NEW LIMITATION CHANGE

TO
Approved for public release, distribution unlimited

FROM
Distribution authorized to U.S. Gov't. agencies and their contractors; Administrative/Operational Use; 5 Aug 1963. Other requests shall be referred to Center for Naval Analysis, Operations Evaluation Group, Washington, DC.

AUTHORITY
CNA D/N ltr, 13 Sep 1971
This is an interim report of continuing research. It does not necessarily represent the opinion of OEG or the U.S. Navy. It may be modified or withdrawn at any time.

Interim Research Memorandum

OPERATIONS EVALUATION GROUP
Center for Naval Analyses
WASHINGTON 25, D.C.
This is an interim report of continuing research. It does not necessarily represent the opinion of OEG or the U.S. Navy. It may be modified or withdrawn at any time.
This memorandum presents a usage manual for an IBM 7090 computer program which uses a Monte Carlo simulation to determine the probability of destroying a rectangular target with air-to-ground gunnery. The effect of correlation of successive aim points is considered. It is assumed that the aim point is distributed with a bivariate normal aiming error and that the individual rounds are distributed with an independent bivariate normal ballistic dispersion. A flow chart, a listing of the FORTRAN program and a sample problem are included.
I. INTRODUCTION

Hitherto, models for air-to-ground gunnery have been of two kinds. The first assumes that the aim points for successive rounds are uncorrelated and independent, i.e. that the correlation of successive aim points is zero. The second model is the salvo type which assumes that successive aim points are completely correlated. It is well known that both of these models are unsatisfactory descriptions of air-to-ground gunnery since the aim point does wander and the correlation is neither 0 nor 1, but some intermediate value. These two models have been used because the more general case defies simple mathematical analyses. However, with the introduction of high speed computers and Monte Carlo techniques, it is possible to simulate the more general case.

II. GENERAL DESCRIPTION

This program is a Monte Carlo simulation for use on the IBM 7090 computer. It determines the probability of destroying a rectangular target with air-to-ground gunnery on the basis of the following assumptions:

a. The aiming error and the ballistic dispersion are bivariate normal and are independent in the range coordinate (along the flight path of the aircraft) and the deflection coordinate (normal to the flight path in the horizontal plane).

b. There is a conditional kill probability (probability of kill given a hit) associated with each round, and there is no cumulative damage effect.

c. The correlation coefficient between the n-th and m-th round is \(|n-m| a\) in the range coordinate and \(|n-m| b\) in the deflection coordinate where a and b represent the correlation constants (Note: when a and b are zero or one, the process degenerates into the independent or salvo cases, respectively).

d. The mean aim point is at the center of the target which is rectangular and is located with one axis parallel to the flight path.

In addition to the parameters specifying the target size, aiming and ballistic dispersion, correlation and conditional kill probability, the program inputs include the aircraft speed, the firing rate of the gun, the slant range from the target at the commencement of the firing run, the maximum number of rounds per pass, the number of Monte Carlo runs to be made and the increment in the number of rounds at which target kill probabilities should be determined.

III MATHEMATICAL DERIVATION

Let \(x_n\) and \(y_n\) be random variables denoting the position of the aim point of the n-th round in the plane normal to the line of sight between the aircraft and the target. Let \(\sigma_n\) and \(\beta_n\) be independent Gaussian random variables with standard deviation 1 and mean 0. Let

\[ x_1 = \sigma x_1 \]  

(1a)
where \( a, b, \sigma_x, \) and \( \sigma_y \) are constants.

Equation (1) may be rewritten so that

\[
x_n = \sigma_x (a^{n-1} \alpha_1 + (1-a^2)^{1/2} \sum_{i=2}^n a^{n-i} \alpha_i)
\]

\[
y_n = \sigma_y (b^{n-1} \beta_1 + (1-b^2)^{1/2} \sum_{i=2}^n b^{n-i} \beta_i)
\]

It can be seen from equation (2) that

\[
E(x_n) = E(y_n) = 0
\]

\[
E(x_n^2) = \sigma_x^2
\]

\[
E(y_n^2) = \sigma_y^2
\]

\[
E(x_n x_m) = \sigma_x^2 |m-n|
\]

\[
E(y_n y_m) = \sigma_y^2 |m-n|
\]

We therefore have a process in which:

- each aim point is normally distributed with mean zero and standard deviation \( \sigma_x, \sigma_y \).

- the correlation coefficient between the \( n^{th} \) and \( m^{th} \) round is \( \frac{|n-m|}{\sigma_x^2} \) in the \( x \) coordinate and \( \frac{|n-m|}{\sigma_y^2} \) in the \( y \) coordinate.

- when \( a \) and \( b \) equal zero or one, the process degenerates into the independent or salvo cases, respectively.
IV. METHOD OF SOLUTION

The inputs for the program consist of the following parameters:

