Sequential Partitioning

Mark F. Schilling

You have just agreed to repaint your parents' guest bedroom. In their garage sits a crusty old one gallon can of paint left over from the last time the room was painted. You try to pry the lid off with a screwdriver, loosening here and there at the edge of the lid, but the lid does not yield easily. The problem is a sticky one.

Soon you begin to wonder just how many places around the rim you will have to pry before the lid can be removed, and what pattern will produce the desired result most quickly. If it were known in advance exactly how many pryings would be required, it would clearly be best to pry at points equally spaced around the lid's circumference, knowing that the last of these actions would free the lid. Unfortunately, you are not able to anticipate this number so the above strategy cannot be used. The next best procedure would be one in which the prying locations are as evenly spaced as possible around the lid's rim for every potential stopping point of the process—but how to accomplish this?

In order to mount an analytical attack on this problem, it is necessary to adopt a criterion by which to gauge the degree of evenness of spacing that a collection of points scattered around a circle possesses. Clearly there are several possible measures of evenness which could be used. The standard we shall primarily use here is the size of the maximum gap (arc length) between any two consecutive points, not only due to its simplicity, but because if the largest spacing can be kept sufficiently small, this will necessarily impose considerable evenness among the other spacings as well. (It should also be noted that when one is opening a sticky can of paint, the size of the largest unloosened arc will probably be the primary determinant of whether the lid will be removable—and also the main factor in the tendency of the can to be secure from leaks when it is hammered shut after use.)

To bring the above problem into a more formal setting, consider a circle of circumference 1 obtained by joining together the ends of the interval $[0, 1]$. In the discussion and figures below, this circle will be traced out in a counterclockwise direction with 0 and 1 meeting at the top of the circle. A *cutting sequence* will refer to an infinite sequence of distinct points selected on this circle; any individual point belonging to this sequence will be termed a *cut*, inasmuch as (except for the first point) it subdivides an existing arc into two subarcs, thereby increasing the total number of arcs by one. We may assume that the location of the first cut is at $0 = 1$, thereby returning the circle to the original unit interval. Hence the problem under consideration is really one of sequentially partioning an interval evenly in the sense described above; however, there are advantages to working on a circle which we will see later.

The goal which we wish to achieve can be loosely stated as:

Keep the largest gap small at all stages of the cutting process. (1)

Thus we are taking a minimax-type approach to the cutting problem—we want to have a "good" partition regardless of when the process is terminated.

Clearly there are conflicts in trying to accomplish this goal. For instance, making the second cut at $1/2$, which is best for stopping after two cuts, offers the worst possible prospects for the maximum gap which will exist after *three* cuts, among all choices of the second cut. Thus sacrifices at particular stages are necessary in order to achieve consistently good performance.

LEAPFROG SEQUENCES. In a search for an optimal cutting scheme, a natural first step is to consider the order in which the cuts and the resulting intervals should be generated. Two rules can be developed: (i) It seems reasonable that each cut should be made in (one of) the largest existing interval(s) present at that stage. Only in this way can the size of the largest gap be reduced as soon as possible—in one step, unless there is a tie for the largest interval size. (ii) Secondly, suppose that n cuts have been made so far, and let the size of the *smallest* interval be S. Then the next $n - 1$ cuts can at best produce a partition in which the new *largest* interval has size S, and the only way that this can happen is if every interval which is larger than S after n cuts is divided into two subintervals each no larger than S.

Taking the above two considerations together yields the following paradigm: Each cut should divide any largest existing interval into two subintervals both no larger than the smallest existing interval. A great benefit of this paradigm is that the gaps generated by such a cutting sequence can be described by a single ordered sequence. Let x_1 represent the size of the initial interval, obtained by cutting at 0; thus $x_1 = 1$. Label the interval sizes obtained from the second cut as x_2 and x_3 with $x_2 \geqslant x_3$; thus $x_1 = x_2 + x_3$. The third cut divides the interval of length x_2 into subintervals of lengths x_4 and x_5, where we shall take $x_4 \geqslant x_5$; we then have three intervals having lengths $x_3 \geqslant x_4 \geqslant x_5$ satisfying $x_3 + x_4 + x_5 = 1$. Continuing in this way, the collection of intervals generated by such a cutting sequence satisfies the following three conditions:

$$x_1 = 1;$$
$$x_n = x_{2n} + x_{2n+1}, \qquad n = 1,2,3,\dots;$$
$$x_1 \geqslant x_2 \geqslant x_3 \geqslant \cdots.$$

