A Technique for Generating Correlated X-Band Weather Degradation Statistics

E. C. Posner and F. J. Zeigler
TDA Planning Office

Flight projects relying on the DSN's X-band receiving facilities for X-band telemetry and/or tracking require some technique for generating test cases of degradations to use in mission sequence planning exercises and even in data rate selection. It is, for example, known that the X-band noise temperature of DSN receivers can go up from 20 K to over 100 K or more, if the air is heavily laden with water vapor, although that is an uncommon occurrence. It is proposed that the DSN furnish flight projects relying on X-band degradation models, one for each DSN Complex. Such models would be in the form of a random process generator, say in an MBASIC program, that would permit the project to generate X-band degradation data with the right autocorrelations for periods of interest to the Projects. The autocorrelation modeling is especially important because bursts of degradation lasting several days can affect data storage and mission sequence design strategy. This article therefore presents one approach which works if the degradation statistics obey a half-gaussian law. That is, the random variables are formed by taking the absolute values of another set of random variables, themselves having a (two-sided) gaussian distribution. The technique of this paper then permits the half-gaussian random variables to have given one- and two-step correlation coefficients.

I. Introduction

Flight projects relying on the DSN's X-band receiving facilities for X-band telemetry and/or tracking require some technique for generating test cases of degradations to use in mission sequence planning exercises and even in data rate selection. It is, for example, known that the X-band noise temperature of DSN receivers can go up from 20 K to over 100 K or more, if the air is heavily laden with water vapor, although that is an uncommon occurrence. It is proposed that the DSN furnish flight projects relying on X-band degradation models, one for each DSN Complex. Such models would be in the form of a random process generator, say in an MBASIC program, that would permit the project to generate X-band degradation data with the right autocorrelation for periods of interest to the projects. The autocorrelation modeling is especially important because bursts of degradation lasting several days can affect data storage and mission sequence design strategy. This article therefore presents one approach which works if the degradation statistics obey a half-gaussian law. That is, the random variables are formed by taking the absolute values of another set of random variables, themselves having a (two-sided) gaussian distribution. The technique of this paper then permits the half-gaussian random variables to have given one- and two-step correlation coefficients, if non-negative. A modification, to be reported separately, can be used to match negative correlations.

Mathematically, the primary problem is to find correlation coefficients ρ_1, ρ_2 for the intermediate gaussian random variables which will lead to the desired correlations λ_1 and λ_2 for the half-gaussian variables. Thus, we need to find $\rho_1(\lambda_1)$, and $\rho_2(\lambda_2)$. The nature of the bivariate gaussian distribution makes the calculation of $\lambda(\rho)$ feasible. A two-step process is therefore followed: the function $\lambda(\rho)$ is found (see Section II), then is inverted to get $\rho(\lambda)$ by numerical means (Section III).

In Section IV we outline a method for generating a sequence of gaussian random variables with one- and two-step correlations ρ_1, ρ_2, from an uncorrelated set of normal gaussian random variables. The algorithm described in Section IV can easily be extended to the case of more than two correlations. Section V briefly describes a program written in the MBASIC language implementing the above algorithms, and it describes how to create a set of standard gaussian random numbers from a set of random variables distributed uniformly. Some sample output from the program is also given. Section VI is an informal "Software Specification Document" (SSD) for the program.

II. Determining the Intermediate Correlation

Given two normally distributed random variables X_1,X_2 with mean 0, variance 1 and correlation $\rho(X_1,X_2)$, we wish to find Corr $(|X_1|,|X_2|) = \lambda(\rho)$. Our starting point is the distribution function

$$F(a_1,a_2) = \int_{-\infty}^{a_1}\int_{-\infty}^{a_2} f(X_1,X_2,\rho)dX_1dX_2$$

for X_1,X_2, where f is the bivariable normal density centered at the origin:

$$f(X_1,X_2,\rho) = \frac{1}{2\pi\sqrt{1-\rho^2}} \times \exp\left[\frac{-1}{2(1-\rho^2)}(X_1^2 - 2\rho X_1X_2 + X_2^2)\right]$$

Let $G(a_1,a_2)$ denote the distribution function for $|X_1|,|X_2|$. If either a_1 or a_2 is less than 0 then $G(a_1,a_2) = 0$; otherwise we have

$$\begin{aligned} G(a_1,a_2) &= \rho(|X_1| \leq a_1, |X_2| \leq a_2) \\ &= \int_{-a_1}^{a_1}\int_{-a_2}^{a_2} f(X_1,X_2,\rho)dX_1dX_2 \\ &= \int_0^{a_1}\int_0^{a_2} 2[f(X_1,X_2,\rho) + f(X_1,X_2,-\rho)]dX_1dX_2 \end{aligned}$$

Therefore, $2[f(X_1,X_2,\rho) + f(X_1,X_2,-\rho)]$ represents the joint density of $|X_1|,|X_2|$

The next step is to calculate the expectation of $|X_1|,|X_2|$:

$$E(|X_1|,|X_2|$$

$$2\int_0^{\infty}\int_0^{\infty} X_1X_2[f(X_1,X_2,\rho) + f(X_1,X_2,-\rho)]dX_1dX_2$$
$$= 2\Phi(\rho) + 2\Phi(-\rho)$$

where

$$\Phi(\rho) = \int_0^{8}\int_0^{\infty} X_1X_2f(X_1,X_2,\rho)dX_1dX_2$$

A variable change to polar coordinates gives

$$\begin{aligned} \Phi(\rho) &= \frac{1}{4\pi\sqrt{1-\rho^2}}\int_0^{\infty}\int_0^{\pi/2} r^3\sin 2\theta \cdot \exp\left[\frac{-1}{2(1-\rho)}(r^2 - \rho r^2\sin 2\theta)\right]dr\,d\theta \\ &= \frac{1}{4\pi\sqrt{1-\rho^2}}\int_0^{\pi/2}\sin 2\theta\int_0^{\infty} r^3\cdot\exp\left(-r^2\cdot\frac{1-\rho\sin 2\theta}{2(1-\rho^2)}\right)dr\,d\theta \\ &= \frac{(1-\rho^2)^{3/2}}{2\pi}\int_0^{\pi/2}\frac{\sin 2\theta d\theta}{1-\rho\sin 2\theta} = \frac{\sqrt{1-\rho^2}}{2\pi} + \frac{\rho}{4} + \frac{\rho\arcsin\rho}{2\pi} \end{aligned}$$

