KfK 3107
März 1981

Evidence of Significant Bias in an Elementary Random Number Generator

H. Borgwaldt, V. Brandl
Institut für Neutronenphysik und Reaktortechnik
Projekt Nukleare Sicherheit

Kernforschungszentrum Karlsruhe

KERNFORSCHUNGSZENTRUM KARLSRUHE

Institut für Neutronenphysik und Reaktortechnik

Projekt Nukleare Sicherheit

KfK 3107

Evidence of Significant Bias in an Elementary Random Number Generator

H. Borgwaldt, V. Brandl

Kernforschungszentrum Karlsruhe GmbH, Karlsruhe

Kernforschungszentrum Karlsruhe GmbH
ISSN 0303-4003

Abstract

An elementary pseudo random number generator for isotropically distributed unit vectors in 3-dimensional space has been tested for bias. This generator uses the IBM-supplied routine RANDU and a transparent rejection technique. The tests show clearly that non-randomness in the pseudo random numbers generated by the primary IBM generator leads to bias in the order of 1 percent in estimates obtained from the secondary random number generator. FORTRAN listings of 4 variants of the random number generator called by a simple test programme and output listings are included for direct reference.

Nachweis erheblicher systematischer Fehler (Bias) bei einem elementaren Zufallszahlen-Generator

Zusammenfassung

Es wurde die Erwartungstreue eines elementaren Zufallszahlen-Generators überprüft. Dieser Generator, mit dem normierte, im 3-dimensionalen Raum isotrop verteilte Richtungsvektoren erzeugt werden sollen, benutzt die IBM-Routine RANDU und ein übersichtliches Verwerfungsverfahren. Die Testrechnungen zeigen eindeutig, daß Abweichungen von reiner Zufälligkeit in der Folge von Pseudo-Zufallszahlen aus dem primären IBM-Generator RANDU beim abgeleiteten Zufallszahlen-Generator für bestimmte Schätzungen zu systematischen Fehlern (Bias) in der Größenordnung von 1 Prozent führen. Teil dieses Berichts sind FORTRAN-Listen für 4 Varianten des Zufallszahlen-Generators, der von einem einfachen Testprogramm aufgerufen wird, sowie die Ausgabeprotokolle der Testserien.

Evidence of Significant Bias in an Elementary Random Number Generator.

H. Borgwaldt, V. Brandl

Kernforschungszentrum Karlsruhe,
Postfach 3640,
D-7500 Karlsruhe 1

An elementary pseudo random number generator for isotropically distributed unit vectors in 3-dimensional space has been tested for bias. This generator uses the IBM-supplied routine RANDU and a transparent rejection technique. The tests show clearly that non-randomness in the pseudo random numbers generated by the primary IBM generator leads to bias in the order of 1 percent in estimates obtained from the secondary random number generator. FORTRAN listings of 4 variants of the random number generator called by a simple test programme and output listings are included for direct reference.

1. Introduction.

Monte Carlo techniques have been popular in reactor neutron physics calculations in situations where diffusion theory is not accurate enough and the geometry too complicated for transport codes. Meanwhile the performance of 2-dimensional neutronic transport codes has been improved considerably, reducing the field of Monte Carlo applications.

But, in reactor technology neutron transport is not the only area of Monte Carlo application. The response surface method, used in reactor safety research, takes an approximate response surface equation as a fast-running substitute for the accurate response of a complex safety code to input parameter vectors. Conventional Monte Carlo techniques are, then, used to sample repeatedly input vectors from an assumed probability distribution, evaluate the approximate responses, and obtain finally an estimate of the probability distribution of the response in form of a histogram (or a set of moments) /1/. The results thus obtained depend on several factors: the goodness of fit of the response function approximation in the region of concern, the sample size, the use of special sampling techniques (e.g. Latin hypercube sampling), and finally on the properties of the random number generator (RNG).

