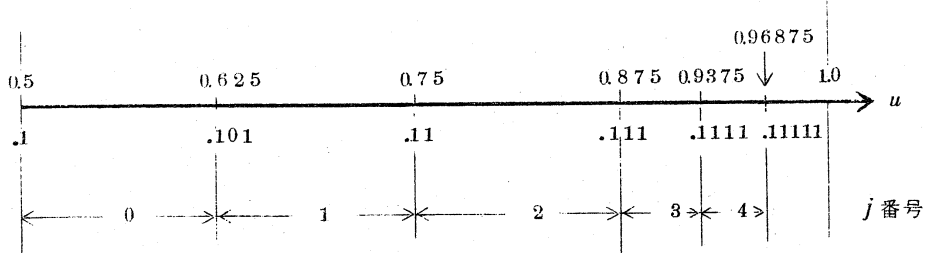


図2-2 ミニマックス有理式近似(2.4)式のため $u$ の  
[0.5, 1.0) 区間をサブ区分にわけ

すなわち、

$$4/8 \leq u_0 \leq 5/8$$

$$5/8 \leq u_1 \leq 6/8$$

$$(2^j - 1)/2^j \leq u_j \leq (2^{j+1} - 1)/2^{j+1} \quad (j=2, 3, \dots, 21)$$

として、各 $u_j$ に対して、

$$x_j = a_{0j} \cdot u_j + a_{1j} + \frac{b_{0j}}{(1.0 - u_j) + b_{1j}} \quad (j=0, 1, 2, \dots, 21) \quad (2.4)$$

で計算する。このとき  $x_j$  の真値を  $\bar{x}_j$  とし、相対誤差  $E_R(u_j)$  とすると、 $E_R(u_j) = (x_j(u_j) - \bar{x}_j) / \bar{x}_j$  である。

各々の区分  $j$  において  $E_R(u_j)$  がミニマックスになるように (2.4) の係数を決めた。(2.4) 式の誤差曲線  $E_R(u_j)$  を次の図 2-3 に示す。係数は次の表で示す(附録のプログラムリストには 8 進表示で与えている)。

J	A0	A1	B0	B1
0	.24789809E+01	-.12447336E+01	.10405172E-02	-.30154771E+00
1	.22860349E+01	-.12333789E+01	.42388247E-01	-.30903297E-01
2	.21033506E+01	-.12453123E+01	.11148991E+00	.75767737E-01
3	.21672418E+01	-.12555724E+01	.97310666E-01	.65919351E-01
4	.24380721E+01	-.13290065E+01	.59425777E-01	.40379643E-01
5	.28232681E+01	-.14734938E+01	.32321199E-01	.27491959E-01
6	.32721248E+01	-.16684887E+01	.16601244E-01	.11969536E-01
7	.37509305E+01	-.18917359E+01	.82360955E-02	.61963355E-02
8	.42366073E+01	-.21262499E+01	.39981204E-02	.31512073E-02
9	.47145188E+01	-.23610799E+01	.19162446E-02	.15846752E-02
10	.51765333E+01	-.25900890E+01	.91255517E-03	.79155594E-03
11	.56190174E+01	-.28103753E+01	.43362474E-03	.39392518E-03
12	.60411078E+01	-.30209661E+01	.20613067E-03	.19569230E-03
13	.64434547E+01	-.32219237E+01	.98168190E-04	.97154838E-04
14	.68274165E+01	-.34138020E+01	.46870226E-04	.48235570E-04
15	.71945938E+01	-.35973418E+01	.22439951E-04	.23956655E-04
16	.75465857E+01	-.37733144E+01	.10772967E-04	.11904269E-04
17	.78848768E+01	-.39424489E+01	.51852741E-05	.59185305E-05
18	.82107966E+01	-.41054034E+01	.25017402E-05	.29441034E-05
19	.85255134E+01	-.42627593E+01	.12096295E-05	.14652206E-05
20	.88300471E+01	-.44150249E+01	.58601846E-06	.72952939E-06
21	.91252851E+01	-.45626432E+01	.28440384E-06	.36337175E-06

