Throwing nature's dice

Ricardo Aguayo,[a] Geoff Simms, and P. B. Siegel
Physics Department, California State Polytechnic University Pomona, Pomona, California 91768

(Received 12 May 1995; accepted 20 September 1995)

A simple experimental setup to produce true random numbers is described. The experiment involves measuring successive times between decays of a radioactive source. We discuss two different ways of comparing these times: one that generates a string of random zeros and ones, and another that produces numbers derivable from the permutation group. It is shown that the methods used in the experiment produce the same results for any random process and can therefore be used as a test for randomness in other physical systems. © *1996 American Association of Physics Teachers.*

I. INTRODUCTION

Random number generation is of interest in Monte Carlo calculations, computer simulations, and other applications in physics. Due to the need of rapid random number production, a lot of study has gone into developing computer algorithms for this purpose. The numbers generated by these computer programs are not truly random, but are referred to as "pseudo-random." There has been less effort in developing true random number generators, since they do not offer many advantages over pseudo-random numbers in applications, and they are not as fast as a built-in computer algorithm. We found only a few articles in the literature about true random number generation, mostly in electronics journals.[1–3] A common method is to use white electronic noise, or radioactive decay, to produce a random bit stream. The signal is sampled by means of flip-flops at equal time intervals to produce the random bits. These methods have imperfections which introduce small correlations.[1] Another method presented involves detecting the different isotopes of a gas at equilibrium as the molecules escape through a narrow nozzle.[3] We also discovered a patent from 1969 that compares voltage measurements of electronic noise to produce random bits.[4] These approaches focus on the problem of producing true random numbers fast using hardware techniques. However from a teaching standpoint, it is interesting for physics students to consider ways in which truly random numbers can be generated from physical systems, and conversely to test the randomness of these systems.

In designing truly random number generators from physical processes, one is led to the question: How do we know which processes in nature are truly random, and how does one test physical systems for randomness? Microscopic processes which follow the laws of quantum mechanics, for example, are believed to be truly random. As a system evolves in time, the theory can only calculate the probabilities of what will happen. To verify this basic tenet of quantum mechanics, experiments are being performed to test the randomness of atomic decay.[5] The time it takes for an isolated atom to decay is measured many times in succession. Techniques from cryptography are used to test for any patterns in the decay times. At present, no patterns have been seen. In this article, we discuss a similar experiment, appropriate for the undergraduate physics laboratory, which can be used to generate true random numbers as well as to introduce students to the methods of testing for randomness. We measure the successive decay times of a sample of radioactive nuclei. The data are used in two ways: to generate true random numbers, and to test the degree to which the quantum mechanical decay process is truly random. Note that since the sample size is slowly decreasing in time, the results are not as "pure" as in the atomic experiment. As discussed in the text, the difference is calculable and is of the order of 1 part in 10^{12}.

Random processes, like chaotic processes, are interdisciplinary. We consulted faculty in the computer science and mathematics departments to determine an appropriate test for randomness. This helped us come up with a simple test, which we call the bit test, suited for our applications. We were then led to apply the bit test to other systems. Here, we discuss an application from our biology department in which the time between heartbeats in humans is measured. In this article we describe different aspects of the experiment and analysis. We start by explaining the setup for the radioactive decay experiment in the first section. We then discuss how to generate truly random numbers and to throw the dice from the data. This is followed by a section on tests for randomness and a description of the bit test. We conclude with applications of the bit test for pseudo-random number generators and the time between heartbeats.

II. THE EXPERIMENTAL SETUP

The apparatus is designed to measure the time between detected radioactive decays. The experimental setup, shown in Fig. 1, consists of a Geiger counter, an LED emitter–detector pair, and a computer to collect and analyze the data. The Geiger counter setup is used to detect the radiation, and has an output jack for a speaker. In our experiment, we used a Thornton Decade Counter/Power Supply (DEC-102 and APS-101), and our source was 1-μCi ^{137}Cs. The speaker output from the Geiger counter is connected directly to the LED emitter, and the coupled detector is connected directly to a digital input on the computer. For this connection, one can use a digital-to-digital input on a data acquisition card, a game port, or a serial port.

The time between radioactive decays is measured via software and the computer processor clock in the following way. When a decay is detected in the Geiger counter, a voltage pulse is sent to the speaker output which is connected to the LED emitter. The pulse is strong enough to "light" the diode. This signal is then picked up by the detector which is connected directly to a port in the computer. The status of the detector is determined by sampling the computer port. A sample program, written in Pascal, to determine the time between pulses is shown in Appendix A. The program records n pulses, and stores the number of times the port is read while the detector is active in on[i] and inactive in off[i]. The time between pulse i and i+1 is the sum of on[i]