Another typical Monte Carlo application, also in reactor safety research , is fault tree evaluation by simulation, in cases where analytical methods are not available. Here too one must rely on a reasonable behaviour of the RNGs used. Therefore, it seems appropriate to communicate, as a general warning, adverse experience originating from a neutron transport application.

RNGs for sampling from arbitrary distributions can be realized by several means, e.g. transformation, rejection and special techniques. The common feature of all these techniques is that they use an input stream of values from a primary RNG, usually supplied with the computer software. This RNG yields uniformly distributed values in the open interval (0., 1.). They must be sufficiently random for all practical applications. If this cannot be assured then all derived results may be questioned.

It has been recognized long ago that RNGs can be demonstrated to be far from perfect /2/. On the other hand, the authors have, as many other practitioners, believed that, at least for the established RNGs, their imperfections can be demonstrated only by sophisticated mathematical methods, based on the theory of numbers or similar tools. Therefore, we expected that straight-forward applications should not show any effects comparable to the inevitable statistical errors, known to decrease with the square root of the sample size. This conviction got lost, when one of us (V. B.) investigated neutron transport in an anisotropic medium using a modified version of the Monte Carlo neutron transport code KAMCCO /3/.

In addition to the expected anisotropy of the z-direction versus the transversal directions the results showed also a marked anisotropy in the (x,y)-plane not explainable by any feature of the physical model. After some search, in which coding errors, especially in the ASSEMBLER versions of RNGs and truncation effects were suspected, we recognized that a secondary, derived RNG used for generating isotropically distributed unit vectors in 3-dimensional space was very sensitive to the inherent weakness of the IBM-supplied primary RNG RANDU /4/.

2. Specification of the RNG tested.

In 3-dimensional space the marginal distribution for each component of isotropically distributed vectors (normalized to unit length) is uniform in the interval (-1.0, 1.0). The projection of such vectors into any 2-dimensional plane has an isotropic distribution of directions in this plane. This leads to a simple recipe for the pertinent RNG of pseudo random vectors (X,Y,Z) :

Step 1: Sample one component, e.g. Z, from the uniform distribution in the interval (-1.0, 1.0), using RANDU.

Step 2: Sample similarly the remaining components X, Y.

Step 3: If the point (X,Y) is inside a circular disk of unit radius, then continue; else return to step 2 (This rejection technique has an efficiency of 78.5 percent).

Step 4: Normalize the projection (X,Y) such that the complete vector (X,Y,Z) gets unit normalization.

With an ideal primary RNG this secondary RNG should perform very well. For one 3-component vector an average of only 3.55 calls of the primary RNG and one call of the SQRT function are needed.

3. Test procedure, including variants of the RNG.

The marginal distributions of the absolute values of each vector component are uniform on the interval (0., 1.) with a mean of 0.50 . This was taken as a criterion for the test programmes reproduced in the Appendix. The results given are deviations in percents for the estimated mean absolute value of all 3 vector components. In addition, these errors have been converted to standard deviations to show their significance. Although the IBM-supplied RNG RANDU yields single precision values only, we have employed double precision throughout the test programmes to eliminate any possible truncation effect. For each test case an adequate sample size of 100,000 realisations and a sequence of 10 runs was chosen to obtain significant results.

Case A of our test programme is the reference case, coded as explained above. The cases B, C, and D each contain one modification versus the reference case. Case A (cf. Table 1) shows over 10 runs an average bias for the x-component of .49 percent, and a bias of -.62 percent for the z-component. These values are quite high and look significant, corresponding to 2.7 and -3.4 (single run) standard deviations, respectively. Throughout the series of 10 runs there is no change of sign in the errors for these 2 components. As to the estimates for the y-component, the registered average deviation of .15 percent corresponding to .79 standard deviations is significantly smaller. The sign of the error is positive in 9 out of 10 runs, indicating bias also for this component. But here a more careful analysis would be necessary to exclude pure coincidence.

