

3.3.4. The Spectral Test

In this section we shall study an especially important way to check the quality of linear congruential random number generators; not only do all good generators pass this test, all generators now known to be bad actually *fail* it. Thus it is by far the most powerful test known, and it deserves particular attention. Our discussion will also bring out some fundamental limitations on the degree of randomness we can expect from linear congruential sequences and their generalizations.

The spectral test embodies aspects of both the empirical and theoretical tests studied in previous sections: it is like the theoretical tests because it deals with properties of the full period of the sequence, and it is like the empirical tests because it requires a computer program to determine the results.

A. Ideas underlying the test. The most important randomness criteria seem to rely on properties of the joint distribution of t consecutive elements of the sequence, and the spectral test deals directly with this distribution. If we have a sequence $\langle U_n \rangle$ of period m , the idea is to analyze the set of all m points

$$\{(U_n, U_{n+1}, \dots, U_{n+t-1})\} \quad (1)$$

in t -dimensional space.

For simplicity we shall assume that we have a linear congruential sequence (X_0, a, c, m) of maximum period length m (so that $c \neq 0$), or that m is prime and $c = 0$ and the period length is $m - 1$. In the latter case we shall add the point $(0, 0, \dots, 0)$ to the set (1), so that there are always m points in all; this extra point has a negligible effect when m is large, and it makes the theory much simpler. Under these assumptions, (1) can be rewritten as

$$\left\{ \frac{1}{m} (x, s(x), s(s(x)), \dots, s^{t-1}(x)) \mid 0 \leq x < m \right\}, \quad (2)$$

where

$$s(x) = (ax + c) \bmod m \quad (3)$$

is the "successor" of x . Note that we are considering only the set of all such points in t dimensions, not the order in which those points are actually generated. But the order of generation is reflected in the dependence between components of the vectors; and the spectral test studies such dependence for various dimensions t by dealing with the totality of all points (2).

For example, Fig. 8 shows a typical small case in 2 and 3 dimensions, for the generator with

$$s(x) = (137x + 187) \bmod 256. \quad (4)$$

Of course a generator with period length 256 will hardly be random, but 256 is small enough that we can draw the diagram and gain some understanding before we turn to the larger m 's that are of practical interest.

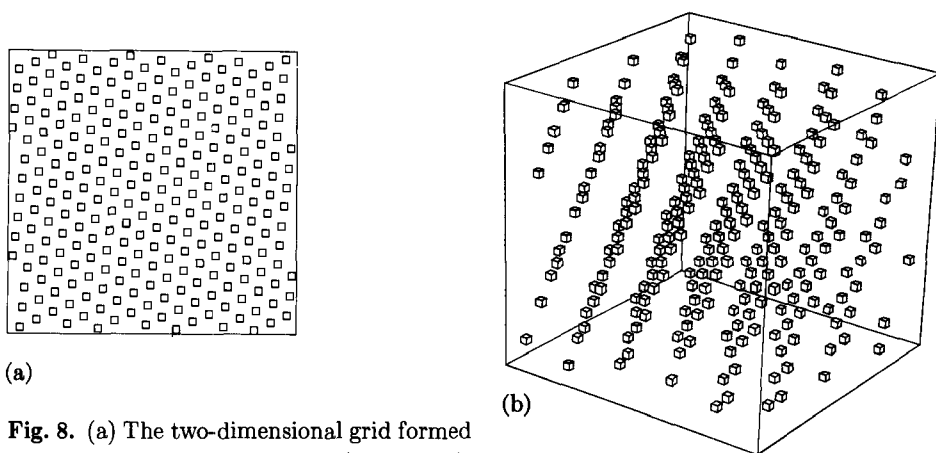


Fig. 8. (a) The two-dimensional grid formed by all pairs of successive points (X_n, X_{n+1}) , when $X_{n+1} = (137X_n + 187) \bmod 256$.
 (b) The three-dimensional grid of triplets (X_n, X_{n+1}, X_{n+2}) . [Illustrations courtesy of Bruce G. Baumgart.]

Perhaps the most striking thing about the pattern of boxes in Fig. 8 is that we can cover them all by a fairly small number of parallel lines; indeed, there are many different families of parallel lines that will hit all the points. For example, a set of 20 nearly vertical lines will do the job, as will a set of 21 lines that tilt upward at roughly a 30° angle. We commonly observe similar patterns when driving past farmlands that have been planted in a systematic manner.

If the same generator is considered in three dimensions, we obtain 256 points in a cube, obtained by appending a "height" component $s(x)$ to each of the 256 points $(x, s(x))$ in the plane of Fig. 8(a), as shown in Fig. 8(b). Let's imagine that this 3-D crystal structure has been made into a physical model, a cube that we can turn in our hands; as we rotate it, we will notice various families of parallel planes that encompass all of the points. In the words of Wallace Givens, the random numbers stay "mainly in the planes."

At first glance we might think that such systematic behavior is so nonrandom as to make congruential generators quite worthless; but more careful reflection, remembering that m is quite large in practice, provides a better insight. The regular structure in Fig. 8 is essentially the "grain" we see when examining our random numbers under a high-power microscope. If we take truly random numbers between 0 and 1, and round or truncate them to finite accuracy so that each is an integer multiple of $1/\nu$ for some given number ν , then the t -dimensional points (1) we obtain will have an extremely regular character when viewed through a microscope.

Let $1/\nu_2$ be the maximum distance between lines, taken over all families of parallel straight lines that cover the points $\{(x/m, s(x)/m)\}$ in two dimensions. We shall call ν_2 the two-dimensional accuracy of the random number generator, since the pairs of successive numbers have a fine structure that is

essentially good to one part in ν_2 . Similarly, let $1/\nu_3$ be the maximum distance between planes, taken over all families of parallel planes that cover all points $\{(x/m, s(x)/m, s(s(x))/m)\}$; we shall call ν_3 the accuracy in three dimensions. The t -dimensional accuracy ν_t is the reciprocal of the maximum distance between hyperplanes, taken over all families of parallel $(t-1)$ -dimensional hyperplanes that cover all points $\{(x/m, s(x)/m, \dots, s^{t-1}(x)/m)\}$.

The essential difference between periodic sequences and truly random sequences that have been truncated to multiples of $1/\nu$ is that the "accuracy" of truly random sequences is the same in all dimensions, while the "accuracy" of periodic sequences decreases as t increases. Indeed, since there are only m points in the t -dimensional cube when m is the period length, we can't achieve a t -dimensional accuracy of more than about $m^{1/t}$.

When the independence of t consecutive values is considered, computer-generated random numbers will behave essentially as if we took truly random numbers and truncated them to $\lg \nu_t$ bits, where ν_t decreases with increasing t . In practice, such varying accuracy is usually all we need. We don't insist that the 10-dimensional accuracy be 2^{35} , in the sense that all $(2^{35})^{10}$ possible 10-tuples $(U_n, U_{n+1}, \dots, U_{n+9})$ should be equally likely on a 35-bit machine; for such large values of t we want only a few of the leading bits of $(U_n, U_{n+1}, \dots, U_{n+t-1})$ to behave as if they were independently random.