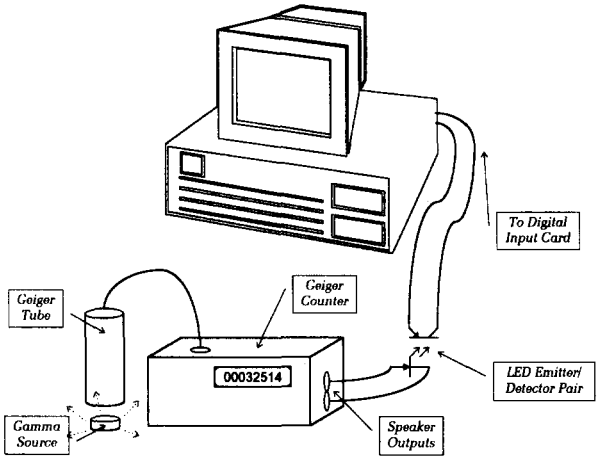

Fig. 1. A diagram of the experimental setup. The speaker output of the Geiger counter is connected directly to the LED infared emitter of an LED-detector pair. The output of the detector is connected directly to a digital input port in the computer. In our case it was connected to a digital-to-digital input on a general purpose acquisition card. The time between pulses is done by sampling the port via software.

and off[i]. By using software to measure the time between decays, the hardware problems of Refs. 1 and 2 are eliminated.

This simple setup has a number of virtues. Since most physics departments have Geiger counters and computers, the experiment can be set up with little extra expense. The LED emitter/detector pair cost around $1 and a serial card or game card runs around $10. The LED emitter/detector pair allows the computer to be isolated from the Geiger counter, which eliminates damage to the computer due to large pulses. This experiment is ideal for implementation as a special student project or in a laboratory class on computer interfacing.[6]

The advantage of using a digital input is that the port can be read fast, the speed being determined by the machine and bus speed. We used an IBM386 compatible with a clock speed of 20 MHz, and were able to sample the port at roughly 200 000 counts/s. At this speed the pulse width is around 80 counts. Placing our ^{137}Cs source up against the tube for maximum activity, the number of counts from the end of one pulse to the beginning of another averaged around 400. This is 480 counts between pulses, which enabled us to count roughly 400 pulses/s cleanly. With a more active source and faster equipment, the speed could of course be increased.

III. GENERATING RANDOM BITS

A simple way to generate random numbers is to compare the times between successive pulses from the Geiger counter. For example, consider measuring the times between three pulses. Let t_1 be the time between the first and second pulse, and t_2 the time between the second and third pulse. The generated bit is a 0 if $t_1>t_2$, and a 1 if $t_1<t_2$. The two times are discarded in the rare case that $t_1=t_2$. Based on the principles of quantum mechanics, radioactive decay is probabilistic by nature. The decay constant λ is the probability that one nucleus will decay between time t and $t+dt$. If one is observing A radioactive nuclei, the probability that any one of these will decay between time t and $t+dt$ we label as $P_A(t)dt$, and is given by

$$P_A(t)=A\lambda e^{-A\lambda t}. \tag{1}$$

Thus, the probability that t_2 is greater than t_1 is the product of $P_A(t)dt$ times $P_{A-1}(t')dt'$ integrated over all possible times t and t' such that $t'>t$:

$$\begin{aligned}\text{Prob}(t_2>t_1)&=\int_0^\infty P_A(t)\int_t^\infty P_{A-1}(t')dt'\,dt\\&=\int_0^\infty A\lambda e^{-A\lambda t}\int_t^\infty (A-1)\lambda e^{-(A-1)\lambda t'}\,dt'\,dt\\&=\frac{A}{2A-1}.\end{aligned} \tag{2}$$

Here, A is equal to the initial activity measured by the detector, A_0, divided by λ. Usually this number is very large. Using our 1-μCi source, A_0 was 20 000 counts/min when the source was up against the detector, and for ^{137}Cs the decay constant λ is 4.4×10^{-8} min^{-1}. So A is of the order 10^{12}, which makes the probability to obtain a bit value of one very close to 1/2, most likely even better than a real coin.

To a very good approximation (one part in 10^{12}) the probabilities $P_A(t)$ and $P_{A-1}(t)$ are equal. If they were exactly equal, then the probability of obtaining a one is exactly 1/2, *independent of the probability distribution*. This can be seen as follows. Let $P(t)dt$ be the probability that a decay will occur between time t and $t+dt$. Then the probability that $t_2>t_1$ is

$$\text{Prob}(t_2>t_1)=\int_0^\infty P(t)\int_t^\infty P(t')dt'\,dt. \tag{3}$$

Let $I(t)\equiv\int_t^\infty P(t')dt'$. Then we have $I(0)=1$, and $P(t)=-dI/dt$. Therefore, the probability that t_2 is greater than t_1 is

$$\begin{aligned}\text{Prob}(t_2>t_1)&=\int_0^\infty P(t)I(t)dt=\int_0^\infty\left(-\frac{dI}{dt}\right)I(t)dt\\&=-\frac{I(t)^2}{2}\Big|_0^\infty=\tfrac{1}{2}\end{aligned} \tag{4}$$

after integrating by parts. This simple result holds for any $P(t)$. Thus, even if $P(t)$ is not exactly exponential due to dead time or other experimental factors,[7] a "0" or a "1" will be generated with equal probability. The only effect that will upset the equal 0 or 1 probability is if $P(t)$ varies from one pulse to another. Such a situation might arise from a large background that varies rapidly in time. This unlikely condition can be eliminated by using proper shielding.