### (3) 直接法 (Box - Muller 法)

$u_1, u_2$  を  $(0, 1)$  の上の独立な一様乱数とする。

$$x_1 = \sqrt{-2 \log_e u_1} \cdot \cos 2\pi u_2, \quad x_2 = \sqrt{-2 \log_e u_1} \cdot \sin 2\pi u_2 \text{ とすると } (x_1, x_2)$$

の同時分布  $g(x_1, x_2)$  は、

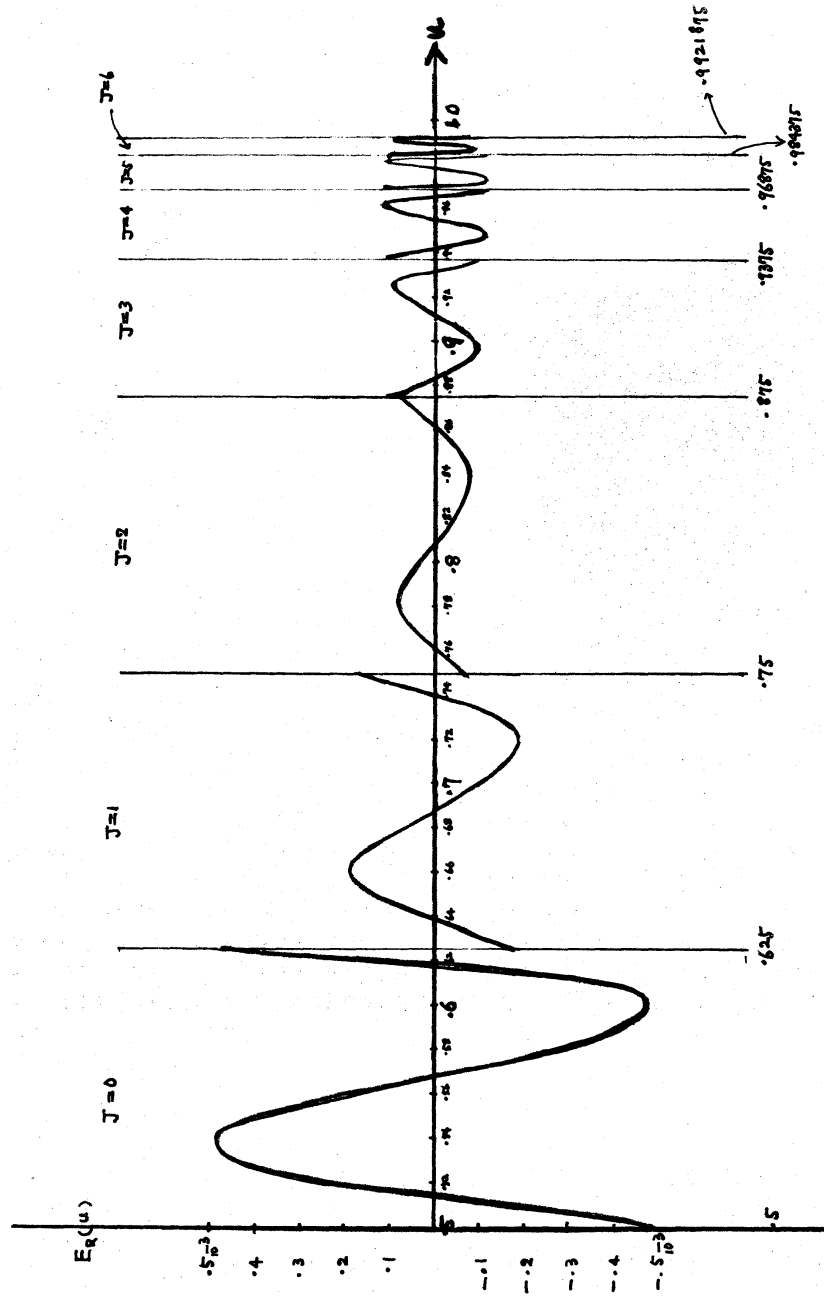


図 2-8 区分わけニマックス有理式の誤差曲線

$$f(x_1, x_2) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x_1^2} \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x_2^2}$$