On the other hand when an application demands high resolution of the random number sequence, simple linear congruential sequences will necessarily be inadequate; a generator with larger period should be used instead, even though only a small fraction of the period will actually be generated. Squaring the period will essentially square the accuracy in higher dimensions, i.e., it will double the effective number of bits of precision.

The spectral test is based on the values of ν_t for small t , say $2 \leq t \leq 6$. Dimensions 2, 3, and 4 seem to be adequate to detect important deficiencies in a sequence, but since we are considering the entire period it seems best to be somewhat cautious and go up into another dimension or two; on the other hand the values of ν_t for $t \geq 10$ seem to be of no practical significance whatever. (This is fortunate, because it appears to be rather difficult to calculate ν_t when $t \geq 10$.)

Note that there is a vague relation between the spectral test and the serial test; for example, a special case of the serial test, taken over the entire period as in exercise 3.3.3-19, counts the number of boxes in each of 64 subsquares of Fig. 8(a). The main difference is that the spectral test rotates the dots so as to discover the least favorable orientation. We shall return to a consideration of the serial test later in this section.

It may appear at first that we should apply the spectral test only for one suitably high value of t ; if a generator passes the test in three dimensions, it seems plausible that it should also pass the 2-D test, hence we might as well omit the latter. The fallacy in this reasoning occurs because we apply more stringent conditions in lower dimensions. A similar situation occurs with the serial test: Consider a generator that (quite properly) has almost the same number of points

in each subcube of the unit cube, when the unit cube has been divided into 64 subcubes of size $\frac{1}{4} \times \frac{1}{4} \times \frac{1}{4}$; this same generator might yield completely *empty* subsquares of the unit square, when the unit square has been divided into 64 subsquares of size $\frac{1}{8} \times \frac{1}{8}$. Since we increase our expectations in lower dimensions, a separate test for each dimension is required.

It is not always true that $\nu_t \leq m^{1/t}$, although this upper bound is valid when the points form a rectangular grid. For example, it turns out that $\nu_2 = \sqrt{274} > \sqrt{256}$ in Fig. 8, because a nearly hexagonal structure brings the m points closer together than would be possible in a strictly rectangular arrangement.

In order to develop an algorithm that computes ν_t efficiently, we must look more deeply at the associated mathematical theory. Therefore a reader who is not mathematically inclined is advised to skip to part D of this section, where the spectral test is presented as a "plug-in" method accompanied by several examples. On the other hand, we shall see that the mathematics behind the spectral test requires only some elementary manipulations of vectors.

Some authors have suggested using the minimum number N_t of parallel covering lines or hyperplanes as the criterion, instead of the maximum distance $1/\nu_t$ between them. However, this number does not appear to be as important as the concept of accuracy defined above, because it is biased by how nearly the slope of the lines or hyperplanes matches the coordinate axes of the cube. For example, the 20 nearly vertical lines that cover all the points of Fig. 8 are actually $1/\sqrt{328}$ units apart, and this might falsely imply an accuracy of one part in $\sqrt{328}$, or perhaps even of one part in 20. The true accuracy of only one part in $\sqrt{274}$ is realized only for the larger family of 21 lines with a slope of $7/15$; another family of 24 lines, with a slope of $-11/13$, also has a greater inter-line distance than the 20-line family, since $1/\sqrt{290} > 1/\sqrt{328}$. The precise way in which families of lines act at the boundaries of the unit hypercube does not seem to be an especially "clean" or significant criterion; however, for those people who prefer to count hyperplanes, it is possible to compute N_t using a method quite similar to the way in which we shall calculate ν_t (see exercise 16).

***B. Theory behind the test.** In order to analyze the basic set (2), we start with the observation that

$$\frac{1}{m} s^j(x) = \left(\frac{a^j x + (1 + a + \dots + a^{j-1})c}{m} \right) \bmod 1. \quad (5)$$

We can get rid of the "mod 1" operation by extending the set periodically, making infinitely many copies of the original t -dimensional hypercube, proceeding in all directions. This gives us the set

$$\begin{aligned} L &= \left\{ \left(\frac{x}{m} + k_1, \frac{s(x)}{m} + k_2, \dots, \frac{s^{t-1}(x)}{m} + k_t \right) \mid \text{integer } x, k_1, k_2, \dots, k_t \right\} \\ &= \left\{ V_0 + \left(\frac{x}{m} + k_1, \frac{ax}{m} + k_2, \dots, \frac{a^{t-1}x}{m} + k_t \right) \mid \text{integer } x, k_1, k_2, \dots, k_t \right\}, \end{aligned}$$

where

$$V_0 = \frac{1}{m}(0, c, (1+a)c, \dots, (1+a+\dots+a^{t-2})c) \quad (6)$$

is a constant vector. The variable k_1 is redundant in this representation of L , because we can change $(x, k_1, k_2, \dots, k_t)$ to $(x+k_1m, 0, k_2-ak_1, \dots, k_t-a^{t-1}k_1)$, reducing k_1 to zero without loss of generality. Therefore we obtain the comparatively simple formula

$$L = \{ V_0 + y_1V_1 + y_2V_2 + \dots + y_tV_t \mid \text{integer } y_1, y_2, \dots, y_t \}, \quad (7)$$

where

$$V_1 = \frac{1}{m}(1, a, a^2, \dots, a^{t-1}); \quad (8)$$

$$V_2 = (0, 1, 0, \dots, 0), \quad V_3 = (0, 0, 1, \dots, 0), \quad \dots, \quad V_t = (0, 0, 0, \dots, 1). \quad (9)$$

The points (x_1, x_2, \dots, x_t) of L that satisfy $0 \leq x_j < 1$ for all j are precisely the m points of our original set (2).

Note that the increment c appears only in V_0 , and the effect of V_0 is merely to shift all elements of L without changing their relative distances; hence c does not affect the spectral test in any way, and we might as well assume that $V_0 = (0, 0, \dots, 0)$ when we are calculating ν_t . When V_0 is the zero vector we have a so-called *lattice* of points

$$L_0 = \{ y_1V_1 + y_2V_2 + \dots + y_tV_t \mid \text{integer } y_1, y_2, \dots, y_t \}, \quad (10)$$

and our goal is to study the distances between adjacent $(t-1)$ -dimensional hyperplanes, in families of parallel hyperplanes that cover all the points of L_0 .