Any sequence satisfying the above three conditions shall be referred to as a *leapfrog sequence*. Note that after any number n of cuts there will be intervals with lengths $x_n \geqslant x_{n+1} \geqslant \cdots \geqslant x_{2n-1}$ summing to 1; the $(n + 1)$-st cut then causes the leftmost term, x_n, to "leapfrog" over the other interval sizes to form two new terms on the right. Our goal is to keep the maximum gap size x_n 'small' for all n.

There are an infinite number of leapfrog sequences. A simple case is the one which follows the rule of always bisecting a largest existing interval; this produces the sequence $\{x_n\} = \{1, 1/2, 1/2, 1/4, 1/4, 1/4, 1/4, 1/8, \dots\}$. Figure 1 shows the

<table>
<tr><th>Cut #</th><th colspan="6">Intervals</th></tr>
<tr><td>1</td><td colspan="6">x_1</td></tr>
<tr><td>2</td><td colspan="3">x_2</td><td colspan="3">x_3</td></tr>
<tr><td>3</td><td colspan="2">x_4</td><td>x_5</td><td colspan="3">x_3</td></tr>
<tr><td>4</td><td colspan="2">x_4</td><td>x_5</td><td colspan="2">x_6</td><td>x_7</td></tr>
<tr><td>5</td><td>x_8</td><td>x_9</td><td>x_5</td><td colspan="2">x_6</td><td>x_7</td></tr>
</table>

Figure 1

relationship of a leapfrog sequence x_n to the corresponding partitioning of the unit interval created by the first cut. The particular interval sizes shown represent the initial stages of the bisecting sequence.

THE OPTIMAL LEAPFROG SEQUENCE. It is easy to show that any leapfrog sequence tends to zero at the rate of $1/n$. Clearly $x_n \geqslant 1/n$ for each n, with equality possible only if the gap sizes are all equal for some particular n, as in the bisecting sequence above for $n = 1, 2, 4, 8, \ldots$. To obtain an upper bound on x_n note that $1 = x_n + x_{n+1} + \cdots + x_{2n-1} \geqslant nx_{2n}$ since $\{x_n\}$ is nonincreasing, hence $x_{2n} \leqslant 1/n$, i.e., $x_n \leqslant 2/n$ for n even; a similar argument justifies the same bound for odd values of n as well.

Let us therefore study the behavior of the *normalized* maximum gap $M_n = nx_n$, which is of stable order and remains between the values 1 and 2 for all n for any leapfrog sequence. A refined version of the objective given in (1) concerning the long run behavior of the maximum gap can now be formulated:

$$\textit{Find } \{x_n\} \textit{ such that } L = \limsup_{n\to\infty} M_n \textit{ is minimized}. \tag{2}$$

For the bisecting sequence described above,

$$\{M_n\} = \{1, 1, 3/2, 1, 5/4, 3/2, 7/4, 1, \ldots\}.$$

Figure 2 shows the graph of $\{M_n\}$ for this sequence. When n is any power of 2, all gaps are of equal size and $M_n = 1$, the lowest possible value. However, for intermediate values of n the bisecting sequence can do very poorly indeed. In fact, this cutting strategy exhibits the worst possible value of L, 2, among all leapfrog sequences (recall that $x_n \leqslant 2/n$ for all n).

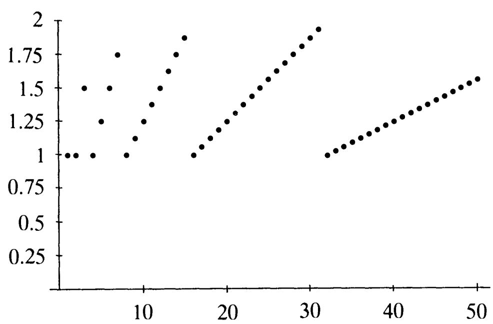

Figure 2

It seems likely that for an optimal leapfrog cutting sequence, the graph of M_n would not contain peaks and valleys such as those in Figure 2. Is it possible, then, to find leapfrog sequences for which M_n possesses a *limit*, and will this lead to a solution to (2)?