に従うという性質を利用したものである。

この方法は、正規分布の端の方まで精度よく求められると M. E. Muller [4] は述べている。

#### (4) 吉 沢 法 [1]

$n$  個の一樣乱数から、Yates の算出を用いる直交変換で  $n$  個の互いに無相関な乱数を作る方法である。

中心極限定理を用いる (1) の方法では、 $n$  個の一樣乱数から 1 個の正規乱数しか作れなかつたのに比べて、この方法は  $n$  個から  $n$  個できるので非常に能率がよい。変換された乱数は個々には擬似正規乱数とみなすことが出来るが、逆変換によつてもとの一樣乱数にもどつてしまう性質があるので、完全な正規乱数とみなすことは出来ないが、使い方によつては非常に有用である。

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = T \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix}, \quad \begin{pmatrix} y_5 \\ y_6 \\ y_7 \\ y_8 \end{pmatrix} = T \begin{pmatrix} r_5 \\ r_6 \\ r_7 \\ r_8 \end{pmatrix}, \quad \begin{pmatrix} y_9 \\ y_{10} \\ y_{11} \\ y_{12} \end{pmatrix} = T \begin{pmatrix} r_9 \\ r_{10} \\ r_{11} \\ r_{12} \end{pmatrix}, \quad \begin{pmatrix} y_{13} \\ y_{14} \\ y_{15} \\ y_{16} \end{pmatrix} = T \begin{pmatrix} r_{13} \\ r_{14} \\ r_{15} \\ r_{16} \end{pmatrix} \quad (4.1)$$

$$\begin{pmatrix} x_1 \\ x_5 \\ x_9 \\ x_{13} \end{pmatrix} = T \begin{pmatrix} y_1 \\ y_5 \\ y_9 \\ y_{13} \end{pmatrix}, \quad \begin{pmatrix} x_2 \\ x_6 \\ x_{10} \\ x_{14} \end{pmatrix} = T \begin{pmatrix} y_2 \\ y_6 \\ y_{10} \\ y_{14} \end{pmatrix}, \quad \begin{pmatrix} x_3 \\ x_7 \\ x_{11} \\ x_{15} \end{pmatrix} = T \begin{pmatrix} y_3 \\ y_7 \\ y_{11} \\ y_{15} \end{pmatrix}, \quad \begin{pmatrix} x_4 \\ x_8 \\ x_{12} \\ x_{16} \end{pmatrix} = T \begin{pmatrix} y_4 \\ y_8 \\ y_{12} \\ y_{16} \end{pmatrix} \quad (4.2)$$

この算法は  $r_i$ ,  $i=1, 2, \dots, 16$  とし、まず (4.1) で  $y_i$ ,  $i=1, 2, \dots, 16$  を計算する。次に (4.2) で  $x_i$ ,  $i=1, 2, \dots, 16$  によつて計算す



る。この  $x_i$  ( $i=1, 2, \dots, 16$ ) が正規乱数となる。ただし  $r_i$  が  $(0, 1)$  の一様乱数のときは、 $x_i$  から  $\theta$  をひけば他の  $x_i$  と同じ期待値をもつ。