A family of parallel $(t-1)$ -dimensional hyperplanes can be defined by a nonzero vector $U = (u_1, \dots, u_t)$ that is perpendicular to all of them; and the set of points on a particular hyperplane is then

$$\{ (x_1, \dots, x_t) \mid x_1u_1 + \dots + x_tu_t = q \}, \quad (11)$$

where q is a different constant for each hyperplane in the family. In other words, each hyperplane is the set of all X for which the *dot product* $X \cdot U$ has a given value q . In our case the hyperplanes are all separated by a fixed distance, and one of them contains $(0, 0, \dots, 0)$; hence we can adjust the magnitude of U so that the set of all *integer* values q gives all the hyperplanes in the family. Then the distance between neighboring hyperplanes is the minimum distance from $(0, 0, \dots, 0)$ to the hyperplane for $q = 1$, namely

$$\min_{\text{real } x_1, \dots, x_t} \left\{ \sqrt{x_1^2 + \dots + x_t^2} \mid x_1u_1 + \dots + x_tu_t = 1 \right\}. \quad (12)$$

Cauchy's inequality (cf. exercise 1.2.3-30) tells us that

$$(x_1u_1 + \dots + x_tu_t)^2 \leq (x_1^2 + \dots + x_t^2)(u_1^2 + \dots + u_t^2), \quad (13)$$

hence the minimum in (12) occurs when each $x_j = u_j/(u_1^2 + \dots + u_t^2)$; the distance between neighboring hyperplanes is

$$1/\sqrt{u_1^2 + \dots + u_t^2} = 1/\text{length}(U). \quad (14)$$

In other words, the quantity ν_t we seek is precisely the length of the shortest vector U that defines a family of hyperplanes $\{X \cdot U = q \mid \text{integer } q\}$ containing all the elements of L_0 .

Such a vector $U = (u_1, \dots, u_t)$ must be nonzero, and it must satisfy $V \cdot U = \text{integer}$ for all V in L_0 . In particular, since the points $(1, 0, \dots, 0)$, $(0, 1, \dots, 0)$, \dots , $(0, 0, \dots, 1)$ are all in L_0 , all of the u_j must be integers. Furthermore since V_1 is in L_0 , we must have $\frac{1}{m}(u_1 + au_2 + \dots + a^{t-1}u_t) = \text{integer}$, i.e.,

$$u_1 + au_2 + \dots + a^{t-1}u_t \equiv 0 \pmod{m}. \quad (15)$$

Conversely, any nonzero integer vector $U = (u_1, \dots, u_t)$ satisfying (15) defines a family of hyperplanes with the required properties, since all of L_0 will be covered: $(y_1V_1 + \dots + y_tV_t) \cdot U$ will be an integer for all integers y_1, \dots, y_t . We have proved that

$$\begin{aligned} \nu_t^2 &= \min_{(u_1, \dots, u_t) \neq (0, \dots, 0)} \{u_1^2 + \dots + u_t^2 \mid u_1 + au_2 + \dots + a^{t-1}u_t \equiv 0 \pmod{m}\} \\ &= \min_{(x_1, \dots, x_t) \neq (0, \dots, 0)} ((mx_1 - ax_2 - a^2x_3 - \dots - a^{t-1}x_t)^2 + x_2^2 + x_3^2 + \dots + x_t^2). \end{aligned} \quad (16)$$

C. Deriving a computational method. We have now reduced the spectral test to the problem of finding the minimum value (16); but how on earth can we determine that minimum value in a reasonable amount of time? A brute-force search is out of the question, since m is very large in cases of practical interest.

It will be interesting and probably more useful if we develop a computational method for solving an even more general problem: *Find the minimum value of the quantity*

$$f(x_1, \dots, x_t) = (u_{11}x_1 + \dots + u_{t1}x_t)^2 + \dots + (u_{1t}x_1 + \dots + u_{tt}x_t)^2 \quad (17)$$

over all nonzero integer vectors (x_1, \dots, x_t) , given any nonsingular matrix of coefficients $U = (u_{ij})$. The expression (17) is called a "positive definite quadratic form" in t variables. Since U is nonsingular, (17) cannot be zero unless the x_j are all zero.

Let us write U_1, \dots, U_t for the rows of U . Then (17) may be written

$$f(x_1, \dots, x_t) = (x_1U_1 + \dots + x_tU_t) \cdot (x_1U_1 + \dots + x_tU_t), \quad (18)$$

the square of the length of the vector $x_1U_1 + \dots + x_tU_t$. The nonsingular matrix U has an inverse, which means that we can find uniquely determined vectors

V_1, \dots, V_t such that

$$U_i \cdot V_j = \delta_{ij}, \quad 1 \leq i, j \leq t. \quad (19)$$

For example, in the special form (16) that arises in the spectral test, we have

$$\begin{aligned} U_1 &= (m, 0, 0, \dots, 0), & V_1 &= \frac{1}{m}(1, a, a^2, \dots, a^{t-1}), \\ U_2 &= (-a, 1, 0, \dots, 0), & V_2 &= (0, 1, 0, \dots, 0), \\ U_3 &= (-a^2, 0, 1, \dots, 0), & V_3 &= (0, 0, 1, \dots, 0), \\ &\vdots & & \vdots \\ U_t &= (-a^{t-1}, 0, 0, \dots, 1), & V_t &= (0, 0, 0, \dots, 1). \end{aligned} \quad (20)$$

These V_j are precisely the vectors (8), (9) that we used to define our original lattice L_0 . As the reader may well suspect, this is not a coincidence—indeed, if we had begun with an arbitrary lattice L_0 , defined by any set of linearly independent vectors V_1, \dots, V_t , the argument we have used above can be generalized to show that the maximum separation between hyperplanes in a covering family is equivalent to minimizing (17), where the coefficients u_{ij} are defined by (19). (See exercise 2.)

Our first step in minimizing (18) is to reduce it to a finite problem, i.e., to show that we won't need to test infinitely many vectors (x_1, \dots, x_t) to find the minimum. This is where the vectors V_1, \dots, V_t come in handy; we have

$$x_k = (x_1 U_1 + \dots + x_t U_t) \cdot V_k,$$

and Cauchy's inequality tells us that

$$((x_1 U_1 + \dots + x_t U_t) \cdot V_k)^2 \leq f(x_1, \dots, x_t)(V_k \cdot V_k).$$

Hence we have derived a useful upper bound on each coordinate x_k :

Lemma A. *Let (x_1, \dots, x_t) be a nonzero vector that minimizes (18) and let (y_1, \dots, y_t) be any nonzero integer vector. Then*

$$x_k^2 \leq (V_k \cdot V_k)f(y_1, \dots, y_t), \quad \text{for } 1 \leq k \leq t. \quad (21)$$

In particular, letting $y_i = \delta_{ij}$ for all i ,

$$x_k^2 \leq (V_k \cdot V_k)(U_j \cdot U_j), \quad \text{for } 1 \leq j, k \leq t. \quad \blacksquare \quad (22)$$

Lemma A reduces the problem to a finite search, but the right-hand side of (21) is usually much too large to make an exhaustive search feasible; we need at least one more idea. On such occasions, an old maxim provides sound advice: "If you can't solve a problem as it is stated, change it into a simpler problem that

has the same answer." For example, Euclid's algorithm has this form; if we don't know the gcd of the input numbers, we change them into smaller numbers having the same gcd. (In fact, a slightly more general approach probably underlies the discovery of nearly all algorithms: "If you can't solve a problem directly, change it into one or more simpler problems, from whose solution you can solve the original one.")