The above procedure can be used to simulate a coin toss or generate a random string of any number of bits. For example if one wants to produce a random number of N bits, $2N+1$ times are recorded. The first time is thrown out since it is not a complete timing from one pulse to another. This is actually not necessary for radioactive decay because of the "good-as-new" postulate.[8] However, to eliminate any measurement effects which might modify the "good-as-new" postulate we throw out the first timing. The other $2N$ times are paired off to make N pairs. For each pair, the above rule is used to produce the N random bits: "0" if $t_1>t_2$ and "1" if $t_2>t_1$.

Table I. The outcome of throwing "nature's dice" 120 000 times in succession. If the dice are fair, each number should occur 20 000 times on average with a standard deviation of 129. All numbers are within two standard deviations, and four of the numbers lie within one standard deviation of the average. The chi-square per data point is 0.90.

Dice number	Occurrances
1	19 997
2	19 865
3	20 025
4	19 930
5	19 925
6	20 247

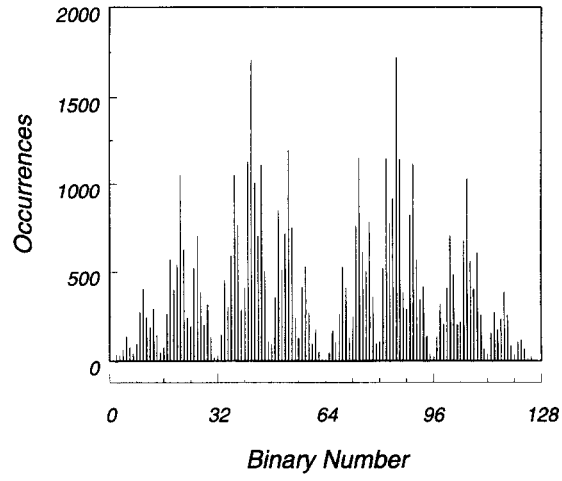

Fig. 2. A histogram of the 50 000 seven-bit binary numbers produced from the 400 000 experimental time measurements between radioactive decays. As discussed in the text, 50 000 groupings of eight consecutive times are used to obtain the 7 bit numbers. There are statistical fluctuations from the expected values of $50\,000*B_j^7$, and the histogram is consistent with that from a truly random process. The χ^2/D is 1.21.

IV. THROWING DICE

Another exercise for the students is to have them consider the following problem: Determine the most efficient way to use radioactive decay to simulate the throwing of a dice. The solution to this problem involves the permutation group and leads to a method for testing whether a sequence of numbers is random or not.

A simple way to throw the dice is to measure three consecutive times between pulses. Label the three times t_1, t_2, and t_3. The toss of the dice is determined as follows: If $t_1>t_2>t_3$ a 1, if $t_1>t_3>t_2$ a 2, if $t_2>t_3>t_1$ a 3, if $t_2>t_1>t_3$ a 4, if $t_3>t_1>t_2$ a 5, and if $t_3>t_2>t_1$ a 6 results. As discussed in Appendix B each of these time permutations occurs with equal probability. The results from our radioactive decay experiment are shown in Table I. Here we have thrown "nature's dice" 120 000 times in succession. For a purely random process, one should obtain each number 20 000 times on the average, with a standard deviation of $\sqrt{120\,000(1/6)(5/6)} = 129$ for a binomial distribution. From Table I we see that two thirds of the numbers are within one standard deviation of the average. The chi-square per data point, χ^2/D, for this experiment is

$$\frac{\chi^2}{D}=\frac{1}{6}\sum_{i=1}^{6}\left[\frac{N_i-20\,000}{129}\right]^2=0.90, \tag{5}$$

where N_i is the number of occurances for number i. It is not possible to prove that the probability is 1/6 for each number or that the process is even truly random. However, the experimental results suggest that nature is not throwing loaded dice.

Examining different permutation combinations is a method for testing if a sequence of pseudo-random numbers adequately simulates a truly random sequence.[9] We would like to test longer time sequences of the radioactive decay data; however, the number of permutations grows as $N!$ To make the analysis simple for larger N we use the "bit test," which we describe in the next section. The test is applied to the radioactive decay data, pseudo-random numbers, and the time interval between heart beats.