Thus,

$$E(|X_1|,|X_2|) = \frac{2}{\pi}(\sqrt{1-\rho^2} + \rho \arcsin \rho)$$

For $\rho = 1$ we obtain $E(X_1^2) = 1$; $\rho = 0$ implies

$$[E|X_i|]^2 = \frac{2}{\pi}, \text{so} \quad \text{var}(X_i) = 1 - \frac{2}{\pi}$$

Therefore

$$\text{Corr}(|X_1|,|X_2|) = \frac{\frac{2}{\pi}(\sqrt{1-\rho^2} + \rho \arcsin \rho - 1)}{1 - \frac{2}{\pi}}$$

$$= \frac{\sqrt{1-\rho^2} + \rho \arcsin \rho - 1}{\frac{\pi}{2} - 1} = \lambda(\rho)$$

III. Calculation of $\rho(\lambda)$

We calculate $\rho(\lambda)$ given the relation

$$\lambda(\rho) = \frac{\sqrt{1-\rho^2} + \rho \arcsin \rho - 1}{\frac{\pi}{2} - 1}, \qquad (0 \leq \lambda \leq 1) \qquad (1)$$

A useful observation is $(\pi/2 - 1)\, d\lambda/d\rho = \arcsin \rho = \rho + 1/6\,\rho^3 + 3/40\,\rho^5 + \ldots$. Integrating this formula gives a Taylor series for $(\pi/2 - 1)\lambda$ and, upon formal inversion, we obtain:

$$\rho^2(\lambda) = 2\lambda_1 - 1/3\,\lambda_1^2 - 4/45\,\lambda_1^3 - 11/189\,\lambda_1^4 - 722/14175\,\lambda_1^5 - 0.05203\,\lambda_1^6 - 0.05087\,\lambda_1^7$$

where $\lambda_1 = (\pi/2 - 1)\lambda$. Over the interval [0,0.5] this seven-term series will find ρ to within the theoretical error bound of $2 \cdot 10^{-4}$. An actual computer calculation using this series, then converting back via Eq. (1) gives an accuracy to within 10^{-5} over [0,0.5] with the largest error at $\lambda = 0.5$. (The difference between the two numbers is simply an example of the fact that calculations often work out much better than theoretical error bounds would indicate.)

When λ approaches 1, the above series gives inaccurate results. An alternate method is to use the Lagrange Interpolation polynomial for ρ over the interval [0.5,1], with mesh points 0.5, 0.6,..,1. The result is $\rho(\lambda) = 0.2862 + 1.0558\lambda - 0.0470\lambda^2 - 0.9506\lambda^3 + 1.0072\lambda^4 - 0.3516\lambda^5$. The theoretical error bound is $7 \cdot 10^{-4}$, mainly due to rounding. A computer calculation using this formula gave accuracy to within $4 \cdot 10^{-4}$ over [0.5,1].

IV. Generating Random Variables With Given One and Two-Step Correlations

We now give a method for generating a sequence $\{X_i\}$ of gaussian random variables with given one-step correlation $\rho_1(X_i,X_{i+1})$ and two-step correlation $\rho_2(X_i,X_{i+2})$. Consider a process of the form

$$X_{i+2} = \alpha X_{i+1} + \beta X_i + y_{i+2} \qquad (1)$$

where α and β are coefficients to be determined, and the y_i are standard normal gaussian random variables.

Multiplying Eq. (1) by X_i,X_{i+1} and taking expectations, we find that $\rho_1\sigma_{i+2}^2 = \alpha\sigma_{i+1}^2 + \beta\rho_1\sigma_i^2$ and $\rho_2\sigma_{i+2}^2 = \alpha\rho_1\sigma_{i+1}^2 + \beta\sigma_i^2$ where $\sigma_i^2 = E(X_i^2)$. For Eq. (1) to be a stationary process it is required that $\sigma_i = \sigma_{i+k}$ for $k > 0$; in this case we may solve for α and β, giving $\alpha = \rho_1 - \rho_2\rho_2/1 - \rho_1^2$ and $\beta = \rho_2 - \rho_1^2/1 - \rho_1^2$. It is also possible to solve for $\sigma_i = \sigma$. Writing $E(X_{i+2} - \alpha X_{i+1} - \beta X_i)^2 = Ey_{i+2}^2 = 1$ we find $\sigma^2 = 1 - \rho_1^2/(1 - \rho_2)(1 + \rho_2 - 2\rho_1^2)$. Since $\sigma^2 \geq 0$ this gives us the inequality $1 + \rho_2 - 2\rho_1^2 \geq 0$.

To begin generating $\{X_i\}$ we normalize X_1,X_2 using the above value for σ as follows:

$$X_1 = \sigma\sqrt{\frac{1+\rho_1}{2}}\,y_1 + \sigma\sqrt{\frac{1-\rho_1}{2}}\,y_2,$$

$$X_2 = \sigma\sqrt{\frac{1+\rho_1}{2}}\,y_1 - \sigma\sqrt{\frac{1-\rho_1}{2}}\,y_2$$

This gives the proper correlation and variance for X_1,X_2. Equation (1) may now be used to calculate $X_3,X_4,\ldots$.

V. Program Organization

A program has been written in the MBASIC[1] language to carry out the generation of a sequence of correlated "half gaussian" random variables, relying on the procedures described above.

The MBASIC random number generator gives numbers which are uniformly distributed, so the first step is to produce a sequence of standard normal gaussian numbers from the uniform distribution, via the central limit

[1]The DSN standard nonreal-time language.

theorem. If y_i is the ith standard gaussian, X_K the Kth random number from the MBASIC generator, we may write

$$y_i = \sum_{K=1} (X_{12(i-1)+K} - {}^{1/2})$$

A uniform distribution over [0,1] has mean 1/2, variance 1/12, so each $X_K - {}^{1/2}$ has mean 0, and adding 12 of them gives a random variable with variance 1. In addition, the y_i will be very nearly gaussian, as a result of the central limit theorem.

Using the numerical methods described above, the next step in the program is to calculate the intermediate correlations. Then the stationary random process is used (if the correlations satisfy the required inequality) to generate random variables with the intermediate correlations. The absolute values of these random variables are then taken to give our desired sequence of "half gaussian" random variables with the given correlations.