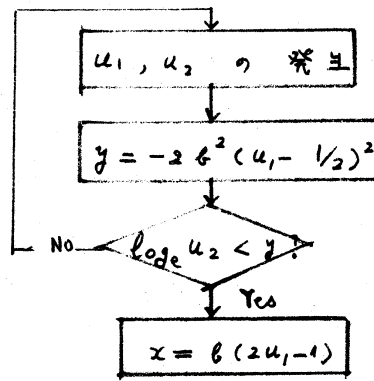
ただし

$$T = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

である。

### (5) 棄却法

考え方は Von Neumann によるといわれる。渋谷正昭 [6] により他の面白い方法も考えられているが、ここでは D. Teichroew [7] ([5] から引用) の方法によると、 $-\theta \leq x \leq \theta$  の区間の正規乱数を作る手順は次のようになる。



この方法は両側で打ち切られた区間 (truncated region  $-\theta \leq x \leq \theta$ ) での正規乱数を作るのに適しているが、能率はあまりよくないと言われている。

## 3. 附 録

## 3.1 プログラムリスト

(1) 一様乱数の12個の和

(2) 逆関数法(区分わけミニマックス有理式)

~~(3) Box-Muller法~~

についてリストを参考のため示す。ここに用いている  $\log_e x$  は FAST

LOG関数副プログラム【3】を用いてある。従来のALOGの $\frac{1}{2} \sim \frac{1}{3}$ の速  
 さで計算できるが精度は小数点以下6桁である。

(1) 一様乱数の12個の和

RAN1	J	*-*		RAN10020
	STX,3	ST3	* SUM OF 12 UNIFORM DEVIATES *	RAN10030
	LX,2	RAN1		RAN10040
	LXI,3	0		RAN10050
	STZ	X		RAN10060
REPEAT	LAI	0		RAN10070
	LR	CONST		RAN10080
	LARL	7		RAN10090
	XAR	0		RAN10100
	ANT	CONST		RAN10110
	ANT	ONE		RAN10120
	AND	MASK		RAN10130
	STA	CONST		RAN10140
	LZI	256		RAN10150
	FA	ZFRO		RAN10160
	FA	X		RAN10170
	FST	X		RAN10180
	AXI,3	1		RAN10190
	TXL,3	12		RAN10200
	J	REPEAT		RAN10210
	FS	R6		RAN10220
ST3	LXI,3	*-*		RAN10230
	J*	RAN1		RAN10240
*				RAN10250
X	OCTAL	0		RAN10260
CONST	OCTAL	303725614326		RAN10270
MASK	OCTAL	377777777777		RAN10280
ONE	OCTAL	1		RAN10290
R6	OCTAL	300000000403		RAN10300
ZERO	OCTAL	0		RAN10310
	END			RAN10320