To answer these questions, the following ingredients are needed. First, note that since the x_n's are nonincreasing, $x_n = x_{2n} + x_{2n+1} \leqslant 2x_{2n}$, thus $M_n \leqslant M_{2n}$ for all n. Hence $L \geqslant M_n$ for all n since every M_n is a member of a nondecreasing infinite subsequence of M_n's.

Now let $S_n = L/n + L/(n+1) + \cdots + L/(2n-1)$. From the result just shown we have that for each n, $S_n \geqslant x_n + x_{n+1} + \cdots + x_{2n-1} = 1$. Furthermore, comparing the partial harmonic series S_n/L to $\int(1/x)\,dx$ shows that S_n approaches the limit $L \ln 2$ from above. Thus the best value of L that can be hoped for is $L = 1/\ln 2 \approx 1.44$.

This result shows that the minimum price which must be paid to achieve optimality in the sense of (2) is a 44% increase in maximum gap size (for large n) compared with the equal-spaced design which would be used if the total number of cuts to be made was specified in advance. It remains to show that a leapfrog sequence $\{x_n\}$ achieving this value of L exists.

To this end, define $y_n = \Sigma_{i=1}^{n-1} x_i$ for $n = 2, 3, \ldots$. For $n = 2^k$ we have

$$\begin{aligned} y_n &= x_1 + (x_2 + x_3) + (x_4 + x_5 + x_6 + x_7) + \cdots + (x_{2^{k-1}} + \cdots + x_{2^k - 1}) \\ &= k = \log_2 n. \end{aligned}$$

This suggests that to obtain a cutting sequence whose gap sizes decrease smoothly, we could set $y_n = \log_2 n$ for *all* n to determine values of x_n from the relationship $x_n = y_{n+1} - y_n$; we obtain from this the sequence

$$x_n = \log_2((n+1)/n), \qquad n = 1, 2, \ldots .$$

To check that $\{x_n\}$ is in fact a leapfrog sequence, note that

$$\begin{aligned} x_{2n} + x_{2n+1} &= \log_2\left(\frac{2n+1}{2n}\right) + \log_2\left(\frac{2n+2}{2n+1}\right) \\ &= \log_2\left(\frac{2n+2}{2n}\right) = \log_2\left(\frac{n+1}{n}\right) = x_n; \end{aligned}$$

the other two conditions for a leapfrog sequence are apparent at once. We shall refer to this sequence as the *logarithmic cutting sequence*. It is easy to see that for this sequence, M_n possesses a limit and that that limit is indeed $1/\ln 2$, hence the logarithmic cutting sequence is asymptotically optimal.

The graph of $\{M_n\}$ for the logarithmic cutting sequence is shown in Figure 3. Note that the curve approaches its limit from below, hence we have obtained as a bonus that the sequence performs particularly well for small values of n. Rather remarkably, although there are many partitioning schemes that yield a smaller maximum gap size for *some* values of n (such as the bisecting sequence), only the logarithmic sequence achieves criterion (2):

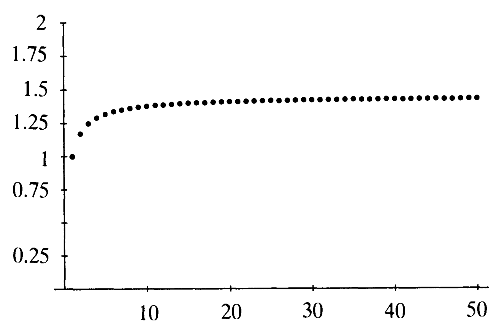

Figure 3

Theorem. *Let $\{x_n\}$ be the logarithmic sequence and let $\{z_n\}$ be any competing leapfrog sequence. Then $\limsup_{n\to\infty} nz_n > 1/\ln 2$.*

Proof: Write $z_n = x_n + \varepsilon_n$ for $n = 1, 2, 3, \ldots$ and let $\xi_n = n\varepsilon_n$. Since $nz_n = nx_n + \xi_n$, it suffices to show that $\limsup_{n\to\infty} \xi_n > 0$. Now the conditions of a leapfrog sequence give $\varepsilon_1 = 0$ and $\varepsilon_n = \varepsilon_{2n} + \varepsilon_{2n+1}$ for each n. Thus either all $\varepsilon_n = 0$ or some $\varepsilon_n > 0$. If a particular $\varepsilon_n > 0$ then $\max(\varepsilon_{2n}, \varepsilon_{2n+1}) > \varepsilon_n/2$, which immediately yields $\max(\xi_{2n}, \xi_{2n+1}) > \xi_n$. Since this argument can be repeated indefinitely, the theorem follows.