A word of caution. We have not yet proved that any non-negative two-step correlation function $\{\lambda_1, \lambda_2\}$ which can arise as the correlation function of some stationary process whose 1-dimensional marginals have the same distribution as the absolute value of a centered gaussian can also arise as the two-step correlation function of the absolute value of a centered jointly (the key word) gaussian process. We can prove, however, that this is the case provided.

$$\lambda_1 \leq \lambda\left(\frac{1}{\sqrt{2}}\right) \simeq 0.4598$$

Since in most applications λ_1 and λ_2 would be reasonably close to 0, this gap should not arise in practice, if indeed it can arise at all.

A typical run of the program is shown in Fig. 1.

VI. Program Description

A. General Description

This section is essentially a Software Specification Document (SSD) for an MBASIC program CORGS2. The purpose of CORGS2 is the generation of a set of "half-gaussian" random variables (the elements of which are formed by taking the absolute values of the members of a set with a normal distribution) with given one- and two-step correlations. The subsequent numbering refers to module numbers in the structured flowcharts. Three main steps are involved. First, a set of random variables (normal distribution) is produced with mean 0, variance 1. Then these are used to generate a new set of gaussian random variables with known one- and two-step correlations, via a stationary gaussian stochastic process. Finally, the absolute values of these numbers are taken (with the appropriate scaling factor introduced to give the correct standard deviation) to produce our set of half-gaussian random numbers.

B. Level 1 Detailed Description

1.1

This module has two purposes. The first is the declaration of the subroutine addresses for the subprograms of CORGS2. Then input is obtained for N, the number of half-gaussian random variables to be produced by the user response to a prompting message.

Variables:

N: user response to the prompting message, the number of half-gaussians to be generated.

1.2 (ASSIGN)

In this module all necessary variables are declared and the random number generator is initialized to some positive value in order to insure a repeatable sequence of random variables.

1.3 (INPT)

Input is obtained for LAMBDA(1), LAMBDA(2) (the one- and two-step correlations) and sigma (the final scaling factor) by means of user response to prompting messages.

Variables

LAMBDA(1): The one-step correlation of the half-gaussian random variables.

LAMBDA(2): The two-step correlation of the half-gaussian random variables.

SIGMA: The scaling factor which determines the final standard deviation.

1.4 (INDGS)

In this module we generate an array (G01) of gaussian random variables with mean 0, variance 1, by means of two nested loops.

Variable:

G01:	A numeric array variable of length N consisting of uncorrelated standard normal random numbers.

1.5 (CORGS)

An array (GRHO) of random numbers is generated with intermediate one-step correlation RHO(1), two-step correlation RHO(2).

Variable:

GRHO:	A numeric array of length N consisting of random variables with mean 0, variance 1 and one-step correlation RHO(1), two-step correlation RHO(2).

1.6 (HALFGS)

An array of half-gaussian random numbers (GHALF) is produced with one-step correlation LAMBDA(1), two-step correlation LAMBDA(2).

Variable:

GHALF:	A numeric array variable of length N, the "half-gaussian" random variables.

1.7 (PRNT)

A prompting question determines whether the user wants a printout of the array GHALF of half-gaussian random numbers; with an affirmative answer the array GHALF is printed.

Variables:

ANS$:	A simple string variable accepting only the initial character of the user's response. ANS$='Y' means output is desired, ANS$='N' means the opposite.
OK:	A simple numeric variable, the condition of a correct user response to the prompting question.

See Fig. 2 for flowchart.

C. Level 2 Detailed Design

1.1

The subroutine addresses are first declared as follows: ASSIGN = 100200, INPT = 100300, INDGS = 100500, CORGS = 100600, HALFGS = 100800, PRNT = 100900.

Then a prompting message is printed asking for the number of random variables needed. The message is "ENTER N, THE NUMBER OF CORRELATED HALF-GAUSSIAN RANDOM NUMBERS DESIRED:".

Variable defined

N, a simple numeric variable input in 1.1.

1.2 ASSIGN Detailed Design: Level 2

The purpose of ASSIGN is to declare all necessary variables and to initialize the random number generator at a positive value to produce a repeatable sequence of random numbers.

1.2.1 Declare variables with explicit declarations. The declaration of the numeric variables is: Real G01(N), X(N), GRHO(N), GHALF(N), LAMBDA(2), RHO(2), SIGMA, ADJ, OK, R, S. The string declaration is: STRING ANS$:1.

Variable definitions

G01 a numeric array of length N generated in 1.4.1–1.4.3; X, a numeric array of length N generated in 1.5.8–1.5.10; GHALF, a numeric array of length N (consisting entirely of positive numbers) calculated in 1.6.1–1.6.2; LAMBDA, a numeric array of length 2 input in 1.3.1–1.3.2; SIGMA, a simple numeric variable input in 1.3.3; RHO, a numeric array of length 2 calculated in 1.5.1–1.5.4; ADJ, a simple numeric variable defined in 1.5.3; OK, a simple numeric variable defined in 1.7.1 and reevaluated in 1.7.4; ANS$, a simple string variable input in 1.6.3.

1.2.2 Initialize the random number generator with the declaration: RANDOMIZE 518997. Any positive number may be used, but the following should be noted. Large positive numbers seem to give the best results, so six digits are advisable. Also, they should be chosen by some random means, which in this case consisted of choosing cards from a deck, face cards removed, and sampling with replacement.

See Fig. 3 for flowchart.

1.3 INPT Detailed Design: Level 2

In this module we obtain from user, via a series of prompting messages, values which will determine the statistical characteristics of the half-gaussian random variables.

Inputs

A. LAMBDA(1), a numeric variable input in 1.3.1 which must satisfy $0 \leq \text{LAMBDA}(1) \leq 1$.

B. LAMBDA(2), a numeric variable input in 1.3.2 which must also satisfy $0 \leq \text{LAMBDA}(2) \leq 1$.

C. SIGMA, a numeric variable input in 1.3.3 which must be positive.

Outputs

A. Prompting message "ENTER LAMBDA(1), THE ONE-STEP CORRELATION OF THE HALF-GAUSSIAN RANDOM NUMBERS (NOTE THAT LAMBDA(1) MUST BE NON-NEGATIVE AND LESS THAN 1):"

B. Prompting message "ENTER LAMBDA(2), THE TWO-STEP CORRELATION OF THE HALF-GAUSSIAN RANDOM NUMBERS (NOTE THAT LAMBDA(2) MUST BE NON-NEGATIVE AND LESS THAN 1:"

C. Prompting message "ENTER SIGMA, THE STANDARD DEVIATION OF THE FULL-GAUSSIAN DISTRIBUTION (NOTE THAT SIGMA MUST BE POSITIVE):"

1.3.1 Print prompting message A to user asking for the final one-step correlation.

Variable defined

LAMBDA(1), a numeric variable input in 1.3.1.