V. TESTS FOR RANDOMNESS IN PHYSICAL SYSTEMS

Techniques from computer science which test pseudo-random number generators can be used to examine physical systems for randomness. If pseudo-random numbers are to be used as random numbers in applications, they need to satisfy certain properties. A good discussion of these properties is given in Chap. 4 of Ref. 9, where a number of tests are described. Tests referred to as the permutation, gap, run, and poker tests, examine uniformity and correlations between the numbers being generated. In addition, other tests[10] such as a return map can be used to spot any pattern in the so-called random numbers. Some of these tests are applicable to physical systems. In this section we introduce a test for randomness, similar to the permutation and run test, which we call the bit test.

Consider the following method for generating a string of N bits from the radioactive decay experiment. First measure $N+1$ times between pulses. The ith bit of the string is "0" if $t_i>t_{i+1}$ and "1" if $t_i<t_{i+1}$. The N-bit number generated is random, but not with uniform probability (see Fig. 2). For example, if three times are measured a two bit number (b_1b_2) is generated. Each of the six permutations of the three times is equally likely. Thus the probability of obtaining 0 (00) is 1/6, of 1 (01) is 1/3, of 2(10) is 1/3, and of 3 (11) is 1/6. We denote the probability that the N-bit number has the value j as B_j^N. These probabilities are calculated in Appendix B, where we also list the results for $N=3$ and $N=4$ in Tables II(a) and II(b). In Fig. 2 we show the results of recording 8 pulse times from the radioactive decay experiment for 50 000 trials. We have binned the 400 000 measurements into their values between 0 and 127. The interesting property of these probabilities is that *the results are independent of the probability function P(t).*

One application of the above procedure is that it offers a quick and easy test for randomness which can be applied to a series of numbers or certain physical processes (see Appendix B). One takes groups of $N+1$ independent numbers (or measurements), and bins the N-bit numbers produced. If the series is truly random, then the binning will follow the probabilities listed in Appendix B. After M such binnings (of the N bit numbers), the integers j should occur $B_j^N M$ times with a standard deviation of $\sqrt{MB_j^N(1-B_j^N)}$. A chi-square test can be performed on the distribution of integers to test for randomness. We refer to this test as the bit test. This is not a new test for random numbers, but rather a subtest of the permutation test described in Ref. 9. In the permutation test,

Table II. Values of B_j^N for (a) $N=3$ and (b) $N=4$. Larger values of N are obtained from a simple computer program (Ref. 17).

(a)

j	B_j^3*24	j	B_j^3*24
0	1	4	3
1	3	5	5
2	5	6	3
3	3	7	1

(b)

j	B_j^4*120	j	B_j^4*120	j	B_j^4*120	j	B_j^4*120
0	1	4	9	8	4	12	6
1	4	5	16	9	11	13	9
2	9	6	11	10	16	14	4
3	6	7	4	11	9	15	1

strings of N independent numbers are chosen. These strings are then examined to see if each of the possible $N!$ permutations occurs with equal probability. In practice this is a difficult test to carry out if N is large, since the number of permutations becomes unmanageable. However, instead of examining the exact permutations, one can generate and bin the N bit number determined by the above rule. This is a lot quicker, and although is not as powerful a test as the permutation test, is still a strong test with the same features. Due to its simplicity larger values of N can quickly be tested for the permutation quality of their randomness. We demonstrate the usefulness of this test with an example of a pseudo-random number generator and physical processes.

A common algorithm for generating pseudo-random numbers is the linear congruential generator.[10] The $n+1$th number generated, x_{n+1}, is determined from the previous number, x_n, using the simple formula $x_{n+1}=(ax_n+b)$ mod m. The numbers generated are not truly random, and eventually the series repeats. For an appropriate bit length and enough data points, the sequence of numbers will eventually fail the bit test. For example, we choose $a=899$, $b=0$, and $m=32768$. The series repeats after about 4000 numbers.[11] For 7 bits (i.e., eight consecutive numbers) the χ^2/D is greater than 2 after 3000 samples. In Fig. 3(a) we graph the χ^2/D as a function of the number of strings sampled M. Since the numbers repeat, the χ^2/D eventually grows linearly with M. For comparison, we graph in Fig. 3(b) the χ^2/D for 6 and 8 bits. In both cases, the linear congruential generator fail the bit test. However, more trials are needed than with 7 bits. With 6 bits the string is not long enough, and for 8 bits more trials are needed to obtain enough data points for the χ^2 analysis. The bit test works best if the probability density function $P(x)$ is not uniform.