DEMONSTRATION OF BIASSED R.N.G.   CASE A

SAMPLE SIZE: 100000,   RUNS: 10

RUN	BIAS (PCT.) FOR X	Y	Z	BIAS (ST. DEV.) FOR X	Y	Z
1	0.548	-0.139	-0.480	3.00	-0.76	-2.63
2	0.665	0.100	-0.839	3.64	0.55	-4.61
3	0.328	0.218	-0.440	1.79	1.19	-2.42
4	0.616	0.153	-0.587	3.38	0.84	-3.23
5	0.501	0.098	-0.604	2.74	0.54	-3.31
6	0.319	0.380	-0.600	1.75	2.08	-3.29
7	0.352	0.011	-0.448	1.93	0.06	-2.46
8	0.808	0.022	-0.809	4.42	0.12	-4.45
9	0.349	0.291	-0.706	1.90	1.59	-3.87
10	0.413	0.314	-0.667	2.26	1.72	-3.66

Table 1. Results for reference case A.

DEMONSTRATION OF BIASSED R.N.G.   CASE B

SAMPLE SIZE: 100000,   RUNS: 10

RUN	BIAS (PCT.) FOR X	Y	Z	BIAS (ST. DEV.) FOR X	Y	Z
1	0.638	-0.087	-0.483	3.50	-0.48	-2.65
2	0.838	-0.003	-0.838	4.58	-0.02	-4.60
3	0.519	0.183	-0.440	2.85	1.00	-2.42
4	0.707	0.144	-0.588	3.88	0.79	-3.23
5	0.526	0.212	-0.604	2.88	1.16	-3.32
6	0.477	0.345	-0.599	2.62	1.89	-3.29
7	0.490	-0.051	-0.449	2.68	-0.28	-2.46
8	0.906	0.015	-0.808	4.96	0.08	-4.44
9	0.432	0.306	-0.706	2.36	1.68	-3.87
10	0.619	0.220	-0.668	3.38	1.21	-3.67

Table 2. Results of test series B.

For Case B the order, in which vector components are determined, has been changed. The z-component is selected after the (x,y)-direction has been determined. Note, that under these circumstances the random numbers used to determine X, Y, Z are always in sequence, whereas in the reference Case A the rejection technique for X, Y sometimes breaks up this triplet into one isolated random number (for Z) and a doublet (for X, Y), with an even number of rejected random numbers in between. The results (cf. Table 2) seem to indicate that the behaviour of the modified RNG becomes worse for the x-component. In terms of standard deviations the average errors of the x-, y-, and z-components become 3.4, .70, and -3.4, respectively. For reasons unexplained, the y-component shows the best behaviour of all 4 cases, considering not only the magnitude of errors but also the higher number of sign changes.

Following a suggestion by E. Gelbard /5/, we have next attempted to decouple somewhat the selection of random numbers used for generating the z-component and the (x,y)-pair, respectively. For Case C this is done by inserting one blind call to the primary RNG RANDU after determining the z-component. Table 3 shows no qualitative changes, in comparison with the reference Case A. The mean deviation in the x-component, .50 percent or 2.8 standard deviations, and the corresponding value for the z-component, -.61 percent or -3.4 standard deviations, stay practically unchanged. Note that through the loop the pairs of random numbers used for the (x,y)-combination of one vector and the random numbers used for the z-component of the following vector still form triplet sequences.

Only the last Case D shows a significant improvement. For this case a blind call of the primary RNG RANDU has been inserted before determining the z-component of the random vectors. Now (cf. Table 4) the mean errors for the x- and z-components are reduced to .19 percent or 1.1 standard deviations and .03 percent or .14 standard deviations, respectively. The corresponding value of -.19 percent (or -1.1 standard deviations) for the y-component seems to indicate that the bias has been partially shifted to this component. But to corroborate this evidence a much more detailed analysis would be necessary.