1.3.2 Print prompting message B to user asking for the final two-step correlation.

Variable defined

LAMBDA(2), a numeric variable input in 1.3.2.

1.3.3 Print prompting message C to user asking for the final scaling factor.

Variable defined

SIGMA, a simple numeric variable input in 1.3.3. The standard deviation of the half-gaussian random variables should be approximately $\text{SIGMA}*\sqrt{(1-2/\pi)}$.

See Fig. 4 for flowchart.

1.4 INDGS Detailed Design: Level 2

In this module an array (G01) of gaussian random variables is generated with mean 0, variance 1, by means of two nested loops.

1.4.1 The index I is first initialized to 1. At each execution of the loop it is determined whether $I>N$; if not, control is passed on to 1.4.2; if the inequality holds, then control is transferred to 1.5.1.

1.4.2 This loop starts with $J=1$; if $J\leq 12$ control is passed to 1.4.3 then back to 1.4.2 where J is incremented by 1, if $J>12$ control is transferred to 1.4.1.

1.4.3 A random variable is produced with mean 0, variance 1. From an MBASIC random number (RNDM) 1/2 is subtracted to produce a random variable with mean 0, variance 1/2. Each time control is passed from 1.4.2 another of these random variables is added on to the previous result, which increments the variance by 1/12; the basic step is $\text{G01(I)}=\text{G01(I)}+\text{RNDM}-1/2$. After control is passed to 1.4.3 for the twelfth time, G01 will have variance 1 (and still have mean ϕ). Control is then returned to 1.4.1.

Variable definition

G01, a numeric array variable generated in 1.4.1–1.4.3.

See Fig. 5 for flowchart.

1.5 CORGS Detailed Design: Level 2

The purpose of CORGS is to produce an array (GRHO) with proper one- and two-step correlations so that when the absolute values of the elements of GRHO are taken, the new array has our desired correlations LAMBDA(1) and LAMBDA(2). Therefore we must first determine the correct intermediate correlations (RHO(1) and RHO(2)) which will lead to the final correlations (LAMBDA(1) and LAMBDA(2)), then produce GRHO using the array G01 by means of a stationary stochastic process.

It is possible that the numbers RHO(1) and RHO(2) will not lead to a stationary process; if they do not, an error message will be printed and the program terminated.

Outputs

A. Error message: "LAMBDA(1) AND LAMBDA(2) ARE NOT ACCEPTABLE CORRELATIONS"

B. TERMINATING MESSAGE "CORGS2 TERMINATED"

C. "THE INTERMEDIATE ONE-STEP CORRELATION IS:"

D. "THE INTERMEDIATE TWO-STEP CORRELATION IS:"

1.5.1 Initialize the index I to 1 and pass control to 1.5.2. Control will be passed back to 1.5.1 from 1.5.3 and I is incremented by 1 at that stage; then if I>N, control is transferred to 1.5.4; if not, control is again passed on to 1.5.2.

1.5.2 The correlation RHO(I) is determined for the interval 0≤LAMBDA(I)≤1/2 by the formula RHO(I)=SQR(2*ADJ**2–(4/45)*ADJ**3–(11/189)*ADJ**4–(722/14175)*ADJ**5–(.05203)*ADJ**6–(.05087)*ADJ**7) where ADJ=LAMBDA(I)*(π/2–1) (this is the first seven terms of a Taylor series). Control is passed to 1.5.3.

Variable definitions

LAMBDA(I), a numeric variable input in 1.3.1 or 1.3.2; RHO(I), a numeric variable calculated in 1.5.1–1.5.3; ADJ, a simple numeric variable defined in 1.5.2.

1.5.3 RHO(I) is calculated over the interval .5<LAMBDA(I)≤1 by the formula RHO(I)=.2862+1.0558*LAMBDA(I)–.0470*LAMBDA(I)**2–.9506*LAMBDA(I)**3+1.0072*LAMBDA(I)**4–.3516*LAMBDA(I)**5. This is the "Lagrange interpolation polynomial" for RHO(I) of degree 5 over [.5,1] with equally spaced points. Control is passed to 1.5.1.

Variable definitions

RHO(I), a numeric variable calculated in 1.5.1–1.5.3; LAMBDA(I), a numeric variable input in 1.3.1 or 1.3.2.

1.5.4 We now use the calculated values of RHO(1), RHO(2), and the array G01 to produce an array GRHO of correlated random variables. This is done by means of a stationary stochastic process and an intermediate array, X, which will give us our array GRHO when its members are normalized. However, if 1+RHO(2)–2*RHO(1)**2≤0 then the stochastic scheme will not be stationary. In this case, we cannot generate an array of half-gaussians corresponding to the values of LAMBDA originally input, so control is passed to 1.5.5 and the program is terminated. If the correlations are valid (i.e., the above inequality does not hold) then control goes to 1.5.6.

Variable definition

RHO(I), a numeric variable calculated in 1.5.1–1.5.3.

1.5.5 Print error messages A and B, then terminate program.

Variable defined

LAMBDA, a numeric variable input in 1.3.1 and 1.3.2.

1.5.6 Print messages C and D along with RHO(1), and RHO(2) to four-decimal-place accuracy.

Variable defined

RHO, a numeric array variable calculated in 1.5.1–1.5.3.

1.5.7 The first two random variables (G01(1) and G01(2) produced in 1.4.3 must be normalized to begin the stationary stochastic process which will lead to the correlated gaussian array GRHO. We set X(1)=R*G01(1)+S*G01(2), X(2)=R*G01(1)–S*G01(2) where

$$R=\sqrt{(1-RHO(1)**2)*(1+RHO(1))}/\sqrt{2*(1-RHO(2))*(1+RHO(2)-2*RHO(1)**2)}$$

and

$$S=\sqrt{(1-RHO(1)**2)*(1-RHO(1))}/\sqrt{2*(1-RHO(2))*(1+RHO(2)-2*RHO(1)**2.}$$

Variable definitions

G01, a numeric array generated in 1.4; GRHO, a numeric array generated in 1.5.10–1.5.11; X, a numeric array generated in 1.5.7–1.5.9; R, S, simple numeric variables defined in 1.5.7.