In our case, a simpler problem is one that requires less searching because the right-hand side of (22) is smaller. The key idea we shall use is that it is possible to change one quadratic form into another one that is equivalent for all practical purposes. Let j be any fixed subscript, $1 \leq j \leq t$; let $(q_1, \dots, q_{j-1}, q_{j+1}, \dots, q_t)$ be any sequence of $t-1$ integers; and consider the following transformation of the vectors:

$$\begin{aligned} V_i' &= V_i - q_i V_j, & x_i' &= x_i - q_i x_j, & U_i' &= U_i, & \text{for } i \neq j; \\ V_j' &= V_j, & x_j' &= x_j, & U_j' &= U_j + \sum_{i \neq j} q_i U_i. \end{aligned} \quad (23)$$

It is easy to see that the new vectors U_1', \dots, U_t' define a quadratic form f' for which $f'(x_1', \dots, x_t') = f(x_1, \dots, x_t)$; furthermore the basic orthogonality condition (19) remains valid, because it is easy to check that $U_i' \cdot V_j' = \delta_{ij}$. As (x_1, \dots, x_t) runs through all nonzero integer vectors, so does (x_1', \dots, x_t') ; hence the new form f' has the same minimum as f .

Our goal is to use transformation (23), replacing U_i by U_i' and V_i by V_i' for all i , in order to make the right-hand side of (22) small; and the right-hand side of (22) will be small when both $U_j \cdot U_j$ and $V_k \cdot V_k$ are small. Therefore it is natural to ask the following two questions about the transformation (23):

- a) What choice of q_i makes $V_i' \cdot V_i'$ as small as possible?
- b) What choice of $q_1, \dots, q_{j-1}, q_{j+1}, \dots, q_t$ makes $U_j' \cdot U_j'$ as small as possible?

It is easiest to solve these questions first for *real* values of the q_i . Question (a) is quite simple, since

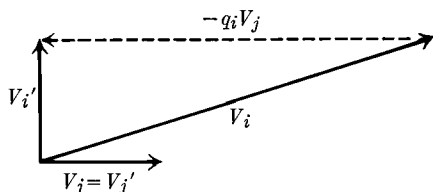
$$\begin{aligned} (V_i - q_i V_j) \cdot (V_i - q_i V_j) &= V_i \cdot V_i - 2q_i V_i \cdot V_j + q_i^2 V_j \cdot V_j \\ &= (V_j \cdot V_j) \left(q_i - (V_i \cdot V_j / V_j \cdot V_j) \right)^2 \\ &\quad + V_i \cdot V_i - (V_i \cdot V_j)^2 / V_j \cdot V_j, \end{aligned}$$

and the minimum occurs when

$$q_i = V_i \cdot V_j / V_j \cdot V_j. \quad (24)$$

Geometrically, we are asking what multiple of V_j should be subtracted from V_i so that the resulting vector V_i' has minimum length, and the answer is to choose q_i so that V_i' is perpendicular to V_j (i.e., so that $V_i' \cdot V_j = 0$); the following

diagram makes this plain.



$$(25)$$

Turning to question (b), we want to choose the q_i so that $U_j + \sum_{i \neq j} q_i U_i$ has minimum length; geometrically, we want to start with U_j and add some vector in the $(t - 1)$ -dimensional hyperplane whose points are the sums of multiples of $\{U_i \mid i \neq j\}$. Again the best solution is to choose things so that U_j' is perpendicular to the hyperplane, i.e., so that $U_j' \cdot U_k = 0$ for all $k \neq j$, i.e.,

$$U_j \cdot U_k + \sum_{i \neq j} q_i (U_i \cdot U_k) = 0, \quad 1 \leq k \leq t, \quad k \neq j. \quad (26)$$

(See exercise 12 for a rigorous proof that a solution to question (b) must satisfy these $t - 1$ equations.)

Now that we have answered questions (a) and (b), we are in a bit of a quandary; should we choose the q_i according to (24), so that the $V_i' \cdot V_i'$ are minimized, or according to (26), so that $U_j' \cdot U_j'$ is minimized? Either of these alternatives makes an improvement in the right-hand side of (22), so it is not immediately clear which choice should get priority. Fortunately, there is a very simple answer to this dilemma: Conditions (24) and (26) are exactly the same! (See exercise 7.) Therefore questions (a) and (b) have the same answer; we have a happy state of affairs in which we can reduce the length of both the U 's and the V 's simultaneously. (It may be worthwhile to point out that we have just rediscovered the "Schmidt orthogonalization process.")

Our joy must be tempered with the realization that we have dealt with questions (a) and (b) only for *real* values of the q_i . Our application restricts us to integer values, so we cannot make V_i' exactly perpendicular to V_j . The best we can do for question (a) is to let q_i be the *nearest integer* to $V_i \cdot V_j / V_j \cdot V_j$ (cf. (25)). It turns out that this is *not* always the best solution to question (b); in fact U_j' may at times be longer than U_j . However, the bound (21) is never increased, since we can remember the smallest value of $f(y_1, \dots, y_t)$ found so far. Thus a choice of q_i based solely on question (a) is quite satisfactory.

If we apply transformation (23) repeatedly in such a way that none of the vectors V_i gets longer and at least one gets shorter, we can never get into a loop; i.e., we will never be considering the same quadratic form again after a sequence of nontrivial transformations of this kind. But eventually we will get "stuck," in the sense that none of the transformations (23) for $1 \leq j \leq t$ will be able to shorten any of the vectors V_1, \dots, V_t . At that point we can revert to an exhaustive search, using the bounds of Lemma A, which will now

be quite small in most cases. Occasionally these bounds (21) will be poor, and another type of transformation will usually get the algorithm unstuck again and reduce the bounds (see exercise 18). However, transformation (23) by itself has proved to be quite adequate for the spectral test; in fact, it has proved to be amazingly powerful when the computations are arranged as in the algorithm discussed below.

***D. How to perform the spectral test.** Here now is an efficient computational procedure that follows from our considerations. R. W. Gosper and U. Dieter have observed that it is possible to use the results of lower dimensions to make the spectral test significantly faster in higher dimensions. This refinement has been incorporated into the following algorithm, together with a significant simplification of the two-dimensional case.

Algorithm S (*The spectral test*). This algorithm determines the value of

$$\nu_t = \min \left\{ \sqrt{x_1^2 + \cdots + x_t^2} \mid x_1 + ax_2 + \cdots + a^{t-1}x_t \equiv 0 \pmod{m} \right\} \quad (27)$$

for $2 \leq t \leq T$, given a , m , and T , where $0 < a < m$ and a is relatively prime to m . (The number ν_t measures the t -dimensional accuracy of random number generators, as discussed in the text above.) All arithmetic within this algorithm is done on integers whose magnitudes rarely if ever exceed m^2 , except in step S8; in fact, nearly all of the integer variables will be less than m in absolute value during the computation.