In Fig. 2 we plot a histogram of the seven-bit binary numbers produced from 400 000 experimental time measurements between radioactive decays. Here, 50 000 groupings of eight consecutive times are used to obtain the seven-bit numbers. The occurrences are close to the expected values of $50\,000*B_j^7$ with statistical fluctuations, and the numbers 0, 63, 64, and 127 rarely occur as expected. Note, to obtain the number 63 (or 64) the last seven times need to be sequentially increasing (or decreasing). The χ^2/D is 1.21 indicating the histogram is consistent with that from a truly random process. This is expected, since one believes that the process is governed by the laws of quantum mechanics and is a truly random process. Nonetheless, it is instructive for the students to observe and test the degree to which nature is random. For comparison, a return map of the data can also be plotted to test for randomness. In a return map, one plots t_{n+1} vs t_n to observe any patterns or strange attractors. A plot of a return map for the times between radioactive decays is shown in Fig. 4. On its own it is not very interesting. However, this exercise is a good complement[12] for the dripping faucet experiment, where the return map produces a strange attractor

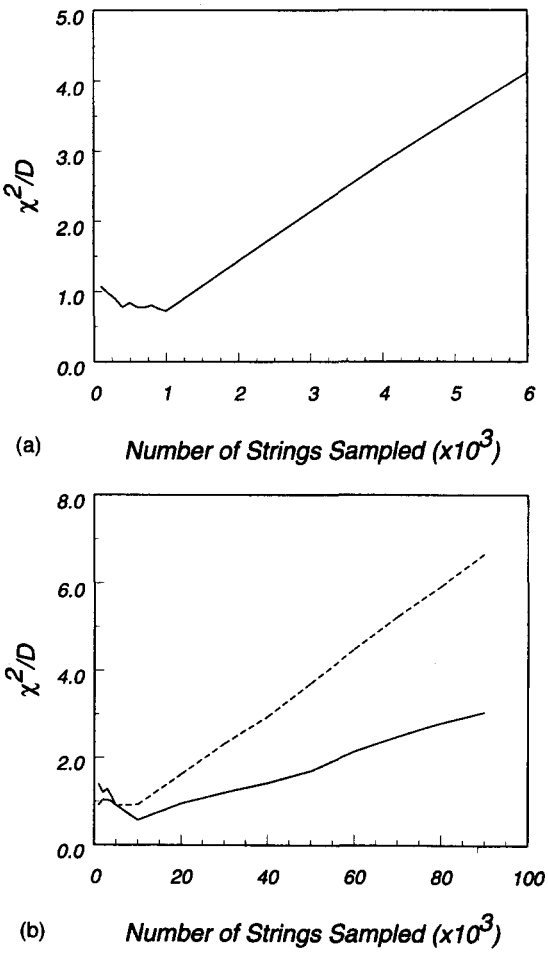

Fig. 3. χ^2/D vs the number of strings sampled M for the congruential random number generator. In (a), the string length is eight numbers producing a 7-bit number. In (b) the solid curve corresponds to a string length of 7 numbers (6 bits), and the dashed curve to a string length of nine numbers (8 bits).

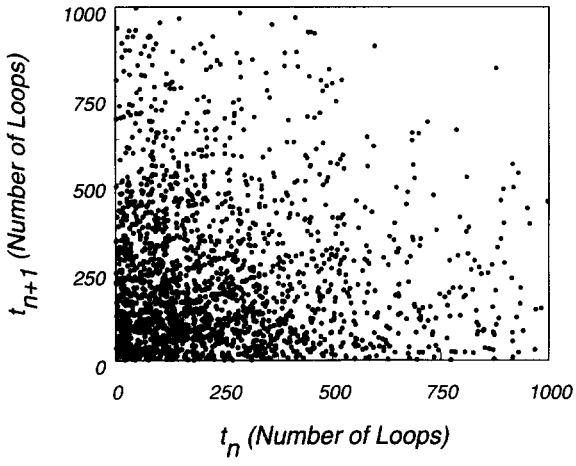

Fig. 4. A two-dimensional return map for the time between radioactive decays is plotted. The times, t_n, are the time from the end of one pulse to the start of the next pulse in terms of iteration loops in the computer program of Appendix A. Two thousand data points are used in the plot.

indicating that the times between drops are chaotic and not random.[13,14] No such pattern is seen in the nondescript blob of points in Fig. 4. One also sees no pattern in a three-dimensional return map, but higher dimensions are difficult to plot. The bit test allows one to examine longer strings of data points. It shows that for the number of bits (or dimensions) tested the data is random with respect to its permutation quality.

These methods are also applicable to the data from biological systems. The time between heart beats, as well as more sophisticated spectral analysis techniques, are being used as a diagnostic tool to investigate the condition of one's heart. To facilitate the analysis, a return map is often made in three dimensions of t_n, t_{n+1}, and t_{n+2}.[15] In Fig. 5 a return map using 1400 times from a volunteer is displayed. The subject is at rest, and the data form a locus of points along the diagonal $t_n=t_{n+1}=t_{n+2}$. From the return map alone, one cannot determine if the times are being generated randomly or not. Results of the bit test for three successive times are shown in Table III. The time between heartbeats clearly fails the test, with the χ^2/D being 38. From the table, it is seen that three consecutive increasing times ($j=0$) and three consecutive decreasing times ($j=3$) occur far more often than