DEMONSTRATION OF BIASSED R.N.G.   CASE C

SAMPLE SIZE: 100000,   RUNS: 10

RUN	BIAS (PCT.) FOR X	Y	Z	BIAS (ST. DEV.) FOR X	Y	Z
1	0.418	0.212	-0.574	2.29	1.16	-3.15
2	0.251	0.183	-0.352	1.38	1.00	-1.93
3	0.534	0.037	-0.409	2.93	0.20	-2.25
4	0.302	0.416	-0.517	1.66	2.28	-2.85
5	0.540	0.419	-0.892	2.95	2.29	-4.90
6	0.418	0.132	-0.455	2.29	0.72	-2.50
7	0.613	0.029	-0.664	3.36	0.16	-3.64
8	0.701	0.048	-0.670	3.83	0.26	-3.69
9	0.553	0.240	-0.881	3.03	1.31	-4.83
10	0.704	-0.110	-0.681	3.85	-0.60	-3.73

Table 3. Results of test series C.

DEMONSTRATION OF BIASSED R.N.G.   CASE D

SAMPLE SIZE: 100000, RUNS: 10

RUN	BIAS (PCT.) FOR X	Y	Z	BIAS (ST. DEV.) FOR X	Y	Z
1	0.139	-0.328	0.200	0.76	-1.80	1.10
2	0.072	-0.058	-0.006	0.40	-0.32	-0.04
3	0.177	-0.346	0.166	0.97	-1.90	0.91
4	0.019	-0.047	0.132	0.11	-0.26	0.73
5	0.068	0.135	-0.202	0.37	0.74	-1.11
6	0.322	-0.005	-0.293	1.77	-0.03	-1.61
7	0.328	-0.378	0.052	1.79	-2.07	0.29
8	0.152	-0.328	0.195	0.84	-1.79	1.07
9	0.221	-0.225	-0.016	1.21	-1.24	-0.09
10	0.438	-0.342	0.031	2.40	-1.87	0.17

Table 4. Results of test series D.

4. Conclusions.

The sample calculations done for this communication demonstrate very clearly that non-randomness in the pseudo random numbers generated by a standard primary RNG like IBM's RANDU can easily lead to bias in the order of 1 percent in estimates from secondary RNGs. This is much more than can be tolerated. We also see, from the Case D data, which direction to take in order to overcome such an effect, at least for this special application. But we are left with a very uneasy feeling, what tricks any RNG may play in situations which are less transparent.

What we really need, is not a RNG which has passed certain statistical tests for randomness; it may fail in the very next one. Instead we would need a RNG, which could be proved to show approximate randomness by some practical standard. By now, sufficient mathematical tools should be available , e.g. in the theory of numbers, the theory of programme complexity, and in tools usually employed to develop cryptographic algorithms.

We want to close with one short remark. It seems to have been widely accepted that primary RNGs should be extremely fast-running. RANDU, like many standard generators, is of the congruential type. Starting with an arbitrary odd integer N(0) for initialisation, a sequence of pseudo random odd integers N(i) is generated by the recursive relation

N(i+1) = A * N(i) modulo (2**31),   for i=0,1,2,...

with A = 65539 = 2**16 + 3.

These pseudo random integers are then normalized. Such procedures are extremely simple, which also means that their programme complexity is very low. Therefore, we may suspect that the generated sequence is far from random /6/. Yet, the use of such extremely simple, fast-running procedures seems to be completely unnecessary. For most realistic Monte Carlo applications a break-down of the computer times shows that only a very minor part is used in calls of the primary RNG. This means that introducing more complex primary RNGs will in most cases not affect adversely the performance of Monte Carlo programmes.

5. References:

/1/ W. Sengpiel, Report KfK 2965 (1980)

/2/ R.R. Coveyou, R.D. MacPherson, Journ. of the ACM, Vol. 14, pp. 100-119 (1967)

/3/ G. Arnecke, H. Borgwaldt, V. Brandl, M. Lalovic, Report KFK 2190 (1976)

/4/ "System/360 Scientific Subroutine Package", Fifth Edition, IBM Corporation (August 1970)

/5/ E. Gelbard, private communication (1980)

/6/ C.P. Schnorr, "Zufälligkeit und Wahrscheinlichkeit", Lecture Notes in Mathematics 218, Springer-Verlag, Berlin-Heidelberg-New York (1971)