When ν_t is being calculated for $t \geq 3$, the algorithm works with two $t \times t$ matrices U and V , whose row vectors are denoted by $U_i = (u_{i1}, \dots, u_{it})$ and $V_i = (v_{i1}, \dots, v_{it})$ for $1 \leq i \leq t$. These vectors satisfy the conditions

$$u_{i1} + au_{i2} + \cdots + a^{t-1}u_{it} \equiv 0 \pmod{m}, \quad 1 \leq i \leq t; \quad (28)$$

$$U_i \cdot V_j = \delta_{ij}m, \quad 1 \leq i, j \leq t. \quad (29)$$

(Thus the V_j of our previous discussion have been multiplied by m , to ensure that their components are integers.) There are three other auxiliary vectors, $X = (x_1, \dots, x_t)$, $Y = (y_1, \dots, y_t)$, and $Z = (z_1, \dots, z_t)$. During the entire algorithm, r will denote $a^{t-1} \bmod m$ and s will denote the smallest upper bound for ν_t^2 that has been discovered so far.

S1. [Initialize.] Set $h \leftarrow a$, $h' \leftarrow m$, $p \leftarrow 1$, $p' \leftarrow 0$, $r \leftarrow a$, $s \leftarrow 1 + a^2$. (The first steps of this algorithm handle the case $t = 2$ by a special method, very much like Euclid's algorithm; we will have

$$h - ap \equiv h' - ap' \equiv 0 \pmod{m} \quad \text{and} \quad hp' - h'p = \pm m \quad (30)$$

during this phase of the calculation.)

S2. [Euclidean step.] Set $q \leftarrow \lfloor h'/h \rfloor$, $u \leftarrow h' - qh$, $v \leftarrow p' - qp$. If $u^2 + v^2 < s$, set $s \leftarrow u^2 + v^2$, $h' \leftarrow h$, $h \leftarrow u$, $p' \leftarrow p$, $p \leftarrow v$, and repeat step S2.

- S3.** [Compute ν_2 .] Set $u \leftarrow u - h$, $v \leftarrow v - p$; and if $u^2 + v^2 < s$, set $s \leftarrow u^2 + v^2$, $h' \leftarrow u$, $p' \leftarrow v$. Then output $\sqrt{s} = \nu_2$. (The validity of this calculation for the two-dimensional case is proved in exercise 5. Now we will set up the U and V matrices satisfying (28) and (29), in preparation for calculations in higher dimensions.) Set

$$U \leftarrow \begin{pmatrix} -h & p \\ -h' & p' \end{pmatrix}, \quad V \leftarrow \pm \begin{pmatrix} p' & h' \\ -p & -h \end{pmatrix},$$

where the $-$ sign is chosen for V if and only if $p' > 0$.

- S4.** [Advance t .] If $t = T$, the algorithm terminates. (Otherwise we want to increase t by 1. At this point U and V are $t \times t$ matrices satisfying (28) and (29), and we must enlarge them by adding an appropriate new row and column.) Set $t \leftarrow t + 1$ and $r \leftarrow (ar) \bmod m$. Set U_t to the new row $(-r, 0, 0, \dots, 0, 1)$ of t elements, and set $u_{it} \leftarrow 0$ for $1 \leq i < t$. Set V_t to the new row $(0, 0, 0, \dots, 0, m)$. Finally, for $1 \leq i < t$, set $q \leftarrow \text{round}(v_{i1} r/m)$, $v_{it} \leftarrow v_{i1} r - qm$, and $U_t \leftarrow U_t + qU_i$. (Here “round(x)” denotes the nearest integer to x , e.g., $\lfloor x + 1/2 \rfloor$. We are essentially setting $v_{it} \leftarrow v_{i1} r$ and immediately applying transformation (23) with $j = t$, since the numbers $|v_{i1} r|$ are so large they ought to be reduced at once.) Finally set $s \leftarrow \min(s, U_t \cdot U_t)$, $k \leftarrow t$, and $j \leftarrow 1$. (In the following steps, j denotes the current row index for transformation (23), and k denotes the last such index where the transformation shortened at least one of the V_i .)
- S5.** [Transform.] For $1 \leq i \leq t$, do the following operations: If $i \neq j$ and $2|V_i \cdot V_j| > V_j \cdot V_j$, set $q \leftarrow \text{round}(V_i \cdot V_j / V_j \cdot V_j)$, $V_i \leftarrow V_i - qV_j$, $U_j \leftarrow U_j + qU_i$, and $k \leftarrow j$. (The fact that we omit this transformation, when $2|V_i \cdot V_j|$ exactly equals $V_j \cdot V_j$, prevents the algorithm from looping endlessly; see exercise 19.)
- S6.** [Examine new bound.] If $k = j$ (i.e., if the transformation in S5 has just done something useful), set $s \leftarrow \min(s, U_j \cdot U_j)$.
- S7.** [Advance j .] If $j = t$, set $j \leftarrow 1$; otherwise set $j \leftarrow j + 1$. Now if $j \neq k$, return to step S5. (If $j = k$, we have gone through $t - 1$ consecutive cycles of no transformation, so the transformation process is stuck.)
- S8.** [Prepare for search.] (Now the absolute minimum will be determined, using an exhaustive search over all (x_1, \dots, x_t) satisfying condition (21) of Lemma A.) Set $X \leftarrow Y \leftarrow (0, \dots, 0)$, set $k \leftarrow t$, and set

$$z_j \leftarrow \left\lfloor \sqrt{[(V_j \cdot V_j)s/m^2]} \right\rfloor, \quad \text{for } 1 \leq j \leq t. \quad (31)$$

(We will examine all $X = (x_1, \dots, x_t)$ with $|x_j| \leq z_j$ for $1 \leq j \leq t$. In hundreds of applications of this algorithm, no z_j has yet turned out to be greater than 1, nor has the exhaustive search in the following steps ever reduced s ; however, such phenomena are probably possible in weird cases,

especially in higher dimensions. During the exhaustive search, the vector Y will always be equal to $x_1U_1 + \cdots + x_tU_t$, so that $f(x_1, \dots, x_t) = Y \cdot Y$. Since $f(-x_1, \dots, -x_t) = f(x_1, \dots, x_t)$, we shall examine only vectors whose first nonzero component is positive. The method is essentially that of counting in steps of one, regarding (x_1, \dots, x_t) as the digits in a balanced number system with mixed radices $(2z_1 + 1, \dots, 2z_t + 1)$; cf. Section 4.1.)

S9. [Advance x_k .] If $x_k = z_k$, go to S11. Otherwise increase x_k by 1 and set $Y \leftarrow Y + U_k$.

