Pergamon

Comput. & Graphics, Vol. 18, No. 3, pp. 417–425, 1994

0097-8493/94 $6.00 + .00

0097-8493(94)E0020-X

Chaos and Graphics

AUTOMATIC GENERATION OF ITERATED FUNCTION SYSTEMS

J. C. SPROTT
Department of Physics, University of Wisconsin, Madison, WI 53706, USA

Abstract—A set of affine mappings with randomly chosen coefficients is repeatedly iterated numerically using the random iteration algorithm to produce an attractor with fractal characteristics. The attractor is tested for boundedness, sensitivity to initial conditions, and correlation dimension. In this way, a computer can generate a large collection of fractal patterns that are all different and most of which have considerable aesthetic appeal. A simple computer program and examples of its output are provided. Many of the attractors have been systematically evaluated for visual appeal, and a correlation is found with the Lyapunov exponent and correlation dimension.

1. INTRODUCTION

In an earlier companion paper, "Automatic Generation of Strange Attractors"[1], a method was described in which a system of coupled finite-difference equations with randomly chosen coefficients was iterated numerically and the solution tested for sensitivity to initial conditions (chaos). This method provides a powerful generator of new visual art forms and inspired a book that includes over 350 examples of such computer art[2]. This paper extends the technique to iterated function systems[3]. Such systems were originally studied by Hutchinson[4] and more recently applied to data compression and transmission by Barnsley[5, 6], who also introduced much of the terminology and the random iteration algorithm for their solution[7].

Whereas most previous work with iterated function systems has involved producing patterns to match some predetermined shape, this paper proposes a way for a computer to search a large class of potentially interesting mappings. The visual appeal of the patterns is shown to correlate with mathematical quantities that characterize them, such as the Lyapunov exponent[8] and correlation dimension[9], suggesting that it might be possible to refine further the automatic selection of patterns with strong visual appeal.

2. TWO-DIMENSIONAL AFFINE MAPS

The simplest example of an iterated function system is a set of two-dimensional affine maps:

$$x_{\text{new}} = a_1x + a_2y + a_5$$

$$y_{\text{new}} = a_3x + a_4y + a_6$$

Such a mapping has a single fixed point (x^*, y^*) given by

$$x^* = [-a_5(a_4 - 1) + a_2a_6]/[(a_1 - 1)(a_4 - 1) - a_2a_3]$$

$$y^* = [-a_6(a_1 - 1) + a_3a_5]/[(a_1 - 1)(a_4 - 1) - a_2a_3]$$

This fixed point may either be stable (attracting) or unstable (repelling). A stable fixed point attracts initial conditions within its basin of attraction, which in this case is the entire xy plane. With an unstable fixed point, successive iterates grow ever larger, and the system is unbounded. For simplicity, we consider saddle points (attracting in one direction and repelling in another) to be unstable, and we ignore periodic orbits, which seldom occur.

With each iteration, such an affine map takes a set of points in the xy plane and moves it to a new location in the plane, generally with scaling, translation, rotation, reflection, and shear. Stable solutions necessarily scale in such a way that the area of the new set is less than the area of the previous set, in which case we say that the set has contracted. Continually contracting mappings cause the set eventually to collapse into a region of negligible area surrounding the fixed point, and continually expanding mappings are unbounded. The amount of area contraction is determined by the magnitude of the determinant of the Jacobian matrix (hereafter simply "Jacobian") given by

$$J = |a_1a_4 - a_2a_3|,$$

which is the ratio of the area after a contraction to the area before. Thus the condition for contraction is $J < 1$. Note that area contraction does not guarantee boundedness, since a set can continually contract in one direction and expand in another, approaching a thin filament of zero area and infinite length. We are generalizing the usual contraction mapping in which every pair of points is moved closer together by the mapping.

Now suppose that there exists a second different affine map,

$$x_{\text{new}} = a_7x + a_8y + a_{11}$$

$$y_{\text{new}} = a_9x + a_{10}y + a_{12}$$

which is applied to the set after some finite number of iterations of the first map. The result is to displace the set of points away from the original fixed point toward which they were converging in the direction of a new fixed point in discrete steps of ever smaller size. However, the progression toward the new fixed point need not be along a straight line but rather can spiral around it. Thus two attracting fixed points successively turned on and off can compete for the set, producing a succession of points distributed over the *xy* plane in some pattern. Each point in the pattern can be transformed into another point in the pattern by some sequence of the two affine mappings. The collection of all such sequences is an iterated function system (IFS).