6. Appendix: Listings of FORTRAN test programmes.

```
C     TEST-PROGRAMME FOR CHECKING BIAS IN THE ABSOLUTE VALUES
C     OF THE COMPONENTS OF VECTORS INTENDED TO BE UNIFORMLY
C     DISTRIBUTED ON THE 3-DIMENSIONAL UNIT SPHERE.
C
C     REFERENCE CASE A.
C
      REAL*8 UNIT/1./,TWO/2./,HALF/.5/,X,Y,Z,SUMX,SUMY,SUMZ,VARX,VARY,
     +       VARZ,DEVX,DEVY,DEVZ,RAND,TERM
      IA = 1
      RAND = 0.
C     RANDU-ROUTINE INITIALIZED
      ITOT = 100000
      NRUN = 10
      WRITE (6,1000) ITOT,NRUN
      ICONT = 1
  100 SUMX = 0.
      SUMY = 0.
      SUMZ = 0.
      VARX = 0.
      VARY = 0.
      VARZ = 0.
      DO 300 I=1,ITOT
        CALL RANDU (IA,IB,RAND)
        IA = IB
        Z = TWO*RAND-UNIT
C     Z UNIFORM IN (-1.,1.)
  200   CALL RANDU (IA,IB,RAND)
        IA = IB
        X = TWO*RAND-UNIT
        CALL RANDU (IA,IB,RAND)
        IA = IB
        Y = TWO*RAND-UNIT
        TERM = X**2+Y**2
        IF (TERM.GT.UNIT) GOTO 200
C     X,Y UNIFORM IN UNIT DISK
        TERM = DSQRT((UNIT-Z**2)/TERM)
        X = X*TERM
        Y = Y*TERM
C     X,Y NORMALIZED
        SUMX = SUMX+DABS(X)
        VARX = VARX+X**2
        SUMY = SUMY+DABS(Y)
        VARY = VARY+Y**2
        SUMZ = SUMZ+DABS(Z)
  300   VARZ = VARZ+Z**2
      TERM = DFLOAT(ITOT)
      SUMX = SUMX/TERM
      SUMY = SUMY/TERM
      SUMZ = SUMZ/TERM
C     MEAN ABSOLUTE VALUES OF VECTOR COMPONENTS
```

```
      VARX = DSQRT((VARX/TERM-SUMX**2)/TERM)
      VARY = DSQRT((VARY/TERM-SUMY**2)/TERM)
      VARZ = DSQRT((VARZ/TERM-SUMZ**2)/TERM)
C     STANDARD DEVIATIONS OF ESTIMATES
      SUMX = (SUMX-HALF)/HALF
      SUMY = (SUMY-HALF)/HALF
      SUMZ = (SUMZ-HALF)/HALF
C     RELATIVE DEVIATIONS OF ESTIMATES
      DEVX = (SUMX*HALF)/VARX
      DEVY = (SUMY*HALF)/VARY
      DEVZ = (SUMZ*HALF)/VARZ
C     NORMALIZED DEVIATIONS OF ESTIMATES
      WRITE (6,2000) ICONT,SUMX,SUMY,SUMZ,DEVX,DEVY,DEVZ
      ICONT = ICONT+1
      IF (ICONT.LE.NRUN) GOTO 100
      STOP
 1000 FORMAT ('0'/'0'/'0'/'0'/'0'20X,'DEMONSTRATION OF BIASSED R.N.G. '
     +       ,'      CASE A'/'0',20X,'SAMPLE SIZE: ',I8,',     RUNS: ',
     +       I4/1X/'0',26X,'BIAS (PCT.) FOR',17X,'BIAS (ST. DEV.) FOR'
     +       /'0',13X,'RUN',7X,'X',9X,'Y',9X,'Z',13X,'X',9X,'Y',9X,
     +       'Z'/1X)
 2000 FORMAT ('0',I15,1X,2P3F10.3,3X,0P3F10.2)
      END
```