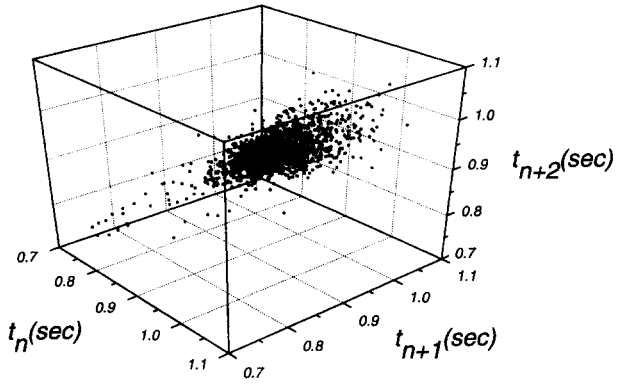

Fig. 5. A three-dimensional return map is plotted of the time between heartbeats, t_n, for a subject at rest. The axes correspond to t_n, t_{n+1}, and t_{n+2}, where n goes from 1 to 1398.

Table III. Results of the bit test applied to the time between heartbeats. If the times were truly random, the numbers in column two would be close to those of column three. There are too many sequentially increasing times ($j=0$) and sequentially decreasing time ($j=3$).

j	B_j^2*467	Heart data
0	78	116
1	156	98
2	156	107
3	78	146

the random case (the second column). This is because one's heart rate tends to speed up gradually and also slow down gradually due to breathing, etc. when one is at rest. Note that in this example both the three-dimensional return map and the 2-bit test examine three consecutive data points for patterns. The return map is useful in uncovering the functional dependence of a measurement with the previous two. The bit test is useful in testing if the sequential ordering of the measurements have the proper qualities for randomness. Using strings of three times, the bit test easily uncovers the nonrandom nature of the time between heartbeats. Perhaps the distribution of the binary numbers j can assist in determining if one's heart is healthy.

VI. SUMMARY

A simple experiment was described in which measurements of the time between radioactive decays enables one to produce random numbers. Successive times are compared and a 0 or a 1 is produced depending if $t_i>t_{i+1}$ or $t_i<t_{i+1}$. In one case, $2N+1$ pulses are recorded, and a random N-bit number is generated with uniform probability. In another case, $N+1$ pulses are recorded, and a random N-bit number is produced with a probability distribution related to the permutation group. The interesting aspect is that the numbers generated have the same probability distribution for any random process. One can use this fact to test for randomness. A simple test, called the bit test, was introduced to test the permutation quality of the randomness of a series of measurements. The radioactive decay process passed the bit test for randomness for bit lengths up to 7 bits.

These methods are well suited for an undergraduate physics class. The equipment is found in most physics stockrooms, and the project is a good exercise in computer interfacing with an experiment. The data analysis introduces the student to the different mathematical tests for randomness, the difference between pseudo-random numbers and truly random numbers, and enables the student to test the degree to which radioactive decay is random. One could also use the random numbers that are generated in a Monte Carlo calculation or other "random walk" applications in a computational physics class.[16] Here, truly random numbers might work better than pseudo-random ones. However, the most fascinating aspect of the exercise is to watch the random numbers appear on the computer screen, and realize that we are allowing nature to toss the dice.

ACKNOWLEDGMENTS

We would like to thank Mandayam Srinivas for many helpful discussions on pseudo-random number generators.

APPENDIX A: PASCAL PROGRAM TO READ DIGITAL PORT

The following is a simple program in Pascal which can be used to read the digital port and determine the time from one pulse to the next for eight consecutive pulses.

```
for i:=1 to 8 do
begin
  count:=0;
  while (port[308]<7) do count:=count+1;
  on[i]:=count;
  count:=0;
  while (port[308]>7) do count:=count+1;
  off[i]:=count;
end;
for i:=1 to 7 do times[i]:=on[i+1]+off[i+1];
bits;
```

Note that the first pulse reading is discarded, and the last seven times are stored into the array times[i]. For our acquisition card port 308 was 15 when the emitter was off, and 3 when it was on. It is important that each pulse after the first one is measured in the same manner. For each cycle after the first, the computer increments i, samples the port until it is greater than 7, stores the result in array on, samples the port until it is less than 7, and stores the result in off. Each cycle is treated the same. This will insure that $P(t)$ is the same for each pulse. It is better to start testing if the emitter is on, since the pulse width is consistent and about 80 counts wide. The procedure "bits" converts the array times into a 7 bit number as described in the text.

APPENDIX B: THE BIT TEST

Consider a series of $N+1$ real numbers (or measurements) x_i. Suppose that each one of these numbers is produced randomly with probability density $P(x_i)$ and that the probability does not depend on when it is produced, $P(x_i)=P(x)$ for all i. That is, the probability that any one of the numbers in the series lies between x and $x+dx$ is $P(x)dx$. We will also assume that $x_i \neq x_{i+1}$ for all i. In practice this is unlikely. With measurements, one can often increase the resolution of the measuring device to differentiate between the two numbers. Note that x_i can be equal to x_j for $j \neq i+1$.