A practical method for producing iterated function systems is the random iteration algorithm in which a computer in essence repeatedly flips a coin and uses one map if it comes up heads and the other if it comes up tails. All possible sequences of heads and tails are eventually obtained, and long sequences of all heads or all tails, which would densely populate the regions near the fixed points, are rare. The locations of the fixed points are usually not obvious in such patterns. The patterns evolve much more quickly and uniformly if the coin is weighted so that the probability of applying each mapping is proportional to its Jacobian. The starting point can be chosen arbitrarily since the basin of attraction is the entire *xy* plane. However, the first few points should be discarded to ensure that the points have collapsed onto the attractor.

The resulting pattern is usually a deterministic fractal, despite being produced by a random algorithm. Different sequences of random numbers will produce the same eventual pattern. The pattern may have an integer dimension, but in such cases the boundary usually has a noninteger dimension. Although the attractor is a fractal, it is not usually called a strange attractor. That term is reserved for chaotic dynamical systems. Affine mappings cannot exhibit chaos because they lack the requisite nonlinearity.

3. SENSITIVITY TO INITIAL CONDITIONS

Like strange attractors, iterated function systems can be categorized by their sensitivity to initial conditions. Imagine two bounded iterated function systems produced by the same pair of affine maps but with initial conditions that differ slightly. If the two are produced by a different sequence of coin flips, the sequences of (x, y) values would bear no relation to one another. Indeed, this would be true even if they had the same initial condition. Thus there is extreme sensitivity to initial conditions resulting from the underlying randomness used to produce the sequence. However, if the same sequence of coin flips was used in the two cases, successive iterates would approach one another, implying insensitivity to initial conditions as expected for a deterministic nonchaotic process.

The difference between the two solutions decreases on average at an exponential rate. The rate of convergence is characterized by the Lyapunov exponent[8], which can be thought of as the power of 2 (or sometimes *e*) by which the separation increases on average for each iteration. Thus if the separation halves with each iteration, the Lyapunov exponent is −1 bit per iteration. The Lyapunov exponent can be thought of as the rate at which information about the initial condition is lost. A negative value means that information is gained; with each iteration we are better able to predict the result.

A two-dimensional IFS has two Lyapunov exponents since a cluster of nearby initial points may contract more in one direction than in another[10]. The least negative exponent is the one that dominates after a few iterations using the above procedure. The sum of the two Lyapunov exponents is related to the weighted average Jacobian $\langle J \rangle$ of the maps by

$$L_1 + L_2 = \log_2\langle J \rangle$$

One difficulty is that the two solutions eventually get very close together, and the computer cannot resolve the difference. This problem can be remedied if after each iteration the points are moved back to their original separation along the direction of the separation[11]. The Lyapunov exponent is then determined by the average of the distance they must be moved for each iteration to maintain a constant separation. If the two cases are separated by a distance d_n after the n'th iteration and the separation after the next iteration is d_{n+1}, the Lyapunov exponent is determined from

$$L = \frac{1}{N} \sum_{n=0}^{N-1} \log_2(d_{n+1}/d_n)$$

where N is the number of iterations. After each iteration, the value of one of the iterates is changed to make $d_{n+1} = d_n$. For the cases here, d_n is arbitrarily taken equal to 10^{-5}.

4. CORRELATION DIMENSION

The fractal patterns produced by iterated function systems arise from repeated affine mappings that make distorted copies of the pattern at successively smaller scales. Because the copies are generally sheared, they are not self-similar, but rather are self-affine. Furthermore, the mappings may overlap one another. Therefore, calculation of the fractal dimension directly from the equations is generally not straightforward.

It is relatively easy to calculate from the series of (x, y) values a related quantity, the correlation dimension[9]. It is defined as the logarithmic slope of the correlation integral $C(r)$, which is the probability that two randomly chosen points from the series are separated by a distance less than r, where

$$r = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

The correlation dimension is a lower bound on the fractal dimension, but in practice it closely approximates it. The two are identical if the points are uniformly distributed over the attractor.

The correlation dimension can be calculated in real time as the pattern develops by taking each new point and calculating its separation from one or more randomly chosen previous points[12]. A counter N_1 is incremented if the separation is less than r_1, and N_2 is incremented if it is less than r_2, where r_1 and r_2 are chosen arbitrarily except that they should be much smaller than the size of the attractor, but large enough that N_1 and N_2 are statistically significant and $r_1 < r_2$. In this case the correlation dimension is

$$F = \log(N_2/N_1)/\log(r_2/r_1)$$

with a statistical uncertainty of order $N_1^{-1/2}$.