******

```
C     TEST-PROGRAMME FOR CHECKING BIAS IN THE ABSOLUTE VALUES
C     OF THE COMPONENTS OF VECTORS INTENDED TO BE UNIFORMLY
C     DISTRIBUTED ON THE 3-DIMENSIONAL UNIT SPHERE.
C
C     CASE B, Z-COMPONENT AFTER X, Y.
C
      REAL*8 UNIT/1./,TWO/2./,HALF/.5/,X,Y,Z,SUMX,SUMY,SUMZ,VARX,VARY,
     +       VARZ,DEVX,DEVY,DEVZ,RAND,TERM
      IA = 1
      RAND = 0.
C     RANDU-ROUTINE INITIALIZED
      ITOT = 100000
      NRUN = 10
      WRITE (6,1000) ITOT,NRUN
      ICONT = 1
  100 SUMX = 0.
      SUMY = 0.
      SUMZ = 0.
      VARX = 0.
      VARY = 0.
      VARZ = 0.
      DO 300 I=1,ITOT
  200   CALL RANDU (IA,IB,RAND)
        IA = IB
        X = TWO*RAND-UNIT
        CALL RANDU (IA,IB,RAND)
        IA = IB
        Y = TWO*RAND-UNIT
```

```
          TERM = X**2+Y**2
          IF (TERM.GT.UNIT) GOTO 200
C       X,Y UNIFORM IN UNIT DISK
          CALL RANDU (IA,IB,RAND)
          IA = IB
          Z = TWO*RAND-UNIT
C       Z UNIFORM IN (-1.,1.)
          TERM = DSQRT((UNIT-Z**2)/TERM)
          X = X*TERM
          Y = Y*TERM
C       X,Y NORMALIZED
          SUMX = SUMX+DABS(X)
          VARX = VARX+X**2
          SUMY = SUMY+DABS(Y)
          VARY = VARY+Y**2
          SUMZ = SUMZ+DABS(Z)
  300     VARZ = VARZ+Z**2
        TERM = DFLOAT(ITOT)
        SUMX = SUMX/TERM
        SUMY = SUMY/TERM
        SUMZ = SUMZ/TERM
C       MEAN ABSOLUTE VALUES OF VECTOR COMPONENTS
        VARX = DSQRT((VARX/TERM-SUMX**2)/TERM)
        VARY = DSQRT((VARY/TERM-SUMY**2)/TERM)
        VARZ = DSQRT((VARZ/TERM-SUMZ**2)/TERM)
C       STANDARD DEVIATIONS OF ESTIMATES
        SUMX = (SUMX-HALF)/HALF
        SUMY = (SUMY-HALF)/HALF
        SUMZ = (SUMZ-HALF)/HALF
C       RELATIVE DEVIATIONS OF ESTIMATES
        DEVX = (SUMX*HALF)/VARX
        DEVY = (SUMY*HALF)/VARY
        DEVZ = (SUMZ*HALF)/VARZ
C       NORMALIZED DEVIATIONS OF ESTIMATES
        WRITE (6,2000) ICONT,SUMX,SUMY,SUMZ,DEVX,DEVY,DEVZ
        ICONT = ICONT+1
        IF (ICONT.LE.NRUN) GOTO 100
        STOP
 1000 FORMAT ('0'/'0'/'0'/'0'/'0'20X,'DEMONSTRATION OF BIASSED R.N.G. '
     +         ,'     CASE B'/'0',20X,'SAMPLE SIZE: ',I8,',      RUNS: ',
     +         I4/1X/'0',26X,'BIAS (PCT.) FOR',17X,'BIAS (ST. DEV.) FOR'
     +         /'0',13X,'RUN',7X,'X',9X,'Y',9X,'Z',13X,'X',9X,'Y',9X,
     +         'Z'/1X)
 2000 FORMAT ('0',I15,1X,2P3F10.3,3X,0P3F10.2)
        END
```

```