## (2) 逆関数法 (区分わけミニマックス有理式)

RAN3	J	*-*		RAN30010
	STX,4	LX4	* MINMAX RAT. APPROX OF SUB INT. *	RAN30020
	STX,5	LX5		RAN30030
	LX,2	RAN3	X = A0*U+A1+B0/(1-U+B1)	RAN30040
REPEAT	LXI,5	1		RAN30050
	STZ	ME		RAN30060
	STZ	SIGN		RAN30070
	LAI	0		RAN30080
	LR	CONST		RAN30090
	LARL	7	PSEUDO	RAN30100
	XAR	0	RANDOM	RAN30110
	ANT	CONST	NUMBER	RAN30120
	ANT	IONE	GENERATION	RAN30130
	AND	MASK	ROUTINE.	RAN30140
	STA	CONST		RAN30150
	TB1N,0	CONST		RAN30160
	J	*+4	IF (U.GE.0.5) JUMP	RAN30170
	MBON,1	SIGN	NO NEGATIVE	RAN30180
	NOT	0	SIDE	RAN30190
	AND	MASK		RAN30200
	STA	ME+4		RAN30210
	LZI	256	NORMALIZE	RAN30220
	FA	ME		RAN30230
	FST	X	WORK PSEUDO-RANDOM U	RAN30240
	TB2F,0	ME+4		RAN30250
	J	*+10	IF (U.LT.0.75) JUMP	RAN30260
	LA	ME+4		RAN30270
	LAL	1		RAN30280
	LAL	1	COEFFICIENT LOCATION COUNT	RAN30290
	JFA0	*+3	TEST	RAN30300
	AXI,5	1	INCREMENT XR5	RAN30310
	J	*-3		RAN30320
	TXHE,5	22	IF XR5 .GT. 21 (U=5.035)	RAN30330
	J	GR	YES MESSAGE PRINT	RAN30340
	J	*+3		RAN30350
	TB3F,0	ME+4	IF (U.GE.0.625) XR5=1.	RAN30360
	LXI,5	0	IF (U.LT.0.625) XR5=0.	RAN30370
	LA	SELECT,5		RAN30380
	STAD	*+2	SET LOAD COEFFICIENT LOCATION	RAN30390
	LXI,4	3		RAN30400
	FL	*-* ,4	LOAD COEFFICIENT	RAN30410
	FST	ME,4	A0,A1,B0,B1	RAN30420
	TXHED,4	1		RAN30430
	J	*-3		RAN30440
	FL	ME		RAN30450
	FM	X		RAN30460
	FA	ME+1		RAN30470
	FST	ME+1		RAN30480
	FL	RONE		RAN30490
	FS	X		RAN30500
	FA	ME+3		RAN30510
	FST	ME+3		RAN30520
	FL	ME+2		RAN30530
	FD	ME+3		RAN30540
	FA	ME+1		RAN30550
	FST	X		RAN30560
	TB0F,1	SIGN	IF POSITIVE SIDE	RAN30570
	J	*+4	YES EXIT	RAN30580

CHANGE SIGN X.

LX4	LAI	0		RAN30590
LX5	FS	X		RAN30600
	FST	X		RAN30610
	LXI,4	*--		RAN30620
	LXI,5	*--		RAN30630
	J*	RAN3	EXIT	RAN30640
* SELECT	NOP	TAB.00,,1		RAN30642
	NOP	TAB.01,,1		RAN30644
	NOP	TAB.02,,1		RAN30646
	NOP	TAB.03,,1		RAN30660
	NOP	TAB.04,,1		RAN30670
	NOP	TAB.05,,1		RAN30680
	NOP	TAB.06,,1		RAN30690
	NOP	TAB.07,,1		RAN30700
	NOP	TAB.08,,1		RAN30710
	NOP	TAB.09,,1		RAN30720
	NOP	TAB.10,,1		RAN30730
	NOP	TAB.11,,1		RAN30740
	NOP	TAB.12,,1		RAN30750
	NOP	TAB.13,,1		RAN30760
	NOP	TAB.14,,1		RAN30770
	NOP	TAB.15,,1		RAN30780
	NOP	TAB.16,,1		RAN30790
	NOP	TAB.17,,1		RAN30800
	NOP	TAB.18,,1		RAN30810
	NOP	TAB.19,,1		RAN30812
	NOP	TAB.20,,1		RAN30814
	NOP	TAB.21,,1		RAN30816
				RAN30817
* TAB.00	OCTAL	236517176402,540531106401,210303732367,545467047377		RAN30818
TAB.01	OCTAL	222234625402,542101220401,255476462374,402656132373		RAN30819
TAB.02	OCTAL	206472460402,540463155401,344251512375,233130166375		RAN30820
TAB.03	OCTAL	212550134402,537222635401,307225502375,207001346375		RAN30822
TAB.04	OCTAL	234022576402,525706166401,363320706374,245312200374		RAN30824
TAB.05	OCTAL	264540663402,503311071401,204306357374,270202072373		RAN30826
TAB.06	OCTAL	321324770402,452335733401,207776523373,304067575372		RAN30828
TAB.07	OCTAL	360036374402,415667146401,206741301372,313025205371		RAN30830
TAB.08	OCTAL	207444446403,567727026402,203005251371,316410761370		RAN30832
TAB.09	OCTAL	226672532403,550710105402,373125000367,317551601367		RAN30834
TAB.10	OCTAL	245514245403,532167756402,357161052366,317400656366		RAN30836
TAB.11	OCTAL	263635766403,514105476402,343260202365,316417371365		RAN30838
TAB.12	OCTAL	301241405403,476520770402,330111440364,315145403364		RAN30840
TAB.13	OCTAL	316141440403,461630002402,315677225363,313577155363		RAN30842
TAB.14	OCTAL	332364311403,445410422402,304455027362,312241062362		RAN30844
TAB.15	OCTAL	346164163403,431612233402,274172667361,310755031361		RAN30846
TAB.16	OCTAL	361373205403,416404021402,264573052360,307560712360		RAN30848
TAB.17	OCTAL	374241645403,403535654402,255772251357,306457451357		RAN30850
TAB.18	OCTAL	203276661404,574501044403,247707107356,305446477356		RAN30852
TAB.19	OCTAL	210321003404,567456747403,242265067355,304521143355		RAN30854
TAB.20	OCTAL	215217576404,562560167403,235235607354,303651603354		RAN30856
TAB.21	OCTAL	222002254404,555775517403,230540252353,303052673353		RAN30858
* X	OCTAL	0		RAN30860
CONST	OCTAL	303725614326		RAN30990
MASK	OCTAL	377777777777		RAN31000
ME	OCTAL	0,0,0,0,0		RAN31010
SIGN	OCTAL	0		RAN31020
				RAN31030