We have shown that the logarithmic cutting sequence $x_n = \log_2((n+1)/n)$ is the *unique* optimal leapfrog cutting sequence with respect to the minimax criterion (2).

THE DUAL PROBLEM. The criterion given in (1) and more precisely in (2) is of course not the only standard which could be used to measure the evenness of a sequential partitioning algorithm. One obvious alternative is to concentrate instead on the *smallest* interval which exists at each stage rather than the largest. This leads to the following dual to the objective given in (2):

$$\textit{Find } \{x_n\} \textit{ such that } 1 = \liminf_{n\to\infty} m_n \textit{ is maximized}, \tag{3}$$

where $m_n = nx_{2n-1}$ is the normalized smallest gap which exists after n cuts of a leapfrog sequence.

One might naturally conjecture that, since making the larger intervals smaller must make the smaller intervals bigger because the sum of all the interval lengths is constrained at 1, the logarithmic cutting sequence is again the unique optimal solution to this new criterion. The following theorem verifies that this is indeed the case:

Theorem. *The logarithmic cutting sequence is uniquely optimal with respect to criterion* (3), *achieving a value of* $l = (1/2)\ln 2$.

Proof: The relationship $x_{2n-1} = x_{4n-2} + x_{4n-1}$ yields $x_{2n-1} \geqslant 2x_{4n-1}$; multiplying by n then gives $m_n \geqslant m_{2n}$ for all n. Thus $m_n \geqslant l$ for all n. Now

$$\begin{aligned}
1 &= x_{2n-1} + \cdots + x_{4n-3} \\
&\geqslant x_{2n-1} + 2(x_{2n+1} + x_{2n+3} + \cdots + x_{4n-3}) \\
&= m_n/n + 2[m_{n+1}/(n+1) + m_{n+2}/(n+2) + \cdots + m_{2n-1}/(2n-1)] \\
&\geqslant 2l[1/(n+1) + 1/(n+2) + \cdots 1/(2n-1) + 1/2n] \\
&\geqslant 2l\int_{n+1}^{2n+1} dx/x = 2l\ln[(2n+1)/(n+1)].
\end{aligned}$$

Taking $n \to \infty$ establishes the claimed maximal value of l. It is again easy to show (by expanding the logarithm function) that the logarithmic cutting sequence achieves this value.

To prove uniqueness, let $\{z_n\}$ be any leapfrog sequence which achieves the optimal value $l = 1/2\ln 2$. First we show that the lim inf can be extended from the odd terms z_{2n-1} to the even terms z_{2n}: using the fact that $\{z_n\}$ is nonincreasing gives $\liminf_{n\to\infty}(n+1/2)z_{2n} \geqslant \liminf_{n\to\infty}(n+1/2)z_{2n+1} = \liminf_{n\to\infty}(n+1)z_{2n+1} = \liminf_{n\to\infty} nz_{2n-1} = 1/2\ln 2$. Combining the two cases then yields $\liminf_{n\to\infty}((n+1)/2)z_n = (1/2)\ln 2$, i.e., $\liminf_{n\to\infty} nz_n = 1/\ln 2$. Writing $z_n = x_n - \varepsilon_n$, $n = 1, 2, 3, \cdots$ and reasoning as in the previous theorem, it follows that ε_n must be 0 for all n so that once again the logarithmic sequence alone is optimal.

Just as the size of the normalized maximum gap M_n *increases* to its limit for the optimal cutting sequence, the size of the normalized minimum gap *decreases* to its limiting value as $n \to \infty$, so the logarithmic sequence is especially good for small values of n under both optimization criteria.

An Application to Data Analysis: Sunflower Plots. When a large amount of data on two variables is displayed in a scatterplot, frequently several data values will have the same position on the graph; this may cause a distorted impression of the data set to be rendered to the observer. Cleveland and McGill **[3]** introduced a graphical device they called *sunflowers* to display such multiple observation points. A sunflower is simply a collection of equal-length spokes each representing one observation, emanating from a common center which represents the variable values that these data share.