<table>
<thead>
<tr>
<th>Address</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a</td>
<td>Correlation constant in the range coordinate (0 \leq a \leq 1)</td>
</tr>
<tr>
<td>2</td>
<td>b</td>
<td>Correlation constant in the deflection coordinate, (0 \leq b \leq 1)</td>
</tr>
<tr>
<td>3</td>
<td>(\sigma_R)</td>
<td>Standard deviation of aim error in the range coordinate (mils)</td>
</tr>
<tr>
<td>4</td>
<td>(\sigma_D)</td>
<td>Standard deviation of aim error in the deflection coordinate (mils)</td>
</tr>
<tr>
<td>5</td>
<td>(\beta_R)</td>
<td>Standard deviation of ballistic dispersion in the range coordinate (mils)</td>
</tr>
<tr>
<td>6</td>
<td>(\beta_D)</td>
<td>Standard deviation of ballistic dispersion in the deflection coordinate (mils)</td>
</tr>
<tr>
<td>7</td>
<td>S</td>
<td>Slant range at commencement of firing run (ft.)</td>
</tr>
<tr>
<td>8</td>
<td>R</td>
<td>Firing rate (rounds per minute)</td>
</tr>
<tr>
<td>9</td>
<td>c</td>
<td>Aircraft speed (knots)</td>
</tr>
<tr>
<td>10</td>
<td>N</td>
<td>Maximum number of rounds per pass</td>
</tr>
<tr>
<td>11</td>
<td>P</td>
<td>Conditional kill probability of a round</td>
</tr>
<tr>
<td>12</td>
<td>L</td>
<td>Target length (normal to flight path in range coordinate) (ft.)</td>
</tr>
<tr>
<td>13</td>
<td>W</td>
<td>Target width (normal to flight path in deflection coordinate) (ft.)</td>
</tr>
<tr>
<td>14</td>
<td>F</td>
<td>Number of Monte Carlo iterations</td>
</tr>
<tr>
<td>15</td>
<td>(N_r)</td>
<td>Number of dummy passes through random number generators</td>
</tr>
<tr>
<td>16</td>
<td>(\Delta N)</td>
<td>Increment in the number of rounds at which target kill probabilities are determined.</td>
</tr>
</tbody>
</table>

*The parameter addresses are explained in appendix C, Data Subroutine.

Unless otherwise specified, the following parameters are assigned the designated values:

<table>
<thead>
<tr>
<th>Address</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>b</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>c</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>P</td>
<td>1000</td>
</tr>
<tr>
<td>14</td>
<td>F</td>
<td>1000</td>
</tr>
<tr>
<td>16</td>
<td>(\Delta N)</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>(N_r)</td>
<td>0</td>
</tr>
</tbody>
</table>
The program flow chart and FORTRAN statements are shown in appendixes A and B. The subroutines are described in appendixes C, D and E. The solution is obtained in the following manner:

1. Before starting the first iteration for the first data set, \( n \) dummy passes are made through the random number generator.

2. For each Monte Carlo iteration:
   a. Given the standard deviations of aim error \((\sigma_R, \sigma_D)\) and the aim point of the \((n-1)\)st round, two Gaussian-distributed \((\text{mean }= 0, \text{ standard deviation }= 1)\) random numbers \((a, \theta)\) are selected, and the aim point of the \(n\)th round \((R_n, D_n)\) is determined:

   \[
   R_n = aR_{n-1} + \sigma_R (1-a^2)^{1/2} a
   \]
   \[
   D_n = bD_{n-1} + \sigma_D (1-b^2)^{1/2} \theta
   \]

   For the first round \((n = 1)\):

   \[
   R_1 = \sigma_R \alpha
   \]
   \[
   D_1 = \sigma_D \beta
   \]

   b. Given the target dimensions \((L, W)\), the firing rate \((R)\), the aircraft speed \((c)\) and the initial slant range \((S)\), the half-target size \((\text{in mils})\) is determined for the time at which the \(n\)th round is fired:

   \[
   \frac{1}{2} L_n = \frac{500L}{S - (n-1)c(60/R)(1.688)}
   \]
   \[
   \frac{1}{2} W_n = \frac{500W}{S - (n-1)c(60/R)(1.688)}
   \]

   c. The previously computed aim point is checked to determine if it is within three standard deviations of ballistic dispersion \((\beta_R, \beta_D)\) from the target:

   \[
   |R_n| \leq \frac{1}{2} L_n + 3\beta_R
   \]
   \[
   |D_n| \leq \frac{1}{2} W_n + 3\beta_D
   \]
If the aim point is not within three standard deviations of ballistic
dispersion from the target in either coordinate, it is assumed that
the round misses the target and the aim point for the \((n + 1)^{\text{st}}\) round
is computed (part a.).

d. If the aim point is within three standard deviations from the target
in both coordinates, a Gaussian-distributed random number \(\delta\) is
selected, and the impact point of the round in the range coordinate is
determined and checked to ascertain whether the round falls within
the limits of the target in the range coordinates.

\[
|R_n + \beta_R \delta| < 1/2 L_n \quad (9)
\]

If the round does not fall within the target limits, the aim point for
the \((n + 1)^{\text{st}}\) round is computed (part a.).

e. If the \(n\)th round does fall within the target limits in the range co-
ordinate, another Gaussian-distributed random number \(\varepsilon\) is
selected and the impact point of the round in the deflection co-
ordinate is determined. Another check is then made to determine
if the round hits the target:

\[
|D_n + \beta_D \varepsilon| < 1/2 W_n \quad (10)
\]

If the round does not hit the target, the aim point for the \((n + 1)^{\text{st}}\)
round is determined (part a.).

f. If the \(n\)th round does hit the target, a uniformly distributed (between
0 and 1) random number (PP) is selected and compared with the
conditional kill probability to determine if the hit results in a target
kill. If \(P < PP\), the weapon does not destroy the target and the aim
point of the \((n + 1)^{\text{st}}\) weapon is computed (part a.).

g. If \(P \geq PP\), the \(n\)th weapon does kill the target. The count \(N_h\) of the
Monte Carlo iterations for which the target is destroyed is increased
by one. In addition, a counter \(Jj_i\) for the next highest multiple \(i\)
of the increment (\(\Delta N\)) in the number of rounds for which the probability
is to be determined is increased by one:

\[
N_h + 1 \rightarrow N_h \\
Jj_i + 1 \rightarrow Jj_i \quad (11)
\]

When a round has resulted in a kill or when the maximum number of
rounds \(N\) has been fired without killing the target, the entire process
is repeated until \(F\) Monte Carlo iterations have been completed.
3. At the conclusion, the estimated probability of destroying the target with \( N \) rounds is determined:

\[
P(N) = \frac{N_h}{F}.
\]

The probability of destroying the target with \( j \) rounds is determined:

\[
P(j) = \sum_{i=1}^{n} \frac{J_j}{F}
\]

\( j = n \Delta N \quad (n = 1, 2, \ldots N/\Delta N) \) (14)

V. USER'S INSTRUCTIONS

Input flexibility has been attained by allowing the user to vary any or all of the parameters in a computer run. There is no programmed limit to the number of data sets which a user may submit in a run. The only restriction is that each data set must terminate with one blank card, and the last set in the run must terminate with two blank cards.