C     TEST-PROGRAMME FOR CHECKING BIAS IN THE ABSOLUTE VALUES
C     OF THE COMPONENTS OF VECTORS INTENDED TO BE UNIFORMLY
C     DISTRIBUTED ON THE 3-DIMENSIONAL UNIT SPHERE.
C
C     CASE C, SELECTION OF (X,Y) MADE MORE INDEPENDENT.
C
      REAL*8 UNIT/1./,TWO/2./,HALF/.5/,X,Y,Z,SUMX,SUMY,SUMZ,VARX,VARY,
     +       VARZ,DEVX,DEVY,DEVZ,RAND,TERM
      IA = 1
      RAND = 0.
C     RANDU-ROUTINE INITIALIZED
      ITOT = 100000
      NRUN = 10
      WRITE (6,1000) ITOT,NRUN
      ICONT = 1
  100 SUMX = 0.
      SUMY = 0.
      SUMZ = 0.
      VARX = 0.
      VARY = 0.
      VARZ = 0.
      DO 300 I=1,ITOT
        CALL RANDU (IA,IB,RAND)
        IA = IB
        Z = TWO*RAND-UNIT
C     Z UNIFORM IN (-1.,1.)
        CALL RANDU (IA,IB,RAND)
        IA = IB
C     DECOUPLING OF (X,Y)-SELECTION
  200   CALL RANDU (IA,IB,RAND)
        IA = IB
        X = TWO*RAND-UNIT
        CALL RANDU (IA,IB,RAND)
        IA = IB
        Y = TWO*RAND-UNIT
        TERM = X**2+Y**2
        IF (TERM.GT.UNIT) GOTO 200
C     X,Y UNIFORM IN UNIT DISK
        TERM = DSQRT((UNIT-Z**2)/TERM)
        X = X*TERM
        Y = Y*TERM
C     X,Y NORMALIZED
        SUMX = SUMX+DABS(X)
        VARX = VARX+X**2
        SUMY = SUMY+DABS(Y)
        VARY = VARY+Y**2
        SUMZ = SUMZ+DABS(Z)
  300   VARZ = VARZ+Z**2
      TERM = DFLOAT(ITOT)
      SUMX = SUMX/TERM
      SUMY = SUMY/TERM
      SUMZ = SUMZ/TERM
C     MEAN ABSOLUTE VALUES OF VECTOR COMPONENTS
```

```
      VARX = DSQRT((VARX/TERM-SUMX**2)/TERM)
      VARY = DSQRT((VARY/TERM-SUMY**2)/TERM)
      VARZ = DSQRT((VARZ/TERM-SUMZ**2)/TERM)
C     STANDARD DEVIATIONS OF ESTIMATES
      SUMX = (SUMX-HALF)/HALF
      SUMY = (SUMY-HALF)/HALF
      SUMZ = (SUMZ-HALF)/HALF
C     RELATIVE DEVIATIONS OF ESTIMATES
      DEVX = (SUMX*HALF)/VARX
      DEVY = (SUMY*HALF)/VARY
      DEVZ = (SUMZ*HALF)/VARZ
C     NORMALIZED DEVIATIONS OF ESTIMATES
      WRITE (6,2000) ICONT,SUMX,SUMY,SUMZ,DEVX,DEVY,DEVZ
      ICONT = ICONT+1
      IF (ICONT.LE.NRUN) GOTO 100
      STOP
 1000 FORMAT ('0'/'0'/'0'/'0'/'0'20X,'DEMONSTRATION OF BIASSED R.N.G. '
     +        ,'     CASE C'/'0',20X,'SAMPLE SIZE: ',I8,',      RUNS: ',
     +        I4/1X/'0',26X,'BIAS (PCT.) FOR',17X,'BIAS (ST. DEV.) FOR'
     +        /'0',13X,'RUN',7X,'X',9X,'Y',9X,'Z',13X,'X',9X,'Y',9X,
     +        'Z'/1X)
 2000 FORMAT ('0',I15,1X,2P3F10.3,3X,0P3F10.2)
      END
```