Normally the data are completely compiled before the scatterplot is made, in which case the spokes of each sunflower are spaced perfectly evenly, with angles of $2\pi/n$ between adjacent spokes. However, frequently situations occur in which the data is processed "on line," or a file is updated when new data becomes available. In these situations it is not possible to anticipate the spacing which the spokes should ultimately have. Clearly, to maximize the resolution of distinct observations at a common site (precisely the problem that motivated the idea of sunflowers in the first place), one would want to keep the minimal angle between any two spokes as large as possible. This is exactly criterion (3); hence the logarithmic cutting sequence is a most appropriate technique for such a situation. Figure 4 illustrates a scatterplot of data from Chambers, Cleveland, Kleiner and Tukey **[2]** in which the sunflowers have been constructed according to this paradigm. The sunflowers are no longer symmetric but the overall appearance of the plot is still very similar to the original (see **[2]**, p. 111). Even for values with as many as twelve coincident observations, the spokes are clearly resolvable.

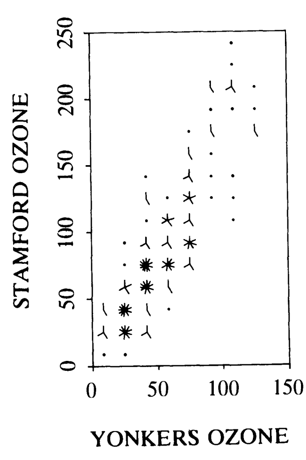

Figure 4

A SIMPLE IMPLEMENTATION OF THE OPTIMAL CUTTING PROCEDURE. The periodicity of the circle allows the logarithmic leapfrog sequence derived above to be implemented in a particularly straightforward manner: Beginning at the point labeled 0 and moving always in the same direction (clockwise or counterclockwise), cut whenever the total distance traveled (arc length) is a value of $\log_2 c$ for $c = 1, 2, 3, \ldots$. Figure 5 shows the locations of the cut points for counterclockwise winding. Note that whenever c is even, i.e., $c = 2k$ for integer k, the cut at that point will already have been made since $\log_2 c = \log_2 k + \log_2 2 = \log_2 k + 1$; thus these cuts can be eliminated. Every odd value of c on the other hand yields a new cut; taking $c = 2n + 1$ for integer n we have $\log_2 c = \log_2(n +$

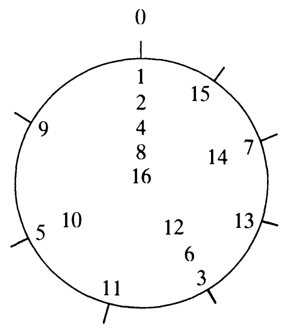

Figure 5

$1/2) + 1$, which shows that the cut divides the interval between the cuts made at $\log_2 n$ and $\log_2(n + 1)$. This interval, which has length $\log_2(n + 1) - \log_2 n = \log_2((n + 1)/n)$, is therefore divided into subintervals of lengths $\log_2(n + 1/2) - \log_2 n = \log_2((2n + 1)/2n)$ and $\log_2(n + 1) - \log_2(n + 1/2) = \log_2((2n + 2)/(2n + 1))$, which is precisely the requirement used above to derive the logarithmic leapfrog sequence.

To summarize this partitioning recipe, starting with a cut at 0, wind in a particular direction by cutting at each value of $\log_2 c$ for odd c. The procedure always jumps from the location of the cut just made, over the next existing cut, to the interior of the next interval which lies ahead in the direction of winding; this gives a second justification for the *leapfrog* adjective. One can easily imagine a machine programmed to carry out this operation very rapidly since no changes of direction are involved. Note that at any stage of the process, the intervals are ordered with respect to size.

OTHER PARTITIONING SCHEMES

Fixed Angle Cutting. Suppose that in the same fashion as described just above, we travel around the circle making each cut after a prescribed arc length has been traversed. Now suppose, however, that this arc length must be the same each time—what is possible in this case? Remarkably, it turns out that no matter what arc length (or equivalently, angle) is selected, there will never be more than three different gap sizes present at any given stage! (The reader may wish to experiment with this surprising result—angles that are simple fractions are the easiest computationally, however the statement holds for all irrational angles as well.) This result is known as the Three Gap Theorem; it was originally a conjecture of Steinhaus. A proof and references to this theorem may be found in a recent article by van Ravenstein **[8]**, where it is also shown that using an angle equal to the golden ratio $\phi = (\sqrt{5} - 1)/2 = .61803\ldots$ is optimal among the class of fixed angle partitioning strategies in the sense that the minimum over n of the ratio of smallest to largest gap sizes is maximized. The author notes that virtually all plants that produce leaves sequentially grow essentially according to this pattern in order to reduce leaf overlap.