For each data set after the first, the user need submit only those parameter values in a set that are different from those in the previous set. This is accomplished by identifying each input parameter by its address.

VI SAMPLE PROBLEM

A target whose dimensions in the plane normal to the flight path are 10 feet x 10 feet is to be attacked by a 500 KTAS aircraft with a firing rate of 1000 rounds per minute. The aircraft commences firing at a slant range of 5000 feet and fires a single burst of 100 rounds at the target. The probability of kill given a hit is 0.2. The standard deviations of aiming error are 6 and 4 mils in range and deflection, respectively, and the standard deviations of ballistic dispersion are 2 and 1 mils in range and deflection. The correlation constants in both range and deflection are assumed to be 0.8. What is the probability of killing the target with the entire burst and with smaller bursts in increments of 10 rounds? A Monte Carlo simulation of 2000 iterations is selected. The input statements are as follows:

<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.8</td>
</tr>
<tr>
<td>2</td>
<td>0.8</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>5000</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
</tr>
</tbody>
</table>
(Continued from preceding page.)

<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>500</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>11</td>
<td>0.2</td>
</tr>
<tr>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>13</td>
<td>10</td>
</tr>
<tr>
<td>14</td>
<td>2000</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>10</td>
</tr>
</tbody>
</table>

The submittal form and output for this sample problem are shown in appendix F.

VII. KEYPUNCH INSTRUCTIONS

Two methods of punching the input parameters can be used. In the first method, one card is used for each parameter. The address of the parameter is punched first, followed by a space and then the parameter value is punched. In the second method, the packed card method, several parameters and their addresses can be punched on a single card (as many as space permits). In each case, the address is followed by its parameter value which is in turn followed by the next address and value. Each address and parameter value must be separated by a space and must also be separated from the value preceding or the address following it.

VIII. OPERATOR'S INSTRUCTIONS

Run under control of the Bell System on the IBM 7090. No special instructions are required.

IX. TIMING

This program requires approximately \( 6NF \) milliseconds running time per data set, where \( N \) is the number of rounds per pass and \( F \) is the number of Monte Carlo iterations.
INTERIM RESEARCH MEMORANDUM
IRM-43

APPENDIX B

FORTRAN STATEMENTS

```
DIMENSION D(16), JJ(200), FF(200)
EQUIVALENCE (D(1), A), (D(2), B), (D(3), SIG1), (D(4), SIGD), (D(5), BETAR)
1, (D(6), BETA), (D(7), S), (D(8), R), (D(9), C), (D(10), FN), (D(11), P),
2, (D(12), FL), (D(13), W), (D(14), F), (D(16), DN)
A=0.0
B=0.0
F=1000.0
P=1.0
DN=1.0
C=0.0
10 CALL DATA (D, IND)
   IF (IND) 40, 40, 20
20 PRINT 30
30 FORMAT (1H1)
   CALL ENOJOB
40 PRINT 30
   II=D(15)
   DO 45 I=1, II
      CALL RANUMB (DUMMY)
45 CALL GRNUMB (DUMMY)
   FLR2=500.0*FL/S
   WR2=500.0*W/S
   V=1.688*C
   VR2=60.0*V/(S*R)
   N=FN/DN
   DO 50 I=1, N
50 JJ(I)=0
   NH=0
   JJJ=FN
   II=F
   DO 170 I=1, II
      CALL GRNUMB (ALPHA)
      CALL GRNUMB (GAMMA)
      RC=SIG1*ALPHA
      DC=SIGD*GAMMA
      DO 150 J=1, JJJ
150 IF (VR) 80,70,80
    60 IF (VR) 80,70,80
70 WN2=WR2
   FLN2=FLR2
   GO TO 90
   FLN2=FLR2/(1.0-FLOAT(J-1)*VR)
   WN2=WR2/(1.0-FLOAT(J-1)*VR)
90 IF (ABS(FRC-FLN2-3.0*BETAR) 100, 140, 140
100 IF (ABS(FD-WN2-3.0*BETAD) 110, 140, 140
110 CALL GRNUMB (DELTA)
   IF (ABS(FRC+DELTA*BETAR)-FLN2) 120, 140, 140
120 CALL GRNUMB (EPS)
   IF (ABS(FD+EPS*BETAD)-WN2) 130, 140, 140
130 CALL RANUMB (PP)
   JAB=J
   IF (PP-P) 160, 140, 140
```