5. COMPUTER SEARCH PROCEDURE

The procedure for implementing a computer search for interesting iterated function systems is straightforward. Choose the 12 coefficients a_1 through a_{12} randomly over some interval, choose initial conditions for x and y, and iterate the equations for the maps (choosing randomly which mapping to use at each iteration) while calculating the Lyapunov exponent and correlation dimension.

A computer program† that repetitively performs these operations is listed in the Appendix. It is written in a primitive version of BASIC so as to be widely accessible and easily understood. The program should run without modification under Microsoft BASICA, GW-BASIC, QBASIC, QuickBASIC, VisualBASIC for MS-DOS, and PowerBASIC, Inc., PowerBASIC on IBM PC or compatibles. It assumes VGA (640 × 480 pixel) graphics. If the hardware or BASIC compiler does not support this graphics mode, change the SCREEN 12 command in line 130 to a lower number (*i.e.*, SCREEN 2 for CGA mode). A compiled BASIC and a computer with a math coprocessor are strongly recommended.

The coefficients are chosen in increments of 0.1 over the range −1.2 to 1.2 (25 possible values) in line 320. Smaller coefficients produce too much contraction, and larger coefficients produce mostly unbounded solutions. The increment was chosen so that each pattern is visibly different and so that the coefficients can be coded into letters of the alphabet A through Y (A = −1.2, B = −1.1, *etc.*) for easy reference and replication. Thus each pattern is uniquely identified by a 12-letter name. The number of possible cases is thus 25^{12} or about 6×10^{16}. Viewing them all at a rate of one per second would require about 2 billion years! Thus it is very unlikely that any patterns produced by the program will ever have been seen before, and like snowflakes, nearly all of them are different.

Initial conditions are set arbitrarily to $x = y = 0.05$ in line 310. Other initial values produce the same result as expected for an attractor. Some computation time could be saved by choosing initial conditions at one of the fixed points. The Lyapunov exponent is calculated using an initial condition in which x is increased by 10^{-5} (line 310). The correlation dimension calculation begins after 200 iterations (line 910) and compares each new point with a randomly chosen previous reference point that is updated on average every 50 iterations (line 930). The correlation dimension is calculated at a scale of about 1% the largest dimension of the attractor with $r_2 = 10r_1$ (line 960 and 970). After 100 iterations, the program begins keeping track of the minimum and maximum values of x and y (lines 520–550) so that after 1000 iterations the screen can be cleared and resized to allow a 10% border around the attractor (line 560). After 1000 iterations the program begins testing the Lyapunov exponent and correlation dimension for a user-supplied criterion (line 650) for the case to be interesting. More will be said about such criteria in Section 7. If 21,000 iterations are reached with a bounded solution (line 630) and an acceptable combination of L and F, the result is assumed to be an interesting candidate IFS. The search immediately resumes after each case is confirmed and continues until a key is pressed (line 660).

The search procedure is surprisingly fast. On a 33-MHz, 80486DX-computer running PowerBASIC 3.0, the program finds about 600 interesting cases per hour. The listing in the Appendix only displays the patterns on the screen. A more versatile program would call a subroutine from line 630 to print the patterns, perhaps after user-confirmation or evaluation, or would save the coded coefficients in a disk file for later analysis.

6. SAMPLE ITERATED FUNCTION SYSTEMS

Figure 1 shows samples of the shapes that arise from the solution of such two-dimensional iterated function systems. These cases were selected for their beauty and diversity from a much larger collection. However, they are not atypical, and there are many others that would have served equally well. It is remarkable that such a diversity of shapes comes from the same simple set of linear equations with only different numerical values of the coefficients.

The cases shown were produced on a laser printer with 300 dots per inch resolution on an 8.5 × 11-inch page after about 500,000 iterations. Of course, the program needs modification to output the plots to a printer at high resolution. However, satisfactory results can be obtained by any of the various utilities that allow one to print a screen image.

Also shown on each figure is the code name preceded by the letter *a* to denote a two-dimensional IFS with two affine maps, the Lyapunov exponent L (in bits per iteration), and the approximate correlation dimension F.