S10. [Advance k .] Set $k \leftarrow k + 1$. Then if $k \leq t$, set $x_k \leftarrow -z_k$, $Y \leftarrow Y - 2z_kU_k$, and repeat step S10. But if $k > t$, set $s \leftarrow \min(s, Y \cdot Y)$.

S11. [Decrease k .] Set $k \leftarrow k - 1$. If $k \geq 1$, return to S9. Otherwise output $\nu_t = \sqrt{s}$ (the exhaustive search is completed) and return to S4. ■

In practice Algorithm S is applied for $T = 5$ or 6, say; it usually works reasonably well when $T = 7$ or 8, but it can be terribly slow when $T \geq 9$ since the exhaustive search tends to make the running time grow as 3^T . (If the minimum value ν_t occurs at many different points, the exhaustive search will hit them all; hence we typically find that all $z_k = 1$ for large t . As remarked above, the values of ν_t are generally irrelevant for practical purposes when t is large.)

An example will help to make Algorithm S clear. Consider the linear congruential sequence defined by

$$m = 10^{10}, \quad a = 3141592621, \quad c = 1, \quad X_0 = 0. \quad (32)$$

Six cycles of the Euclidean algorithm in steps S2 and S3 suffice to prove that the minimum nonzero value of $x_1^2 + x_2^2$ with

$$x_1 + 3141592621x_2 \equiv 0 \pmod{10^{10}}$$

occurs for $x_1 = 67654$, $x_2 = 226$; hence the two-dimensional accuracy of this generator is

$$\nu_2 = \sqrt{67654^2 + 226^2} \approx 67654.37748.$$

Passing to three dimensions, we seek the minimum nonzero value of $x_1^2 + x_2^2 + x_3^2$ such that

$$x_1 + 3141592621x_2 + 3141592621^2x_3 \equiv 0 \pmod{10^{10}}. \quad (33)$$

Step S4 sets up the matrices

$$U = \begin{pmatrix} -67654 & -226 & 0 \\ -44190611 & 191 & 0 \\ 5793866 & 33 & 1 \end{pmatrix}, \quad V = \begin{pmatrix} -191 & -44190611 & 2564918569 \\ -226 & 67654 & 1307181134 \\ 0 & 0 & 10000000000 \end{pmatrix}.$$

The first iteration of step S5, with $q = 1$ for $i = 2$ and $q = 4$ for $i = 3$, changes them to

$$U = \begin{pmatrix} -21082801 & 97 & 4 \\ -44190611 & 191 & 0 \\ 5793866 & 33 & 1 \end{pmatrix}, \quad V = \begin{pmatrix} -191 & -44190611 & 2564918569 \\ -35 & 44258265 & -1257737435 \\ 764 & 176762444 & -259674276 \end{pmatrix}.$$

(Note that the first row U_1 has actually gotten longer in this transformation, although eventually the rows of U should get shorter.)

The next fourteen iterations of step S5 have $(j, q_1, q_2, q_3) = (2, -2, *, 0)$, $(3, 0, 3, *)$, $(1, *, -10, -1)$, $(2, -1, *, -6)$, $(3, -1, 0, *)$, $(1, *, 0, 2)$, $(2, 0, *, -1)$, $(3, 3, 4, *)$, $(1, *, 0, 0)$, $(2, -5, *, 0)$, $(3, 1, 0, *)$, $(1, *, -3, -1)$, $(2, 0, *, 0)$, $(3, 0, 0, *)$. Now the transformation process is stuck, but the rows of the matrices have become significantly shorter:

$$U = \begin{pmatrix} -1479 & 616 & -2777 \\ -3022 & 104 & 918 \\ -227 & -983 & -130 \end{pmatrix}, \quad V = \begin{pmatrix} -888874 & 601246 & -2994234 \\ -2809871 & 438109 & 1593689 \\ -854296 & -9749816 & -1707736 \end{pmatrix}. \quad (34)$$

The search limits (z_1, z_2, z_3) in step S8 turn out to be $(0, 0, 1)$, so U_3 is the shortest solution to (33); we have

$$\nu_3 = \sqrt{227^2 + 983^2 + 130^2} \approx 1017.21089.$$

Note that only a few iterations were needed to find this value, although condition (33) looks quite difficult to deal with at first glance. All points (U_n, U_{n+1}, U_{n+2}) produced by this random number generator lie on a family of parallel planes about 0.001 units apart.

E. Ratings for various generators. So far we haven't really given a criterion that tells us whether or not a particular random number generator "passes" or "flunks" the spectral test. In fact, this depends on the application, since some applications demand higher resolution than others. It appears that $\nu_t \geq 2^{30/t}$ for $2 \leq t \leq 6$ will be quite adequate in most applications (although the author must admit choosing this criterion partly because 30 is conveniently divisible by 2, 3, 5, and 6).

For some purposes we would like a criterion that is relatively independent of m , so we can say that a particular multiplier is good or bad with respect to the set of all other multipliers for the given m , without examining any others. A reasonable figure of merit for rating the goodness of a particular multiplier seems to be the volume of the ellipsoid in t -space defined by the relation $(x_1 m - x_2 a - \dots - x_t a^{t-1})^2 + x_2^2 + \dots + x_t^2 \leq \nu_t^2$, since this volume tends to indicate how likely it is that nonzero integer points (x_1, \dots, x_t) —corresponding to solutions of (15)—are in the ellipsoid. We therefore propose to calculate this volume, namely

$$\mu_t = \frac{\pi^{t/2} \nu_t^t}{(t/2)! m}, \quad (35)$$

as an indication of the effectiveness of the multiplier a for the given m . In this formula,

$$\left(\frac{t}{2}\right)! = \left(\frac{t}{2}\right) \left(\frac{t}{2} - 1\right) \dots \left(\frac{1}{2}\right) \sqrt{\pi}, \quad \text{for } t \text{ odd.} \quad (36)$$