B-1
INTERIM RESEARCH MEMORANDUM
IRM-43

140 CALL GRNUMB(ALPHA)
   CALL GRNUMB(GAMMA)
   CALL GRSIG=SQRTF(1.0-A**2)*ALPHA
   DO 150 DC=B+DC+SIGD=SQRTF(1.0-B**2)*GAMMA
   GO TO 170
160 NH=NH+1
   K=FLOAT(JAB)/DN+0.9999
   JJ(K)=JJ(K)+1
170 CONTINUE
   FN=FLOAT(NH)/F
   DO 180 I=2,N
      JJ(I)=JJ(I)+JJ(I-1)
180   FF(I)=FLOAT(JJ(I))/F
      FF(I)=FLOAT(JJ(I))/F
      PRINT 190, A, SIGR, BETAR, B, SIGD, BETAD
190 FORMAT (1H1, 49X, 11HCORRELATION, 6X, 9HMAIN ERROR, 4X, 9HBALLISTIC/
   13X, 10HCOORDINATE, 7X, 9HCOUNTS, 8X, 6H(MILS), 4X, 17HDISPERSI
   20M (MILS)/34X, 5HRANGE, 19X, F5.3, 10X, F5.1, 9X, FS.1/34X, 10HDE
   3FLECTION, 8X, F5.3, 10X, F5.1, 9X, FS.1)
      PRINT 200, C
200 FORMAT (/34X, 25H AIRCRAFT SPEED (KTAS) = F6.0)
      PRINT 205, R
205 FORMAT (36X, 25H FIRING RATE (RDS/MIN) = F6.0)
      PRINT 210, S
210 FORMAT (34X, 43H SLANT RANGE AT COMMENCEMENT OF FIRING RUN
   14H(FEET) = F7.0)
      KN=FN
      PRINT 215, KN
215 FORMAT (34X, 25H NO. OF ROUNDS PER PASS = S16)
      PRINT 220, P
220 FORMAT (34X, 31H CONDITIONAL KILL PROBABILITY =F6.3)
      PRINT 225, FL
225 FORMAT (/34X, 23H TARGET LENGTH (FEET) = F7.1)
      PRINT 230, W
230 FORMAT (34X, 23H TARGET WIDTH (FEET) = F7.1)
      KF=FF
      PRINT 235, KF
235 FORMAT (/34X, 35H NUMBER OF MONTE CARLO ITERATIONS = S16)
      NEMP=0(15)
      PRINT 240, NEMP
240 FORMAT (34X, 43H NUMBER OF EMPTY PASSES THROUGH RANDOM NO.
   112GENERATORS = S17)
      PRINT 245, FH
245 FORMAT (/34X, 29H PROBABILITY OF TARGET KILL = F6.3)
      PRINT 250
250 FORMAT (/34X, 5H KILL/34X, 13HNO. OF ROUNDS, 9X, 11HPROBABILITY)
      DO 260 I=1,N
   IDN=DN
   IR=+1.IDN
260 PRINT 270, IR, FF(I)
270 FORMAT (39X, 15, 15X, F5.3)
      GO TO 10
      END
1. **Introduction:**

Many computer programs require the flexibility of varying any or all of the parameters in a computer run. Although FORTRAN is fairly flexible in its arithmetic and control statements, its input-output statements are quite rigid. In order to read cards for instance, considerable effort must be expended by the FORTRAN programmer in writing his input statements. This subroutine eliminates some of that tedium. The concept of a "data set" is used. A data set consists of a sequence of punched cards terminated by one blank card. A parameter deck for a computer run may consist of several data sets. Such a parameter deck is terminated by two blank cards.

2. **Parameter Addresses:**

The primary advantage of this subroutine over FORTRAN input statements results from the use of "parameter addresses." An address is a relative location in the computer memory. It is the subscript of an array or matrix. For example, in an array called $X$, the parameter value $X_{53}$ would be located at address 53. By using the parameter addresses, a user of the program need submit only those parameter values in a data set that are different from those in the previous set.

Three types of addresses are permitted by this subroutine.

1. *A numeric address consisting of one to five characters, each of which is a digit 0 - 9.* Such an address $(n)$ refers to the $n$th element in a specified array.

2. *An alpha address consisting of one to six characters,* the first of which must be alphabetic (A-Z). The remaining may be alphabetic or numeric (A-Z or 0-9). Such an address refers to the $n$th element in a specified array $(1 \leq n \leq 26)$, where the first character of the address corresponds to $n$ as the 26 letters of the alphabet correspond to the integers 1-26.

3. *A matrix address consisting of two or more numeric fields separated by commas.* For example, the address 53, 47 refers to the element in the 53rd row and the 47th column of a two-dimensional matrix. There is no limit to the number of dimensions in a matrix address.

3. **Input Card Format:**

A standard submittal form (see attachment) has been designed for the analyst. This form provides for entering parameter values with their associated addresses. The user indicates blank cards to separate data sets. The keypunch operator has the option of punching one address and value per card, or, if the addresses are sequential, of punching one address and several values on a card.
Only columns 1-72 of a card are used. Each column must contain one of the following: a digit (0-9), a "+" or "-" sign or a dash, a letter (A-Z), a period, a comma, or a blank. Each punched card must contain one parameter address. The address may start in column 1, or, if desired, may start in a later column, provided all columns before it are blank. The address is terminated by at least one blank column. Only one address is permitted on the card. Succeeding columns contain one or more parameter values, each separated by one or more blank columns. A value may be signed or unsigned. The length of the value field is variable. No blanks are permitted within a value field. A value may be punched with or without an exponent. An exponent is recognized by the presence of a plus or minus sign (or dash) between the fractional part and exponent part of the value. Decimal points (periods) may be punched in either the fractional or exponent parts of a value. If more than one value is punched on a card, those after the first will be entered at sequential addresses relative to the address of the first value.

4 Usage:

A data set is read by the use of the statement:

CALL DATA (X, I)

in a FORTRAN program for the IBM 7090. The argument X is the name of an array in the program. The argument I is an indicator set by the subroutine. This indicator may be tested by the main program upon return from the subroutine. It will have a value of 0 or 1 or 2:

0: The subroutine has read a data set. The main program will normally proceed to operate on this data.