1.5.8 This loop generates the rest of the array X. The index I is first set to 3, it is increased by one every time control returns from 1.5.9. If I>N control is transferred to 1.5.10, if not, control passes to 1.5.9.

Variable defined

X, a numeric array generated in 1.5.7–1.5.9.

1.5.9 The elements of X are calculated according to the formula X(I)=(RHO(1)*(1–RHO(2))/(1–RHO(1)**2))*X(I–1)+((RHO(2)–RHO(1)**2)/(1–RHO(1)**2))*X(I–2)+G01(I). Control is passed to 1.5.8.

Variable definitions

X, a numeric array generated in 1.5.7–1.5.9; RHO, a numeric array calculated in 1.5.1 to 1.5.3; G01, a numeric array generated in 1.4.

1.5.10 This loop generates the array GRHO. I is set to 1, and each time control is returned to 1.5.11 from 1.5.12 it is increased by 1. If I>N, control passes to 1.6.1; if not, control goes to 1.5.12.

Variable defined

GRHO, a numeric array generated in 1.5.10–1.5.11.

1.5.11 The elements of GRHO are calculated by the formula

$$\text{GRHO(I)} = \sqrt{(1-\text{RHO(2)})*(1+\text{RHO(2)}-2*\text{RHO(1)}**2)*}$$
$$(1/\sqrt{1-\text{RHO(1)}**2)*\text{X(I)}}$$

Control returns to 1.5.10.

Variable definitions

X, a numeric array generated in 1.5.7–7.5.9; GRHO, a numeric array generated in 1.5.10–1.5.11.

See Fig. 6 for flowchart of 1.5.

1.6 HALFGS Detailed Design: Level 2

In this module we generate the set of half-gaussian random variables with one-step correlation LAMBDA(1), two-step correlation LAMBDA (2). This set will have standard deviation SIGMA*$\sqrt{1-2/\pi}$ after multiplication by the scaling factor SIGMA.

1.6.1 This loop generates the set of half-gaussians. I is set to 1 then increased by 1 when control returns from 1.6.2. If I > N, control goes to 1.7.1, otherwise control passes to 1.6.2.

1.6.2 Each half-gaussian is calculated by the formula GHALF(I)=ABS(SIGMA*GRHO(I)).

Variable definitions

SIGMA, a simple numeric variable input in 1.2.3; GRHO, a numeric array generated in 1.5.11–1.5.12; GHALF, a numeric array generated in 1.6.

See Fig. 7 for flowchart.

1.7 PRNT Detailed Design: Level 2

This module prints out the array GHALF when required by the user.

Input

ANS$, a simple string variable accepting only 1 character. If ANS$='Y', the array GHALF is printed. If ANS$='N', the array is not printed. If ANS$≠'Y' or 'N' an error message is given and the loop is repeated until a correct response is given by the user.

Outputs

A. Prompting message "DO YOU WANT A PRINTOUT OF THE CORRELATED HALF-GAUSSIAN RANDOM NUMBERS? (ANSWER YES OR NO):"

B. Error message "ONLY YES OR NO ANSWERS, PLEASE"

C. Optional message "THE CORRELATED HALF-GAUSSIAN RANDOM NUMBERS ARE"\I, GHALF(I) for I=1 to N (GHALF is the array of half-gaussians)

D. Terminating message "END OF CORGS2"

1.7.1 We set OK=0 and let this be the condition that an incorrect user response has been given (something other than 'Y' or 'N').

Variable definition

OK, a simple numeric variable defined in 1.7.1 and reevaluated in 1.7.4.

1.7.2 We set LOOP=1 and ask if OK=0. If OK=0 then control passes to 1.7.3, otherwise control is transferred to 1.7.7.

Variable definition

OK, a simple numeric variable defined in 1.7.1 and reevaluated in 1.7.4.

1.7.3 Give prompting message A asking whether the user wants a printout of the array GHALF.

Variable definition

GHALF, a numeric array generated in 1.6.

1.7.4 We set OK=1*(ANS$='Y')+2*(ANS$='N'). If OK=0 is still true, the loop will be repeated because a correct user response to A has not been given.

Variable definitions

OK, a simple numeric variable defined in 1.7.1 and reevaluated in 1.7.4; ANS$, a simple string variable input in 1.7.3.

1.7.5 Determine if OK=0. If it does, pass control to 1.7.6; if not, control goes to 1.7.2.

Variable definition

OK, a simple numeric variable defined in 1.7.1 and reevaluated in 1.7.4.

1.7.6 Print error message B and return control to 1.7.2.

1.7.7 If OK=1, control passes to 1.7.8; if OK=2, control goes to 1.7.9.

Variable definition

OK, a simple numeric variable defined in 1.7.1 and reevaluated in 1.7.4.

1.7.8 Print message C, with the array GHALF given to four decimal places.

Variable definition

GHALF, a numeric array generated in 1.6.

1.7.9 Print terminating message D.

See Fig. 8 for flowchart.

```
>RUN

ENTER N, THE NUMBER OF CORRELATED HALF GAUSSIAN NUMBERS DESIRED: 10

ENTER LAMBDA(1), THE ONE-STEP CORRELATION OF THE HALF GAUSSIAN RANDOM
NUMBERS (NOTE THAT LAMBDA(1) MUST BE NON-NEGATIVE AND LESS THAN 1): .53

ENTER LAMBDA(2), THE TWO-STEP CORRELATION OF THE HALF GAUSSIAN RANDOM
NUMBERS (NOTE THAT LAMBDA(2) MUST BE NON-NEGATIVE AND LESS THAN 1): .28

ENTER SIGMA, THE STANDARD DEVIATION OF THE FULL GAUSSIAN DISTRIBUTION (NOTE
THAT SIGMA MUST BE POSITIVE): 3.1

THE INTERMEDIATE ONE-STEP CORRELATION IS: 0.7558

THE INTERMEDIATE TWO-STEP CORRELATION IS: 0.5574

DO YOU WANT A PRINTOUT OF THE CORRELATED HALF GAUSSIAN DISTRIBUTION?
(ANSWER YES OR NO): YES

THE CORRELATED HALF GAUSSIAN RANDOM NUMBERS ARE

   1     .7006
   2    1.1159
   3     .6289
   4    2.0626
   5    1.7033
   6    2.6059
   7    3.6343
   8    1.8570
   9    3.0022
  10    2.3408

END OF CORGS2
```