Table 1

SAMPLE RESULTS OF THE SPECTRAL TEST

Line	a	m	ν_2^2	ν_3^2	ν_4^2	ν_5^2	ν_6^2
1	23	$10^8 + 1$	530	530	530	530	447
2	$2^7 + 1$	2^{35}	16642	16642	16642	15602	252
3	$2^{18} + 1$	2^{35}	34359738368	6	4	4	4
4	3141592653	2^{35}	2997222016	1026050	27822	1118	1118
5	137	256	274	30	14	6	4
6	3141592621	10^{10}	4577114792	1034718	62454	1776	542
7	3141592221	10^{10}	4293881050	276266	97450	3366	2382
8	4219755981	10^{10}	10721093248	2595578	49362	5868	820
9	4160984121	10^{10}	9183801602	4615650	16686	6840	1344
10	3141592221	2^{35}	13539813818	5795090	88134	12716	2938
11	2718281829	2^{35}	22939188896	2723830	146116	10782	2914
12	5^{13}	2^{35}	33161885770	2925242	113374	13070	2256
13	5^{15}	2^{35}	22078865098	10274746	167558	5844	2592
14	$2^{23} + 2^{12} + 5$	2^{35}	167510120	8052254	21476	16802	1630
15	$2^{23} + 2^{13} + 5$	2^{35}	168231328	5335322	21476	2008	1134
16	$2^{23} + 2^{14} + 5$	2^{35}	12256151168	5733878	21476	13316	2032
17	$2^{22} + 2^{13} + 5$	2^{35}	8201443840	1830230	21476	7786	3080
18	$2^{24} + 2^{13} + 5$	2^{35}	8364058	8364058	21476	16712	1496
19	19935388837	2^{35}	32300850938	705518	22270	9558	2660
20	1175245817	2^{35}	36436418002	7362242	95306	3006	2860
21	17059465	2^{35}	39341117000	9476606	202796	18758	2382
22	$2^{16} + 3$	2^{29}	536805386	118	116	116	116
23	1812433253	2^{32}	4326934538	1462856	15082	4866	906
24	1566083941	2^{32}	4659748970	2079590	44902	4652	662
25	69069	2^{32}	4243209856	2072544	52804	6990	242
26	1664525	2^{32}	4938916874	2322494	63712	4092	1038
27	314159269	$2^{31} - 1$	1432232969	899290	36985	3427	1144
28	see (39)		$(2^{31} - 1)^2$	1.4×10^{12}	643578623	12930027	837632
29	31167285	2^{48}	3.2×10^{14}	4111841446	17341510	306326	59278
30	see the text	2^{64}	8.8×10^{18}	6.4×10^{12}	4.1×10^9	45662836	1846368

Thus, in six or fewer dimensions the merit is computed as follows:

$$\begin{aligned} \mu_2 &= \pi \nu_2^2 / m, & \mu_3 &= \frac{4}{3} \pi \nu_3^3 / m, & \mu_4 &= \frac{1}{2} \pi^2 \nu_4^4 / m, \\ \mu_5 &= \frac{8}{15} \pi^2 \nu_5^5 / m, & \mu_6 &= \frac{1}{6} \pi^3 \nu_6^6 / m. \end{aligned} \quad (37)$$

We might say that the multiplier a passes the spectral test if μ_t is 0.1 or more for $2 \leq t \leq 6$, and it "passes with flying colors" if $\mu_t \geq 1$ for all these t . A low value of μ_t means that we have probably picked a very unfortunate multiplier, since very few lattices will have integer points so close to the origin. Conversely, a high value of μ_t means that we have found an unusually good multiplier for the given m ; but it does not mean that the random numbers are necessarily very good, since m might be too small. Only the values ν_t truly indicate the degree of randomness.

$(\epsilon = \frac{1}{10})$

$\lg \nu_2$	$\lg \nu_3$	$\lg \nu_4$	$\lg \nu_5$	$\lg \nu_6$	μ_2	μ_3	μ_4	μ_5	μ_6	Line
4.5	4.5	4.5	4.5	4.4	$2\epsilon^5$	$5\epsilon^4$	0.01	0.34	4.62	1
7.0	7.0	7.0	7.0	4.0	$2\epsilon^6$	$3\epsilon^4$	0.04	4.66	$2\epsilon^3$	2
17.5	1.3	1.0	1.0	1.0	3.14	$2\epsilon^9$	$2\epsilon^9$	$5\epsilon^9$	ϵ^8	3
15.7	10.0	7.4	5.0	5.0	0.27	0.13	0.11	0.01	0.21	4
4.0	2.5	1.9	1.3	1.0	3.36	2.69	3.78	1.81	1.29	5
16.0	10.0	8.0	5.4	4.5	1.44	0.44	1.92	0.07	0.08	6
16.0	9.0	8.3	5.9	5.6	1.35	0.06	4.69	0.35	6.98	7
16.7	10.7	7.8	6.3	4.8	3.39	1.75	1.20	1.39	0.28	8
16.5	11.1	7.0	6.4	5.2	2.89	4.15	0.14	2.04	1.25	9
16.8	11.2	8.2	6.8	5.8	1.24	1.70	1.12	2.79	3.81	10
17.2	10.7	8.6	6.7	5.8	2.10	0.55	3.15	1.85	3.72	11
17.5	10.7	8.4	6.8	5.6	3.03	0.61	1.85	2.99	1.73	12
17.2	11.6	8.7	6.3	5.7	2.02	4.02	4.03	0.40	2.62	13
13.7	11.5	7.2	7.0	5.3	0.02	2.79	0.07	5.61	0.65	14
13.7	11.2	7.2	5.5	5.1	0.02	1.50	0.07	0.03	0.22	15
16.8	11.2	7.2	6.9	5.5	1.12	1.67	0.07	3.13	1.26	16
16.5	10.4	7.2	6.5	5.8	0.75	0.30	0.07	0.82	4.39	17
11.5	11.5	7.2	7.0	5.3	$8\epsilon^4$	2.95	0.07	5.53	0.50	18
17.5	9.7	7.2	6.6	5.7	2.95	0.07	0.07	1.37	2.83	19
17.5	11.4	8.3	5.8	5.7	3.33	2.44	1.30	0.08	3.52	20
17.6	11.6	8.8	7.1	5.6	3.60	3.56	5.91	7.38	2.03	21
14.5	3.4	3.4	3.4	3.4	3.14	ϵ^5	ϵ^4	ϵ^3	0.02	22
16.0	10.2	6.9	6.1	4.9	3.16	1.73	0.26	2.02	0.89	23
16.1	10.5	7.7	6.1	4.7	3.41	2.92	2.32	1.81	0.35	24
16.0	10.5	7.8	6.4	4.0	3.10	2.91	3.20	5.01	0.02	25
16.1	10.6	8.0	6.0	5.0	3.61	3.45	4.66	1.31	1.35	26
15.2	9.9	7.6	5.9	5.1	2.10	1.66	3.14	1.69	3.60	27
31.0	20.2	15.6	11.8	9.8	3.14	1.49	0.44	0.69	0.66	28
24.1	16.0	12.0	9.1	7.9	3.60	3.92	5.27	0.97	3.82	29
31.5	21.3	16.0	12.7	10.4	1.50	3.68	4.52	4.02	1.76	30

upper bounds from (40): 3.63 5.92 9.87 14.89 23.87

Table 1 shows what sorts of values occur in typical sequences. Each line of the table considers a particular generator, and lists ν_t , μ_t , and the “number of bits of accuracy” $\lg \nu_t$. Lines 1 through 4 show the generators that were the subject of Figs. 2 and 5 in Section 3.3.1. The generators in lines 1 and 2 suffer from too small a multiplier; a diagram like Fig. 8 will have a nearly vertical “stripes” when a is small. The terrible generator in line 3 has a good μ_2 but very poor μ_3 and μ_4 ; like nearly all generators of potency 2, it has $\nu_3 = \sqrt{6}$ and $\nu_4 = 2$ (see exercise 3). Line 4 shows a “random” multiplier; this generator has satisfactorily passed numerous empirical tests for randomness, but it does not have especially high values of μ_2, \dots, μ_6 . In fact, the value of μ_5 flunks our criterion.