1: The subroutine has read the second blank card which terminates the parameter deck. The main program will normally terminate at this point.

2: The subroutine has read a "bad" data card. The main program may terminate the run, or ignore the card and return to the subroutine to read the rest of the data set.

If the cards to be read contain matrix addresses, additional arguments must be included in the FORTRAN calling statement:

CALL DATA (X, D1, D2, D3, ..., Dn, I)

where D1 is the ith dimension of the matrix X.

5 Method:

See the attached flow chart. DATA reads parameter values and loading addresses from cards. If sense switch 5 is up, it will read the values and addresses from tape (unit A2). It converts the values to floating point numbers, and stores them as elements of an array specified in the calling statement. The elements are specified by the addresses. If a card (or tape record) is read which contains non-permitted characters (see input card format above), DATA prints the statement "bad data card," followed by an image of the card itself.
6. Coding Information:

See the symbolic listing included in this appendix. DATA is written in the 7090 FAP language. It must be used in conjunction with the BELL system. It requires 401 words storage space.
INTERIM RESEARCH MEMORANDUM
IRM-43

SYMBOLIC LISTING

FAP
ENTRY DATA
DATA SXA X1+1
SXA X2+2
SXA X4+4
CAL 1+6
ADD CORE
STO XLOC
AXT 1+1
SXA *+1,1
CAL **3,4
ANA MASK
TNZ **2
TXI *+4,1,1
SXA EXIT*1,1
TX* **1,1,*1
SXA **1,1
CLA **4,4
STA A1
STA F1A
STA 112
AXT 1+1
RETURN INDICATOR = 1
A1 SXD **,1
A TSX HHR=4 READ A CARD
PZE CARD
TPA EXIT
TRA BAD
STZ ADDRES ADDRESS = 0
STZ VALUE VALUE = 0
STZ EXP EXP = 0
STZ POINT POINT = 0
AXT 1+1 FIELD = 1
AXT 13+1
A2 TNX HH+1,1 COLUMN GT 72
AXT 42+2
SXA COLUMN2
B LXA COLUMN2 COLUMN = COLUMN+1
TXN A2+2,6
SXA COLUMN2
LDQ CARD+12,1
ROL 36+2
PXD 0+0
LGL 6
STO CHARAC
ORA FLOAT
FAD FLOAT
STO NUMB
AXT 42+4
CLA CHARAC
CAS TABLE +42,4
TRA **2
TRA *+3
TIX *-3,4,1

C-7
INTERIM RESEARCH MEMORANDUM
IRM-43

TRA PAD
LXA FIELD+2
TRA F1+1,2
TRA F7
TRA F6
TRA F5
TRA F4
TRA F3
TRA F2
F1 TXH B++41
F1A STZ
AXT 2+2
SXA FIELD+2
TXH H++31
TXH I++28
TXL DAD++2
AXT 3+2
SXA FIELD+2
TXI *+1+++2
SXA ADDRESS++
TRA B
F2 TXH J++41
TXH H++31
TXH K++28
TXH BAD++2
TXH L++1
AXT 4+2
SXA FIELD+2
AXT 2+2
SXA DIMENS+2
CLA ADDRESS
TSE BAD
TMI BAD
F2A STZ ADDER
TRA B
F3 TXH J++41
TXH B++31
TXH K++28
TXH B++2
AXT L++1
TRA BAD
F4 TXH M++41
TXH N++31
TXH P++28
TXH BAD++2
TXH Q++41
TRA T
F5 TXH B++41
AXT 6+2
SXA FIELD+2
TXH U++31
TXH M++28
TXH BAD++2
TXH G++1

C-8
INTERIM RESEARCH MEMORANDUM
IRM-43

F6
TRA BAD
TXH X+4+1
TXH U+4*31
TXH Y+4*28
TXH BAD+4*2
TXH G+4+1
TRA BAD
BLANK CHARACTER
NUMERIC CHARACTER
SIGN OR DASH
PERIOD