Fig. 1. Typical CORGS2 run

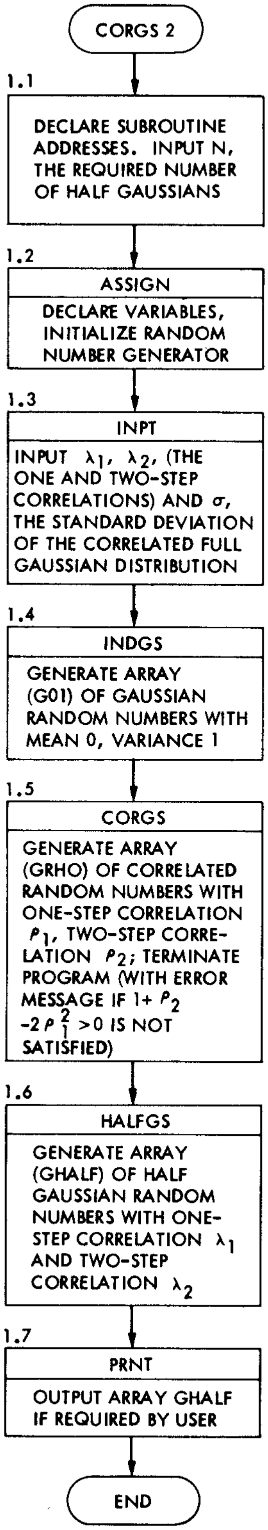

Fig. 2. CORGS2 Level 1 flowchart

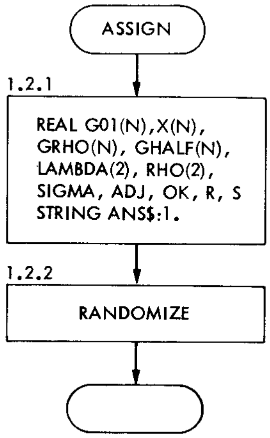

Fig. 3. Module 1.2 flowchart

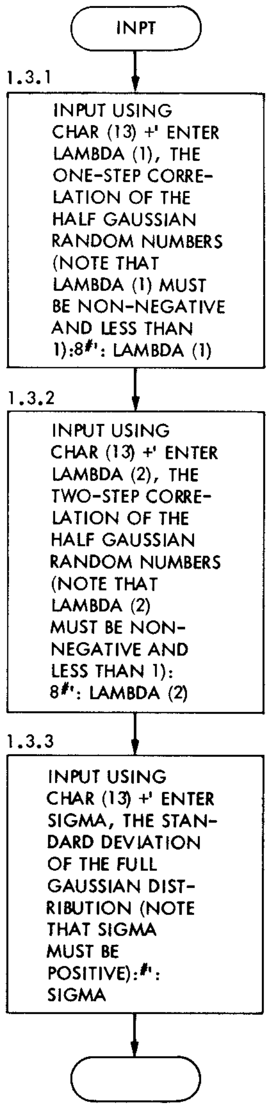

Fig. 4. Module 1.3 flowchart

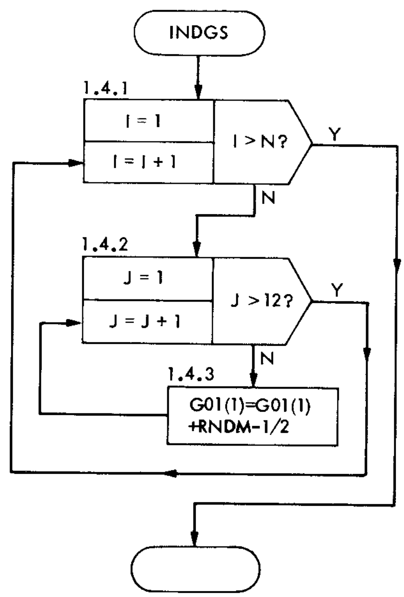

Fig. 5. Module 1.4 flowchart

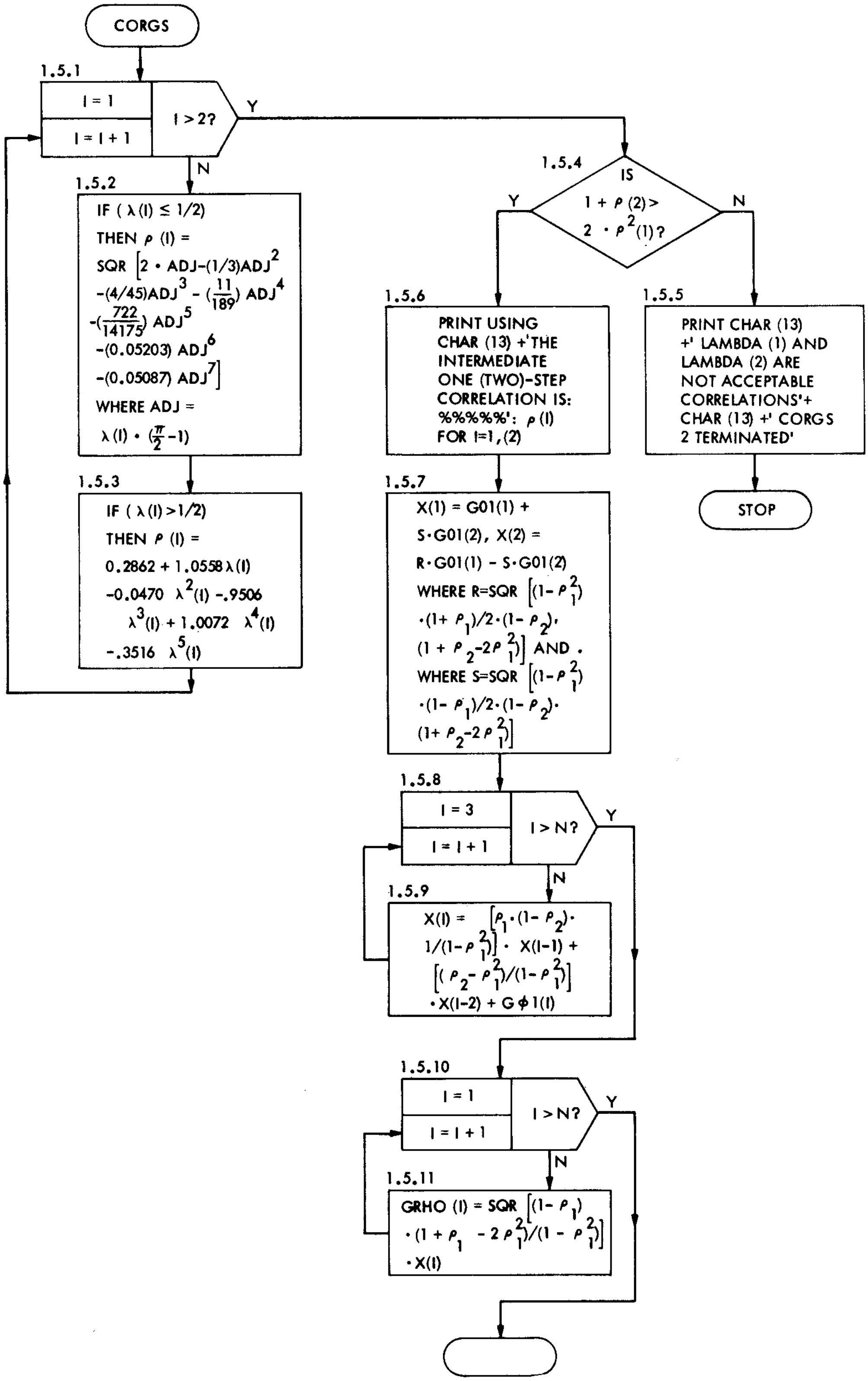

Fig. 6. Module 1.5 flowchart

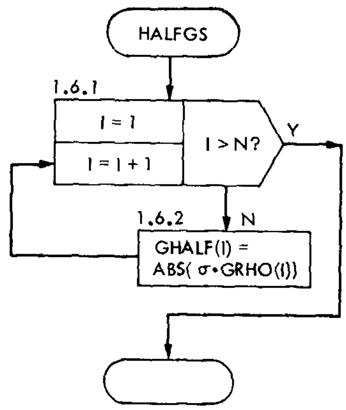

Fig. 7. Module 1.6 flowchart

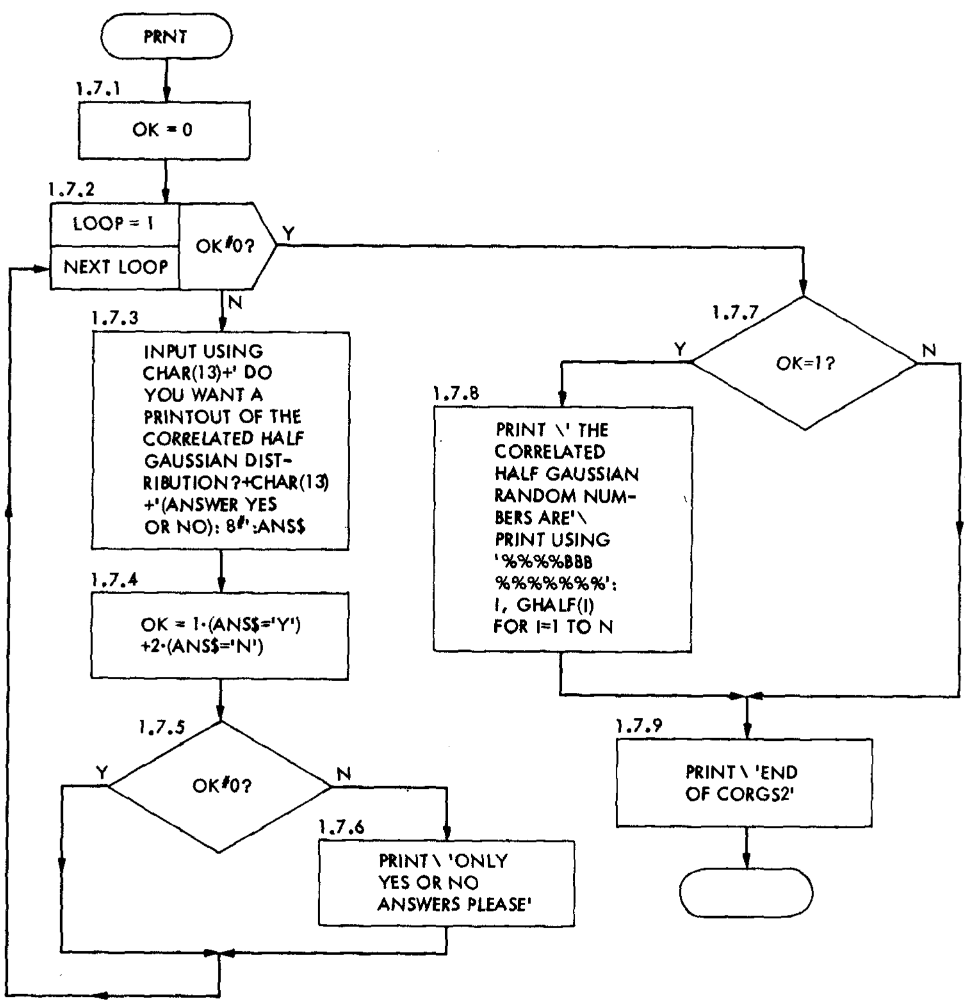

Fig. 8. Module 1.7 flowchart

```
100010  !CORGS2---MODULE 1

100020     ASSIGN=100200, INPT=100300, INDGS=100500, CORGS=100600,
              HALFGS=100800, PRNT=100900
100030     INPUT USING CHAR(13)+CHAR(13)+'ENTER N, THE NUMBER OF CORRELATED'
              +' HALF GAUSSIAN NUMBERS DESIRED:#  ':N       !MODULE #1.1

100040     GOSUB ASSIGN        !MODULE #1.2
100050     GOSUB INPT          !MODULE #1.3
100060     GOSUB INDGS         !MODULE #1.4
100070     GOSUB CORGS         !MODULE #1.5
100080     GOSUB HALFGS        !MODULE #1.6
100090     GOSUB PRNT          !MODULE #1.7

100100     END                 !END MODULE #1


100200  !ASSIGN-DECLARE VARIABLES; INITIALIZE RANDOM NUMBER GENERATOR
           ---MODULE #1.2

100210     REAL G01 (N), X(N), GRHO(N), GHALF(N), LAMBDA(2), RHO(2), SIGMA
              ADJ, OK, R, S
100220     STRING ANS$:1       !MODULE #1.2.1

100230     RANDOMIZE 518997    !MODULE #1.2.2

100240     RETURN              !END MODULE #1.2


100300  !INPT-ENTER PROGRAM PARAMETERS---MODULE #1.3

100310     INPUT USING CHAR(13)+'ENTER LAMBDA(1), THE ONE-STEP CORRELATION'
              +' OF THE HALF GAUSSIAN RANDOM'+CHAR(13)+'NUMBERS (NOTE THAT'
              +' LAMBDA(1) MUST BE NON-NEGATIVE AND LESS THAN 1): 8#':LAMBDA(1)
              !MODULE #1.3.1