From the $N+1$ real numbers (or measurements), an N-bit binary number is produced as follows. The binary number B has N bits $(b_1 b_2 b_3 \ldots b_N)$, where the i'th bit b_i is zero or one. The bit b_i is zero if $x_i > x_{i+1}$ and it is a one if $x_i < x_{i+1}$. The number B is between 0 and 2^N-1, however the distribution of B is not uniform. For example, it is very rare for B to be 0. This can only occur if the x_i are sequentially decreasing, $x_i > x_{i+1}$ for all i. This happens with a probability of $1/(N+1)!$. The same is true for B being equal to 2^N-1. In this case, the x_i must be sequentially increasing, and the probability for this to occur is also $1/(N+1)!$. We will label the probability that the N-bit binary number j occurs as B_j^N. We demonstrate this with a simple example of $N=2$.

Consider the case of $N=2$, with a series of three real numbers x_1, x_2, and x_3. The probability that $B=(11)=3$ is given by

$$B_3^2 = \int_0^\infty P(x) \int_x^\infty P(x') \int_{x'}^\infty P(x'')dx''\, dx'\, dx \qquad \text{(B1)}$$

This integral is easily solved by defining $I(x)=\int_x^\infty P(u)du$. Note that $I(0)=1$ and $dI/dx=-P(x)$. One can now integrate by parts:

$$\begin{aligned} B_3^2 &= -\int_0^\infty P(x) \int_x^\infty \frac{dI(x')}{dx'} I(x')dx'\, dx \\ &= -\int_0^\infty P(x) \frac{I^2(x')}{2}\bigg|_x^\infty dx = \frac{1}{2}\int_0^\infty P(x) I^2(x)dx \\ &= -\frac{1}{2}\int_0^\infty \frac{dI(x)}{dx} I^2(x)dx \\ &= -\frac{1}{2\cdot 3} I^3(x)\big|_0^\infty = \frac{1}{2\cdot 3} = \frac{1}{3!}, \end{aligned}$$

with the result independent of $P(x)$. Similar integrals can be solved to find the probability that B will be 0, 1, or 2. The integrands will be the same as Eq. (B1), but the limits will change. The limits will be from x to ∞ as above if $x_i < x_{i+1}$ or from 0 to x if $x_i > x_{i+1}$. In the latter case, one uses $(1-I(x))$ and integrates by parts. For example,

$$B_1^2 = \int_0^\infty P(x) \int_0^x P(x') \int_{x'}^\infty P(x'')dx''\, dx'\, dx = \frac{1}{3}.$$

The other results for the $N=2$ case are $B_0^2=1/6$ and $B_2^2=1/3$, independent of $P(x)$.

One can generalize the above procedure for any N, since the integrals can still be solved by integrating by parts. Some cases are simple: $B_0^N = B_{2^N-1}^N = 1/(N+1)!$, and $B_1^N = B_{2^N-2}^N = N/(N+1)!$. However, in general, the mathematics can be cumbersome. There is a simpler way to determine the probability that the binary number B will result from a series of random numbers. Consider the case $N=2$. If the three real numbers x_1, x_2, and x_3 are truly random and independent of each other, then they could have been produced in any order with equal probability. We just happened to measure one possibility. Thus, for $N=2$, we have $x_1>x_2>x_3$ giving $B=0$, $x_1>x_3>x_2$ giving $B=1$, $x_3>x_1>x_2$ giving $B=1$, $x_2>x_1>x_3$ giving $B=2$, $x_2>x_3>x_1$ giving $B=2$, and $x_3>x_2>x_1$ giving $B=3$, with each permutation having an equal probability of 1/6. However, $B=1$ and $B=2$ can occur in two different ways, so their probabilities are each 1/3. The same result was found using the integral method.

Examining the $(N+1)!$ permutations to determine the number of times the binary numbers occur is an easier way to obtain B_j^N than solving the integrals of Eq. (B1). A simple computer program[17] was written to compute the different probabilities. In Tables II(a) and II(b) we list results for $N=3$ and $N=4$. Note that the B_j^N have some symmetry properties: They are symmetric about $j=(2^N/2-1)$, they are left–right symmetric, and $B_j^N = B_{\bar{j}}^N$ where $\bar{j}$ is the complement of j $(\bar{j}=2^N-1-j)$.

Since the probabilities B_j^N are the same for all random processes [i.e., independent of $P(x)$], one can use this property to test for randomness. A simple and easy method is to perform a chi-square test on the data. For example, suppose we want to apply the test to strings of $N+1$. One picks M independent strings of $N+1$ data values, and calculates the N-bit number for each string using the rule given above. Let N_j denote the number of times that the N-bit binary number j occurs. On the average the number j will occur MB_j^N times

with a standard deviation of $\sqrt{MB_j^N(1-B_j^N)}$, if the data are truly random. Thus, the chi-square per data point, χ^2/D, defined by

$$\chi^2/D \equiv \frac{1}{2^N}\sum_{j=0}^{2^N-1}\frac{(N_j-MB_j^N)^2}{MB_j^N(1-B_j^N)} \tag{B2}$$

can be used to test for randomness. We refer to this test as the bit test. If χ^2/D is much greater than 2, then most likely the numbers in the series are not produced randomly. If, however, the χ^2/D is small (<2), one has shown that the numbers satisfy the permutation quality of randomness. They might still not be truly random. Note that the string of $N+1$ data values need not be consecutive, they just need to be independent. One could, for example, skip some values or rearrange the order.