F7
TXH Z+4+41
TXH BR+4*31
TXH EE+4*28
TXH BAD+4*2
TXL CAD+4+1
BLANK CHARACTER
NUMERIC CHARACTER
SIGN OR DASH

G
AXT 1:2
SXA POINT+2
TRA B
PERIOD+ POINT = 1

H
LDQ ADDRES
MPY M10
XCA
ACL CHARAC
STO ADDRES
TRA B
ADDRESS = 10 X ADDRESS + N

I
TXH B+4*30
CLA ADDRES
SSM
STO ADDRES
TRA B
+ SIGN
SET SIGN OR ADDRESS

J
AXT 5:2
SXA FIELD+2
TRA B
FIELD = 5

K
TXH L+4*30
CLA VALUE
SSM
STO VALUE
TRA L1
+ SIGN
SET SIGN OF VALUE

L
AXT 1:2
SXA POINT+2
TRA B
POINT = 1

L1
AXT 6:2
SXA FIELD+2
TRA B
FIELD = 6

M
AXT 5:2
SXA FIELD+2
TRA S
FIELD = 5

N
LDQ ADDER
MPY M10
STO ADDER
TXS T1+6
MPY CHARAC
XCA
ADD ADDER
STO ADDER
TRA B
ADDER = 10 X ADDER + N X PROD

P
TXH R+4*30
CLA VALUE
SSM
+ SIGN
SET SIGN OF VALUE

C-9
STO VALUE
TRA R
Q AXT 1+2
SXA POINT 2
R AXT 6+2
SXA FIELD 2
S LXA EXIT 2
TXI #+1,2,-3
PXA 0.2
SUB DIMENS
TNE BAD
T TNE T1+a
CLA ADDER
SUB PROD
STO ADDER
TZE BAD
TMI BAD
ADD ADDRES
STO ADDRES
CLA DIMENS
ADD H1
STO DIMENS
TRA F2A
T1 SXA T4+a
CLA H1
STO PROD
STA T3
LXA DIMENS 2
TXI #+1,2+1
LXA X4+a
T2 CAL T3
ADD H1
STA T3
T3 CLA 0+a
STA #+1
LDQ #
RQL 10
MPY PROD
STQ PROD
TXI T2,2+1
T4 AXT #+a
TRA 1+a
U CLA POINT
TNZ V
LDQ VALUE
FMP DEC10
SSP
FAD NUMB
LDQ VALUE
LLS 0
STO VALUE
TRA B
V LXA POINT 4
CLA NUMB
VALUE = VALUE + N/110*POINT

C-10
FDP DEC10
XCA
TIX =-2*4+1
LDQ VALUE
LLS 0
FAD VALUE
STO VALUE
TRA DD

W
TXH B*4+30 + SIGN
CLA VALUE
SET SIGN OF VALUE
SSM
STO VALUE
TRA B

X
CLA XLOC
SUB ADRES
STA *2
CLA VALUE
STO *2
TRA AA

Y
AXT 7*2
SXA FLC:LD*2 FIELD = 2
TXH GG*4*30 + SIGN
CLA EXP
SET SIGN OF EXP
SSM
STO EXP
TRA GG

Z
CLA XLOC
SUB ADRES
STA Z1
CLA DEC10
LDQ EXP
CALL EXP13
XCA
FHP VALUE
Z1
STO *2
VALUE = 0
AA
STZ VALUE
TRA FF
BB
CLA POINT
TMZ CC
LDQ EXP
EXP = 10 X EXP + N
FMP DEC10
SSP
FAD NUMB
LDQ EXP
LLS 0
STO EXP
TRA B

CC
LXA POINT*4
CLA NUMB
EXP = EXP + N/(10**POINT)
FDP DEC10
XCA
TIX =-2*4+1
LDQ EXP
LLS 0

C-11
FAW EXP
STO EXP

DD CLA POINT
ADD H1
STO POINT
TRA B

EE CLA XLOC
SUB ADDRESS
STA EE1
CLA DEC10
LDQ EXP
CALL EXP13
XCA
FMP VALUE

EE1 STO **
PXD 0,0
TXH **2+4+30
SSM VALUE

STO VALUE

FF STZ EXP
AXT 5+2
SXA FIELD,2
CAL ADDRESS
ADD H1
SLW ADDRESS

GG STZ POINT
TRA B

HH LXA FIELD,1
TXL JJ,1,1
TXL BAD,1,4
TXL A,1,5
TXH 1,1,6
CLA XLOC
SUB ADDRESS
STA **2
CLA VALUE
STO **
TRA A

II TXH BAD,1,7
CLA XLOC
SUB ADDRESS
STA III
CLA DEC10
LDQ EXP
CALL EXP13
XCA
FMP VALUE

III1 STO **
TRA A

BAD TXS HPRINT,4
PIE PRINT,0,19
AXT 2+1

112 SXD **,1
X1 AXT **,1
**INTERIM RESEARCH MEMORANDUM**

**IRM-43**

<table>
<thead>
<tr>
<th>X2</th>
<th>AXT <strong>#12</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>X4</td>
<td>AXT <strong>#14</strong></td>
</tr>
<tr>
<td>EXIT</td>
<td>TRA <strong>#16</strong></td>
</tr>
<tr>
<td>MASK</td>
<td>OCT 777777700000</td>
</tr>
<tr>
<td>PRINT</td>
<td>BCD $ BAD DATA CARD...</td>
</tr>
<tr>
<td>CARD</td>
<td>BSS 12</td>
</tr>
<tr>
<td>ADDRESS</td>
<td>HTR <strong>#</strong></td>
</tr>
<tr>
<td>VALUE</td>
<td>HTR <strong>#</strong></td>
</tr>
<tr>
<td>EXP</td>
<td>HTR <strong>#</strong></td>
</tr>
<tr>
<td>POINT</td>
<td>HTR <strong>#</strong></td>
</tr>
<tr>
<td>FIELD</td>
<td>HTR <strong>#</strong></td>
</tr>
<tr>
<td>COLUMN</td>
<td>HTR <strong>#</strong></td>
</tr>
<tr>
<td>TABLE</td>
<td>OCT 60</td>
</tr>
<tr>
<td></td>
<td>BLANK</td>
</tr>
<tr>
<td>OCT 0</td>
<td>0</td>
</tr>
<tr>
<td>OCT 1</td>
<td>1</td>
</tr>
<tr>
<td>OCT 2</td>
<td>2</td>
</tr>
<tr>
<td>OCT 3</td>
<td>3</td>
</tr>
<tr>
<td>OCT 4</td>
<td>4</td>
</tr>
<tr>
<td>OCT 5</td>
<td>5</td>
</tr>
<tr>
<td>OCT 6</td>
<td>6</td>
</tr>
<tr>
<td>OCT 7</td>
<td>7</td>
</tr>
<tr>
<td>OCT 10</td>
<td>8</td>
</tr>
<tr>
<td>OCT 11</td>
<td>9</td>
</tr>
<tr>
<td>OCT 20</td>
<td># SIGN</td>
</tr>
<tr>
<td>OCT 40</td>
<td>- SIGN</td>
</tr>
<tr>
<td>OCT 14</td>
<td>DASH</td>
</tr>
<tr>
<td>OCT 71</td>
<td>Z</td>
</tr>
<tr>
<td>OCT 70</td>
<td>Y</td>
</tr>
<tr>
<td>OCT 67</td>
<td>X</td>
</tr>
<tr>
<td>OCT 66</td>
<td>W</td>
</tr>
<tr>
<td>OCT 65</td>
<td>V</td>
</tr>
<tr>
<td>OCT 64</td>
<td>U</td>
</tr>
<tr>
<td>OCT 63</td>
<td>T</td>
</tr>
<tr>
<td>OCT 62</td>
<td>S</td>
</tr>
<tr>
<td>OCT 51</td>
<td>R</td>
</tr>
<tr>
<td>OCT 50</td>
<td>Q</td>
</tr>
<tr>
<td>OCT 47</td>
<td>P</td>
</tr>
<tr>
<td>OCT 46</td>
<td>O</td>
</tr>
<tr>
<td>OCT 45</td>
<td>N</td>
</tr>
<tr>
<td>OCT 44</td>
<td>M</td>
</tr>
<tr>
<td>OCT 43</td>
<td>L</td>
</tr>
<tr>
<td>OCT 42</td>
<td>K</td>
</tr>
<tr>
<td>OCT 41</td>
<td>J</td>
</tr>
<tr>
<td>OCT 31</td>
<td>I</td>
</tr>
<tr>
<td>OCT 30</td>
<td>H</td>
</tr>
<tr>
<td>OCT 27</td>
<td>G</td>
</tr>
<tr>
<td>OCT 26</td>
<td>F</td>
</tr>
<tr>
<td>OCT 25</td>
<td>E</td>
</tr>
<tr>
<td>OCT 24</td>
<td>D</td>
</tr>
<tr>
<td>OCT 23</td>
<td>C</td>
</tr>
<tr>
<td>OCT 22</td>
<td>B</td>
</tr>
<tr>
<td>OCT 21</td>
<td>A</td>
</tr>
<tr>
<td>OCT 39</td>
<td>PERIOD</td>
</tr>
<tr>
<td>OCT 38</td>
<td>COMMA</td>
</tr>
</tbody>
</table>