100320     INPUT USING CHAR(13)+'ENTER LAMBDA(2), THE TWO-STEP CORRELATION'
              +' OF THE HALF GAUSSIAN RANDOM'+CHAR(13)+'NUMBERS (NOTE THAT'
              +' LAMBDA(2) MUST BE NON-NEGATIVE AND LESS THAN 1): 8#':LAMBDA(2)
              !MODULE #1.3.2

100330     INPUT USING CHAR(13)+'ENTER SIGMA, THE STANDARD DEVIATION OF THE'
              +' FULL GAUSSIAN DISTRIBUTION (NOTE'+CHAR(13)+'THAT SIGMA MUST'
              +' BE POSITIVE):  #':SIGMA      !MODULE #1.3.3

100340     RETURN      !END MODULE #1.3
```

Fig. 9. **CORGS2 listing**

```
100500   !INDGS-GENERATE ARRAY OF RANDOM NUMBERS (NORMAL DISTRIBUTION)
          WITH MEAN 0, VARIANCE 1---MODULE#1.4

100510    G01(I)=G01 (I)+RNDM-1/2 FOR J=1 TO 12  FOR I=1 TO N
              MODULES #1.4.1,.2,.3

100520    RETURN      !END MODULE #1.4


100600   !CORGS-GENERATE ARRAY OF RANDOM NUMBERS (NORMAL DISTRIBUTION)
          WITH MEAN 0, VARIANCE 1; ALSO WITH ONE-STEP CORRELATION RHO(1),
          AND TWO-STEP CORRELATION RHO(2)---MODULE #1.5

100610    FOR I=1 TO 2        MODULE#1.5.1

100620      IF (LAMBDA(I)<=1/2) THEN
              RHO(I)=SQR(2*ADJ-(1/3)*ADJ**2-(4/45)*ADJ**3-(11/189)*ADJ**4
              -(722/14175)*ADJ**5-(.05203)*ADJ**6-(.05087)*ADJ**7)
               WHERE ADJ=LAMBDA(I)*(PI/2-1)      !MODULE #1.5.2

100630      IF (LAMBDA(I)>1/2) THEN
              RHO(I)=.2862+1.0558*LAMBDA(I)-.0470*LAMBDA(I)**2
              -.9506*LAMBDA(I)**3+1.0072*LAMBDA(I)**4-.3516*LAMBDA(I)**5
               MODULE #1.5.3

100640    NEXT I

100650    IF (1+RHO(2)-2*RHO(1)**2<=0) THEN STOP CHAR(13)+
           'LAMBDA(1) AND LAMBDA(2) ARE NOT ACCEPTABLE CORRELATIONS'+CHAR(13)+
           'CORGS2 TERMINATED'      !MODULES #1.5.4,.5

100660    PRINT USING CHAR(13)+'THE INTERMEDIATE ONE-STEP CORRELATION IS:'
           +' %.%%%%':RHO(1)
100670    PRINT  USING CHAR(13)+'THE INTERMEDIATE TWO-STEP CORRELATION IS:'