For large values of N the bit test becomes unmanageable since there are 2^N values of j. We used $N=7$ for the radioactive decay data with no computational problems. As mentioned in the text, the bit test is not a new test for randomness. It is a subtest of the permutation test. However, it is much easier to apply than the permutation test, because each permutation is easily mapped to an integer. The number of integers grows as 2^N, whereas the number of permutations grows as $N!$. Although not as powerful as the permutation test, one can apply it to longer strings to check for the "permutation quality" of the numbers in the series.

[a]This project was done in partial fullfillment of the Bachelor of Science Degree.

[1]M. Gude, "Concept for a High Performance Random Number Generator Based on Physical Random Phenomena," Frequenz **39**, 187–190 (1985).
[2]H. F. Murray, "A general approach for generating natural random variables," IEEE Trans. Comput. **19**, 1210–1213 (1970).
[3]A. A. Berezin, "Isotopic jet as a perfect random number generator," Int. J. Electron. **63**, 673–675 (1987).
[4]E. S. Kelsey and S. H. Whitaker, "Binary random number generator using switching tree and wide-band noise source," U.S. Patent number 3423683, Jan. 21, 1969.
[5]See "Flipping a quantum mechanical coin," Sci. News **146**, 47 (July 16, 1994).
[6]C. A. Kocher, "A laboratory course in computer interfacing and instrumentation," Am. J. Phys. **60**, 246–251 (1992).
[7]F. Arquerros and J. Campos, "Simple apparatus to measure the temporal distribution between random events," Am. J. Phys. **46**, 191–192 (1978).
[8]Charles S. Barnett, "Probabilistic description of radioactivity based on the good-as-new postulate," Am. J. Phys. **47**, 173–177 (1979).
[9]Donald E. Knuth, *The Art of Computer Programming* (Addison–Wesley, Reading, MA, 1981), Vol. 2.
[10]Pierre L'Ecuyer, "Efficient and Portable Combined Random Number Generators," Communications of the ACM **31**, 742–749 (1988).
[11]Harvey Gould and Jan Tobochnik, *An Introduction to Computer Simulation Methods, Applications to Physical Systems* (Addison–Wesley, Reading, MA, 1988), Part 2.
[12]We thank Robert Alt for suggesting this idea to us.
[13]R. Shaw, *The Dripping Faucet as a Model Chaotic System* (Aerial, Santa Cruz, CA, 1984).
[14]K. Dreyer and F. R. Hickey, "The route to chaos in a dripping water faucet," Am. J. Phys. **59**, 619–627 (1991).
[15]G. E. Morfill and G. Schmidt, "Komplexitätsanalyse in der Kardiologie," Phys. Blätter **50**, 156–160 (1994).
[16]Charles A. Whitney, "Casino physics in the classroom," Am. J. Phys. **54**, 1079–1085 (1986).

[17]A computer program written in fortran which calculates the B_j^N can be obtained from the authors.

Thermally excited liquid surface waves and their study through the quasielastic scattering of light

W. M. Klipstein, J. S. Radnich, and S. K. Lamoreaux
Department of Physics, P.O. Box 351560, University of Washington, Seattle, Washington 98195-1560

(Received 10 August 1995; accepted 20 October 1995)

A simple apparatus employing a semiconductor diode laser and PIN photodiode has been used to investigate thermal capillary waves on liquid surfaces. These waves act as a weak, time-varying diffraction grating for the incident laser light; the diffracted light can be heterodyned with the light directly reflected from the liquid surface to extract fluid properties (surface tension and viscosity). In this paper we present a discussion of the phenomenon of surface waves and describe the construction of an apparatus to observe them. Results of measurements with this apparatus for the surface tension and viscosity of water with and without oil films and of benzyl alcohol at different temperatures demonstrate the effectiveness of the technique as well as environmental effects on liquid properties. This problem provides rich ground for a study of wave and thermal phenomena as well as an introduction to a variety of experimental techniques. © *1996 American Association of Physics Teachers.*

I. INTRODUCTION

When undisturbed, the free surface of a liquid gives an impression of perfect smoothness. However, due to the thermal energy contained in the myriad of surface (capillary) wave modes, the surface is perpetually fluctuating on a microscopic level (a sort of vertical displacement Brownian motion), giving the surface a time dependent roughness. These excited waves can be studied by scattering light from the surface, and liquid properties such as viscosity and surface tension can be determined with no significant direct contact with the liquid (compare, for example, Ref. 1). The