Fixed angle cutting schemes are not leapfrog sequences; however it can be shown (see **[8]**) that each new cut in such a scheme divides a largest existing gap and produces one new gap whose size is equal to that of the smallest existing gap (the other new gap may be larger, however). The values of M and m for the golden ratio strategy are easily found to be $M = 1 + 2/\sqrt{5} = 1.89$ and $m = 1/\sqrt{5} = 0.45$. This represents significantly inferior results to those achieved for the optimal leapfrog sequence.

Random Partitioning. Suppose the partitioning is done by selecting each cutting point completely at random. How much worse is this method? Of course (with probability 1) we will not obtain a leapfrog sequence. It might be conjectured, however, that such a scheme will do fairly well asymptotically inasmuch as the cut points will eventually tend to become quite evenly spread out around the circle. In fact it can be shown that the maximum gap tends to shrink not at the rate of $1/n$ but at the somewhat slower rate of $\ln n/n$. Thus random cutting does infinitely worse asymptotically than any leapfrog sequence.

RELATED PHENOMENA

Benford's Law. There is an interesting connection between the optimal leapfrog sequence and the well known result known as Benford's Law [**1**], which concerns the remarkably consistent but non-uniform distribution of the decimal digits $1, 2, \dots, 9$ which is observed among the first significant digits of naturally occurring numbers, such as those in tables of physical constants, tables in almanacs, etc. Benford's Law states that each digit d occurs with frequency $\log_{10}((d+1)/d)$ for $d = 1, \dots, 9$.

Many arguments attempting to justify Benford's Law have been put forward since it first appeared in print in the paper of Newcomb [**4**], who preceded Benford by fifty-seven years; see [**7**] for a review. One of the most appealing of these, due to Pinkham [**5**], invokes the principle of invariance. The argument goes essentially as follows: Suppose that there is in fact such a law; i.e., each digit d occurs with frequency f_d throughout the great majority of natural tables. Then the law must certainly be independent of the units used; for example, the proportions of each digit's occurrence should not change measurably when a table using inches is retabulated in centimeters. The key observation is that regardless of the general magnitude of a number n, the first digit of n corresponds to a given range for the mantissa of the common logarithm of that number; for example, any number whose first digit is 1 has a mantissa between $\log_{10} 1 = .000$ and $\log_{10} 2 = .301$. Now rescaling the units by any factor F adds $\log_{10} F$ to the logarithms of each value in the table, which cycles the mantissas around by that amount (mod 1). In order for the distribution of first digits of a set of numbers, and hence the mantissas of their logs, to remain unchanged regardless of the scaling factor F, the distribution of the latter must be uniform. This directly yields Benford's Law.

Benford's Law extends easily to second and other leading digits in a straightforward way, using the uniform distribution of the mantissas. The law holds in any base b simply by using logs in that base. Thus, the optimal leapfrog sequence corresponds to Benford's Law for the distribution of leading digits in numbers represented in binary form. Of course for first digits alone, Benford's Law in this case is trivial—every number begins with 1. Suppose however that we look instead at the first k digits where k can be any positive integer. What proportion of naturally occurring numbers, then, have a binary representation which begins with a specific configuration of digits, say those which represent the integer n? Since only the mantissa of the base 2 logarithm of n is important, the answer is $\text{mantissa}(\log_2(n+1)) - \text{mantissa}(\log_2 n) \pmod 1 = \log_2(n+1) - \log_2 n =$

$\log_2((n + 1)/n)$, precisely the values generated by the optimal leapfrog cutting sequence.