Line 5 shows the generator of Fig. 8. It passes the spectral test with very high-flying colors, when μ_2 through μ_6 are considered, but of course m is so

small that the numbers can hardly be called random; the ν_t values are terribly low.

Line 6 is the generator discussed above; line 7 is a similar example, having an abnormally low value of μ_3 . Line 8 shows a nonrandom multiplier for the same modulus m ; all of its partial quotients are 1, 2, or 3. Such multipliers have been suggested by I. Borosh and H. Niederreiter because the Dedekind sums are likely to be especially small and because they produce best results in the two-dimensional serial test (cf. Section 3.3.3 and exercise 30). The particular example in line 8 has only one '3' as a partial quotient; there is no multiplier congruent to 1 modulo 20 whose partial quotients with respect to 10^{10} are only 1's and 2's. The generator in line 9 shows another multiplier chosen with malice aforethought, following a suggestion by A. G. Waterman that guarantees a reasonably high value of μ_2 (see exercise 11).

Lines 10 through 21 of Table 1 show further examples with $m = 2^{35}$, beginning with some random multipliers. The generators in lines 12 and 13 are reminders of the good old days—they were once used extensively since O. Taussky first suggested them in the early 1950s. Lines 14 through 18 show various multipliers of maximum potency having only four 1's in their binary representation. The point of having few 1's is to replace multiplication by a few additions, but only line 16 comes near to being passable. Since these multipliers satisfy $(a - 5)^3 \bmod 2^{35} = 0$, all five of them achieve ν_4 at the same point $(x_1, x_2, x_3, x_4) = (-125, 75, -15, 1)$. Another curiosity is the high value of μ_3 following a very low μ_2 in line 18 (see exercise 8). Lines 19 and 20 are respectively the Borosh–Niederreiter and Waterman multipliers for modulus 2^{35} ; and line 21 was found by M. Lavaux and F. Janssens, in a computer search for spectrally good multipliers having a very high μ_2 .

Lines 22 through 28 apply to System/370 and other machines with 32-bit words; in this case the comparatively small word size calls for comparatively greater care. Line 22 is, regrettably, the generator that has actually been used on such machines in most of the world's scientific computing centers for about a decade; its very name RANDU is enough to bring dismay into the eyes and stomachs of many computer scientists! The actual generator is defined by

$$X_0 \text{ odd,} \quad X_{n+1} = (65539X_n) \bmod 2^{31}, \quad (38)$$

and exercise 20 indicates that 2^{29} is the appropriate modulus for the spectral test. Since $9X_n + 6X_{n+2} + X_{n+2} \equiv 0 \pmod{2^{31}}$, the generator fails most three-dimensional criteria for randomness, and it should never have been used. Almost any multiplier $\equiv 5 \pmod{8}$ would be better. (A curious fact about RANDU, noticed by R. W. Gosper, is that $\nu_4 = \nu_5 = \nu_6 = \nu_7 = \nu_8 = \nu_9 = \sqrt{116}$, hence μ_9 is a spectacular 11.98.) Lines 23 and 24 are the Borosh–Niederreiter and Waterman multipliers for modulus 2^{32} , lines 26 and 29 were found by Lavaux and Janssens, and line 30 (whose excellent multiplier 6364136223846793005 is too big to fit in the column) is due to C. E. Haynes. Line 25 was nominated by George Marsaglia as "a candidate for the best of all multipliers," after a computer search

in dimensions 2 through 5, partly because it is easy to remember. Line 27 uses a random primitive root, modulo the prime $2^{31} - 1$, as multiplier. Line 28 is for the sequence

$$X_n = (271828183X_{n-1} - 314159269X_{n-2}) \bmod (2^{31} - 1), \quad (39)$$

which can be shown to have period length $(2^{31} - 1)^2 - 1$; it has been analyzed with the generalized spectral test of exercise 24.

Theoretical upper bounds on μ_t , which can never be transcended for any m , are shown just below Table 1; it is known that every lattice with m points per unit volume has

$$\nu_t \leq \gamma_t m^{1/t}, \quad (40)$$

where γ_t takes the respective values

$$(4/3)^{1/4}, \quad 2^{1/6}, \quad 2^{1/4}, \quad 2^{3/10}, \quad (64/3)^{1/12}, \quad 2^{3/7}, \quad 2^{1/2} \quad (41)$$

for $t = 2, \dots, 8$. (See exercise 9 and J. W. S. Cassels, *Introduction to the Geometry of Numbers* (Berlin: Springer, 1959), p. 332.) These bounds hold for lattices generated by vectors with arbitrary real coordinates. For example, the optimum lattice for $t = 2$ is hexagonal, and it is generated by vectors of length $2/\sqrt{3}m$ that form two sides of an equilateral triangle. In three dimensions the optimum lattice is generated by vectors V_1, V_2, V_3 that can be rotated into the form $(v, v, -v), (v, -v, v), (-v, v, v)$, where $v = 1/\sqrt[3]{4m}$.

***F. Relation to the serial test.** In a series of important papers published during the 1970s, Harald Niederreiter has shown how to analyze the distribution of the t -dimensional vectors (1) by means of exponential sums. One of the main consequences of his theory is that the serial test in several dimensions will be passed by any generator that passes the spectral test, even when we consider only a sufficiently large part of the period instead of the whole period. We shall now turn briefly to a study of his interesting methods, in the case of linear congruential sequences (X_0, a, c, m) of period length m .

The first idea we need is the notion of *discrepancy* in t dimensions, a quantity that we shall define as the difference between the expected number and the actual number of t -dimensional vectors $(x_n, x_{n+1}, \dots, x_{n+t-1})$ falling into a hyperrectangular region, maximized over all such regions. To be precise, let $\langle x_n \rangle$ be a sequence of integers in the range $0 \leq x_n < m$. We define

$$D_N^{(t)} = \max_R \left| \frac{\text{number of } (x_n, \dots, x_{n+t-1}) \text{ in } R \text{ for } 0 \leq n < N}{N} - \frac{\text{volume of } R}{m^t} \right| \quad (42)$$

where R ranges over all sets of points of the form

$$R = \{ (y_1, \dots, y_t) \mid \alpha_1 \leq y_1 < \beta_1, \dots, \alpha_t \leq y_t < \beta_t \}; \quad (43)$$

here α_j and β_j are integers in the range $0 \leq \alpha_j < \beta_j \leq m$, for $1 \leq j \leq t$. The volume of R is clearly $(\beta_1 - \alpha_1) \dots (\beta_t - \alpha_t)$. To get the discrepancy $D_N^{(t)}$, we imagine looking at all these sets R and finding the one with the greatest excess or deficiency of points (x_n, \dots, x_{n+t-1}) .