C-13
CHARAC HTR **
DIMENS HTR **
ADDER HTR **
H10 HTR 10
DEC10 DEC 10.0
H1 HTR 1
PROD HTR **
AMASK OCT 77777
FLOAT OCT 233000000000
NUMB HTR **
XLOC HTR **
CORE OCT 100001
JJ SYN X1
END
OEG COMPUTER DATA SUBMITTAL FORM

Submitted by: ____________________ Date: ____________________
Program No. _______ Est. Time _______ Classification _______

Special Instructions: _______________________________________

<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
<th>Address</th>
<th>Value</th>
<th>Address</th>
<th>Value</th>
<th>Address</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NOTES:
1. A value of zero must be entered as 0, not left blank.
2. Decimal pts. may be omitted if understood to follow the rightmost digit.
3. The value $3 \times 10^{-5}$ may be entered as .00003 or 3-5, not as $3 \times 10^{-5}$.
4. The factor portion of a value may not contain more than 8 digits.
5. The exponent portion of a value must lie within the range ±39.
6. Exponents may be omitted if zero. If not, they must be signed.
7. Blank cards should be indicated by: __________.

C-15
(REVERSE BLANK)
1. **Purpose:**

GRNUMB provides a floating point pseudo-random number $X$. The distribution of successive values of $X$ are Gaussian with a mean of zero and a standard deviation of one.

2. **Method:**

Consider the set of uniformly distributed pseudo-random numbers $Y_i$. GRNUMB generates a sequence of $Y_i$ by the method of congruences:

$$Y_i = 2^{-35}(5^{15}Y_{i-1}, \mod 2^{35})$$

over the range $0 < Y_i < 1$. The variance of this uniform set is

$$\sigma_Y^2 = \int_0^1 (Y-1/2)^2 \, dY = 1/12 .$$

If $X$ is the mean of any selection of $m$ of the uniform numbers $Y$, the Central Limit Theorem states that the variable $X$ approaches a normal distribution where $m$ is sufficiently large. A satisfactory value for $m$ is 30. Values of $X$ are generated as a sequence of $X_n$, where $n$ denotes the $n$th entry to GRNUMB.

$$X_n = \sqrt{1/m\sigma_Y^2} \sum_{i=1}^{m} (Y_i - 1/2) = \sqrt{4} \sum_{i=1}^{30} (Y_i - 1/2)$$

where $Y_0 = X_{n-1}$, and $X_0 = 2^{-35}$. The variance of this normal set is 1.

3. **Usage:**

$X$ is obtained by use of the statement:

```
CALL GRNUMB (X)
```

in a FORTRAN program for the IBM 7090.

4. **Coding Information:**

See the symbolic listing on the following page. GRNUMB is written in the 7090 FAP language. It requires 40 words storage space and 900 microseconds operating time.
SYMBOLIC LISTING

FAP
REM GRNUMB G. WESTLUND 18 JUNE 1962 (7090)
REM GAUSSIAN DISTRIBUTED RANDOM NUMBER GENERATOR.
REM ENTER VIA FORTRAN STATEMENT CALL GRNUMB(X)
REM SEQUENCE STARTS AT DEC 1, YIELDS X WITH STAND. DEV. =1.
ENTRY GRNUMB

GRNUMB SXA XX1, 1
CLA 1, 4
STA F
AXT 30, 1
STZ NUM
C
LDQ Numb
MPY MULT
STQ Numb
CLA Numb
SUB CHAR
ARS 4
ADD NUM
STO NUM
TIX C, 1, 1
LDQ NUM
MPY MAGIC
LRS 27
TZE D
LRS 8
CLA H125
AND H8
LLS 8
ALS 19
TRA E
D
CLA H125
ALS 27
E
STO NUM
CLA H125
LLS 27
FAD NUM
F
STO**
XX1
AXT**, 1
TRA 2, 4
NUM
HTR**
NUMB
DEC 1
MULT
DEC 3#517578125
CHAR
TIX 0, 0, 0
MAGIC
DEC 0.31622780B0
HE
DEC 8
H125
DEC 125
END

D-2
APPENDIX E

RANUMB SUBROUTINE

1. **Purpose:**
   RANUMB provides a floating point pseudo-random number \( X \). Successive values of \( X \) are uniformly distributed over the range \( 0 \leq X < 1 \).