IONF	OCTAL	000000000001	RAN31040	
RONF	OCTAL	2000000000401	RAN31050	
*			RAN31052	
GR	LA	MGR	MESSAGE PRINT ROUTINE	RAN31060
	TRON,1	SIGN		RAN31070
	LA	PGR		RAN31080
	STA	X		RAN31090
	WD,22	X		RAN31100
	FWD	1		RAN31110
	J	REPEAT		RAN31120
PGR	HCODE	6,G. 5.0		RAN31130
MGR	HCODE	6,G.-5.0		RAN31140
	END			RAN31150

## 4. 文 献

- [1] 吉沢正 (1969)、Yates の算法を応用した擬似正規乱数について、  
日本オペレーションズリサーチ学会発表論文。
- [2] 山内二郎 (1965)、正規分布の百分率点の一次有理関数近似第6回ブ  
ログラミング・シンポジウム報告書。
- [3] 島山高、戸田英雄 (1969)、計算速度の速い  $\log$  の FORTRAN 副  
プログラム、電試彙報 Vol 33, NO.5, PP.548-554.
- [4] Hideo Toda (1967), An Optimal Rational Approxima-  
tion of Normal Deviates for Digital Computers,  
Bul. Electrotech. Lab. ; Vol.31 NO.12, PP.1259-1270.
- [5] M.E.Muller (1959), A comparison of Methods for Genera-  
ting Normal Deviates on Digital Computers, J.ACM.,  
Vol.6, PP.376-383.
- [6] Sibuya, M., Further Consideration on Normal Random  
Variable Generator. 14, 1962, PP.159-165.
- [7] D.Teichroew, Distribution Sampling with High Speed  
Computers. ph.D.Thesis, University of North Carolina,  
1953
- [8] G.E.P.Box and M.E.Muller, A Note on The Generation of  
Normal Deviates. Ann.Math.Stat.28 (1958), 610-611.

その後、渋谷正昭氏 ( I B M ) より下記の文献を教えて頂きました。

[9] R.Kronmal, Evaluation of a Pseudorandom Normal Number Generator, J.A.O.M, Vol. 11, NO.3 (July, 1964), PP.363-375.

[10] G.Marsaglia, Elementary Relations between Uniform and Normal Distributions in the plane. Mathematical Note, NO.268, Mathematics Research Laboratory (1962).

[11] G.Marsaglia, Generating Variable from the Tail of the Normal Distribution, Technometrics, Vol. 6 (1964) PP.101-102.