           +'%.%%%%:RHO(2)         !MODULE #1.5.6

100680    R=SQR((1-RHO(1)**2)*(1+RHO(1))/2*(1-RHO(2))*(1+RHO(2)-2*RHO(1)**2))
100690    S=SQR((1-RHO(1)**2)*(1-RHO(1))/2*(1-RHO(2))*(1+RHO(2)-2*RHO(1)**2))
100700    X(1)=R*G01(1)+S*G01(2)
100710    X(2)=R*G01(1)-S*G01(2)      !MODULE #1.5.7

100720    X(I)=G01(I)+(RHO(1)*(1-RHO(2))/(1-RHO(1)**2))*X(I-1)+
           ((RHO(2)-RHO(1)**2)/(1-RHO(1)**2))*X(I-2)  FOR I=3 TO N
             MODULES #1.5.8,.8

100730    GHRO(I)=SQR((1-RHO(2))*(1+RHO(2)-2*RHO(1)**2)/(1-RHO(1)**2))*X(I)
            FOR I=1 TO N   !MODULES #1.5.10,.11

100740    RETURN      !END MODULE #1.5
```

Fig. 9 (contd)

```
1000800   HALFGS-GENERATE ARRAY OF CORRELATED HALF GAUSSIAN RANDOM NUMBERS
           ---MODULE #1.6

100810     GHALF(I)=ABS(SIGMA*GRHO(I))  FOR I=1 TO N

100820     RETURN      !END MODULE #1.6


100900    PRNT-GENERATES OUTPUT WHEN REQUESTED---MODULE #1.7

100910     FOR LOOP=1 UNTIL OK WHERE OK=0       MODULES #1.7.1,.2

100920        INPUT USING CHAR(13)+'DO YOU WANT A PRINTOUT OF THE CORRELATED'
               +' HALF GAUSSIAN DISTRIBUTION?'+CHAR(13)+'(ANSWER YES OR'
               +' NO):  #':ANS$      !MODULE #1.6.3

100930        OK=1*(ANS$='Y')+2*(ANS$='N')      !MODULE #1.6.4
100940        IF (NOT OK) THEN PRINT\'ONLY YES OR NO ANSWERS, PLEASE'
                 MODULES 1.7.5,.6

100950     NEXT LOOP      !MODULE #1.7.2

100960     IF OK=1 THEN PRINT\' THE CORRELATED HALF GAUSSIAN RANDOM NUMBERS ARE'\
              ELSE GO TO 100980

100970     PRINT USING'%%%%  %%%.%%%%':I, GHALF(I)  FOR I=1 TO N
               MODULES #1.7.7,.8

100980     !DECISION COLLECTOR NODE FOR 100960

100990     PRINT\'END OF CORGS2'      !MODULE #1.7.9

101000     RETURN      !END MODULE #1.7
```

Fig. 9 (contd)

NASA—JPL—Coml., L.A., Calif.