2. **Method:**
   The generated value of \( X \) is a member of a set of pseudo-random numbers. This set is a sequence of \( X_n \), where \( n \) denotes the \( n \)th entry to RANUMB. The set is generated by the method of congruences:
   \[
   N_n = (5^{15}N_{n-1}) \mod 2^{35}
   \]
   \[
   X_n = N_n / 2^{35}
   \]
   where \( N_0 = 1 \).

3. **Usage:**
   \( X \) is obtained by use of the statement:
   
   ```fortran
   CALL RANUMB (X)
   ```

   in a FORTRAN program for the IBM 7090.

4. **Coding Information:**
   See the symbolic listing on the following page. RANUMB is written in the 7090 FAP language. It requires 15 words storage space and 41 microseconds operating time.
FAP
REM RANUMB G. WESTLUND 18 JUNE 1962 (7090)
REM UNIFORMLY DISTRIBUTED RANDOM NUMBER GENERATOR.
REM ENTER VIA FORTRAN STATEMENT CALL RANUMB (X)
REM SEQUENCE STARTS AT DEC 1, YIELDS (0 LE X LT 1).
ENTRY RANUMB

RANUMB
CLA 1, 4
STA C
LDQ NUMB
MPY MULT
STQ NUMB
FPD 0, 0
RQL 9
LGL 27
ACL CHAR
FAD CHAR

C
STO **
TRA 2, 4

NUMB
DEC 1

MULT
DEC 30517578125

CHAR
TIX 0, 0, 0
END
## Sample Problem Submittal Form

**OEG Computer Data Submittal Form**

**Submitted by:** John Doe  **Date:** 26 July 1963  
**Program No.**  
**Est. Time:** 4 min  **Classification:** Unclassified  
**Special Instructions:**

<table>
<thead>
<tr>
<th>Address</th>
<th>Value</th>
<th>Address</th>
<th>Value</th>
<th>Address</th>
<th>Value</th>
<th>Address</th>
<th>Value</th>
<th>Address</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.8</td>
<td>2</td>
<td>0.8</td>
<td>3</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>7</td>
<td>5000</td>
<td>8</td>
<td>1000</td>
<td>9</td>
<td>500</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>11</td>
<td>0.2</td>
<td>12</td>
<td>10</td>
<td>13</td>
<td>10</td>
<td>14</td>
<td>2000</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>10</td>
<td>b</td>
<td></td>
<td>b</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NOTES:**

1. A value of zero must be entered as 0, not left blank.
2. Decimal pts. may be omitted if understood to follow the rightmost digit.
3. The value $3 \times 10^{-3}$ may be entered as .00003 or 3-3, not as $3 \times 10^{-3}$.
4. The factor portion of a value may not contain more than 8 digits.
5. The exponent portion of a value must lie within the range $\pm 39$.
6. Exponents may be omitted if zero. If not, they must be signed.
7. Bank cards should be indicated by: ____________b

(Reverse Blank)
APPENDIX G

SAMPLE PROBLEM OUTPUT

<table>
<thead>
<tr>
<th>COORDINATE</th>
<th>CORRELATION CONSTANTS</th>
<th>AIM ERROR (MILS)</th>
<th>BALLISTIC DISPERSION (MILS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RANGE</td>
<td>0.800</td>
<td>6.0</td>
<td>2.0</td>
</tr>
<tr>
<td>DEFLECTION</td>
<td>0.800</td>
<td>4.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

AIRCRAFT SPEED (KTAS) = 500.
FIRING RATE (RODS/MIN, = 1000.
SLANT RANGE AT COMMENCEMENT OF FIRING RUN (FT) = 5000.
NO. OF ROUNDS PER PASS = 100
CONDITIONAL KILL PROBABILITY = 0.200

TARGET LENGTH (FEET) = 10.0
TARGET WIDTH (FEET) = 10.0

NUMBER OF MONTE CARLO ITERATIONS = 2000
NUMBER OF EMPTY PASSES THROUGH RANDOM NO. GENERATORS = 0

PROBABILITY OF TARGET KILL = 0.993

<table>
<thead>
<tr>
<th>NO. OF ROUNDS</th>
<th>KILL PROBABILITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.048</td>
</tr>
<tr>
<td>20</td>
<td>0.114</td>
</tr>
<tr>
<td>30</td>
<td>0.200</td>
</tr>
<tr>
<td>40</td>
<td>0.287</td>
</tr>
<tr>
<td>50</td>
<td>0.384</td>
</tr>
<tr>
<td>60</td>
<td>0.499</td>
</tr>
<tr>
<td>70</td>
<td>0.658</td>
</tr>
<tr>
<td>80</td>
<td>0.812</td>
</tr>
<tr>
<td>90</td>
<td>0.950</td>
</tr>
<tr>
<td>100</td>
<td>0.993</td>
</tr>
</tbody>
</table>

G-1
(REVERSE BLANK)