Figure 6 is a base 2 version of Figure 5 which can be used to illustrate Benford's Law for binary numbers. Let the circle represent the mantissa scale for base 2 logarithms, winding counterclockwise from 0 to 1, which meet at the top of the circle. The circle is subdivided into eight intervals that correspond to the eight possible configurations of the first four digits of any binary number. A number whose base 2 mantissa falls in a given interval will have as its leading binary digits the string shown at the clockwise edge of the interval. Thus in a collection of binary data, we would expect more numbers to begin with 1000 than 1001, more to start with 1001 than 1010, and so forth, in proportion to the interval lengths shown in Figure 6. To obtain the relative frequencies of occurrence for binary numbers for which a smaller number of leading digits is specified, simply combine the appropriate neighboring intervals in Figure 6. To visualize Benford's Law as it applies to more than four leading digits, just continue the cutting process described for Figure 5. The law for a specific digit after the first can be illustrated by shading in alternating blocks of intervals beginning at the top of the circle. For instance, the frequency of 0 in the third digit of binary numbers according to Benford's Law is shown in Figure 6 by the total length of the arcs between 1000 and 1010 and between 1100 and 1110. Figure 6 can easily be modified to work for the original (base 10) version of Benford's Law or for any other base.

Circular Slide Rules. These now archaic devices have a close connection to the optimal leapfrog cutting sequence. Circular slide rules have two indicators similar to the hands of a clock, situated over a logarithmic scale which is wound several rotations and is numbered typically from 1 to 10. That is, if there are w windings to span the range 1 to 10, each revolution increases the value shown on the slide rule by $10^{1/w}$. The diagrams shown in Figures 5 and 6 have a logarithmic scale which increases by a factor of two per revolution.

A crude slide rule for binary calculations can be constructed by adding two hands to Figure 6 and converting these binary values to the range 1 to 2 by inserting "decimal points" (binary points?). To illustrate, consider the calculation of 9×5, which in base 2 is 1001×101. Placing the first hand of the slide rule at 1.001 and the second at 1.010, rotate the two hands together so that the first hand is now on 1.010 and read off the answer from the position of the second hand. If you try this on Figure 6 you will find that the result is located at a point just

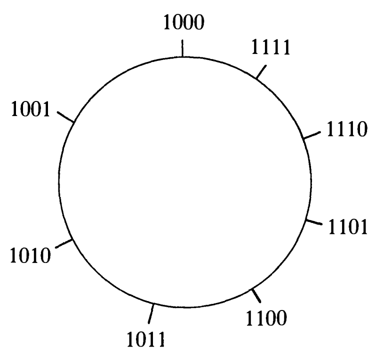

Figure 6

greater than (counterclockwise from) the cutting position 1.011 (labeled as 1011 in Figure 6). This yields the leading digits of the product, 45, which has a binary representation of 101101. Raimi [6] discusses the connection between Benford's Law and the circular slide rule for base 10 numbers in his article on the first digit problem.

ACKNOWLEDGMENT. The author is indebted to Ann Watkins for suggesting the application to sunflower plots.

REFERENCES

1. F. Benford, The law of anomalous numbers, *Proc. Amer. Phil. Soc.* 78 (1938), 551–572.
2. J. M. Chambers, W. S. Cleveland, B. Kleiner and P. A. Tukey, *Graphical Methods for Data Analysis*, Duxbury Press, Boston, MA, 1983.
3. W. S. Cleveland and R. McGill, The many faces of a scatterplot, *J. Amer. Statist. Assoc.* 79 (1984), 807–822.
4. S. Newcomb, Note on the frequency of use of the different digits in natural numbers, *Amer. J. Math.* 4 (1881), 39–40.
5. R. Pinkham, On the distribution of first significant digits, *Ann. Math. Statist.* 32 (1961), 1223–1230.
6. R. A. Raimi, The peculiar distribution of first digits, *Scientific American* 221 (6) (1969), 109–120.
7. R. A. Raimi, The first digit problem, *Amer. Math. Monthly* 83 (1976), 521–538.
8. T. van Ravenstein, Optimal spacing of points on a circle, *The Fibonacci Quarterly* 27 (1) (1989), 18–24.

Department of Mathematics
California State University
Northridge, CA 91330

> It is true that Fourier had the opinion that the principal object of mathematics was public use and the explanation of natural phenomena; but a philosopher like him ought to know that the sole object of the science is the honor of the human spirit and that under this view a problem of [the theory of] numbers is worth as much as a problem on the system of the world.
>
> —*C. Jacobi*