******

```
C     TEST-PROGRAMME FOR CHECKING BIAS IN THE ABSOLUTE VALUES
C     OF THE COMPONENTS OF VECTORS INTENDED TO BE UNIFORMLY
C     DISTRIBUTED ON THE 3-DIMENSIONAL UNIT SPHERE.
C
C     CASE D, SELECTION OF Z MADE MORE INDEPENDENT.
C
      REAL*8 UNIT/1./,TWO/2./,HALF/.5/,X,Y,Z,SUMX,SUMY,SUMZ,VARX,VARY,
     +       VARZ,DEVX,DEVY,DEVZ,RAND,TERM
      IA = 1
      RAND = 0.
C     RANDU-ROUTINE INITIALIZED
      ITOT = 100000
      NRUN = 10
      WRITE (6,1000) ITOT,NRUN
      ICONT = 1
  100 SUMX = 0.
      SUMY = 0.
      SUMZ = 0.
      VARX = 0.
      VARY = 0.
      VARZ = 0.
      DO 300 I=1,ITOT
        CALL RANDU (IA,IB,RAND)
        IA = IB
C     DECOUPLING OF Z-SELECTION
        CALL RANDU (IA,IB,RAND)
        IA = IB
        Z = TWO*RAND-UNIT
C     Z UNIFORM IN (-1.,1.)
```

```
  200   CALL RANDU (IA,IB,RAND)
        IA = IB
        X = TWO*RAND-UNIT
        CALL RANDU (IA,IB,RAND)
        IA = IB
        Y = TWO*RAND-UNIT
        TERM = X**2+Y**2
        IF (TERM.GT.UNIT) GOTO 200
C     X,Y UNIFORM IN UNIT DISK
        TERM = DSQRT((UNIT-Z**2)/TERM)
        X = X*TERM
        Y = Y*TERM
C     X,Y NORMALIZED
        SUMX = SUMX+DABS(X)
        VARX = VARX+X**2
        SUMY = SUMY+DABS(Y)
        VARY = VARY+Y**2
        SUMZ = SUMZ+DABS(Z)
  300   VARZ = VARZ+Z**2
      TERM = DFLOAT(ITOT)
      SUMX = SUMX/TERM
      SUMY = SUMY/TERM
      SUMZ = SUMZ/TERM
C     MEAN ABSOLUTE VALUES OF VECTOR COMPONENTS
      VARX = DSQRT((VARX/TERM-SUMX**2)/TERM)
      VARY = DSQRT((VARY/TERM-SUMY**2)/TERM)
      VARZ = DSQRT((VARZ/TERM-SUMZ**2)/TERM)
C     STANDARD DEVIATIONS OF ESTIMATES
      SUMX = (SUMX-HALF)/HALF
      SUMY = (SUMY-HALF)/HALF
      SUMZ = (SUMZ-HALF)/HALF
C     RELATIVE DEVIATIONS OF ESTIMATES
      DEVX = (SUMX*HALF)/VARX
      DEVY = (SUMY*HALF)/VARY
      DEVZ = (SUMZ*HALF)/VARZ
C     NORMALIZED DEVIATIONS OF ESTIMATES
      WRITE (6,2000) ICONT,SUMX,SUMY,SUMZ,DEVX,DEVY,DEVZ
      ICONT = ICONT+1
      IF (ICONT.LE.NRUN) GOTO 100
      STOP
 1000 FORMAT ('0'/'0'/'0'/'0'/'0'20X,'DEMONSTRATION OF BIASSED R.N.G. '
     +       ,'      CASE D'/'0',20X,'SAMPLE SIZE: ',I8,',     RUNS: ',
     +       I4/1X/'0',26X,'BIAS (PCT.) FOR',17X,'BIAS (ST. DEV.) FOR'
     +       /'0',13X,'RUN',7X,'X',9X,'Y',9X,'Z',13X,'X',9X,'Y',9X,
     +       'Z'/1X)
 2000 FORMAT ('0',I15,1X,2P3F10.3,3X,0P3F10.2)
      END
```