7. AESTHETIC EVALUATION

A collection of 7500 such patterns was systematically examined by the author. The evaluations were done by displaying previously stored but unseen cases sequentially on the computer screen without any indication of the quantities that characterize them. Each case was evaluated on a scale of one to five according to its aesthetic appeal. It only took a few seconds for

† An IBM DOS disk containing the BASIC source code in the Appendix, an executable version of the code, and a more versatile menu-driven strange attractor program with 3-D glasses are available for $30 from the author. Specify 3.5 or 5.25-inch disk.

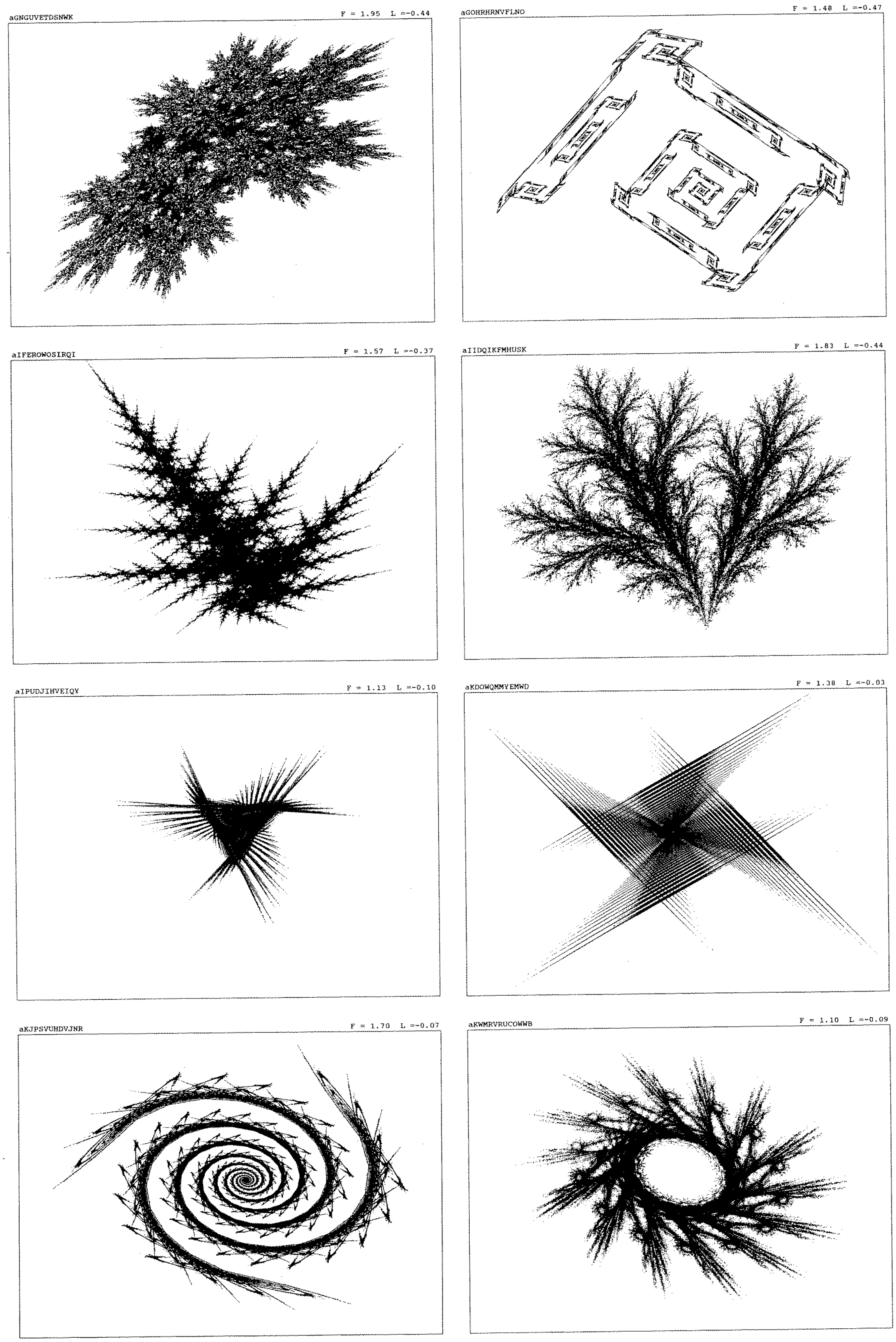

Fig. 1. Examples of iterated function systems produced by pairs of two-dimensional affine maps.

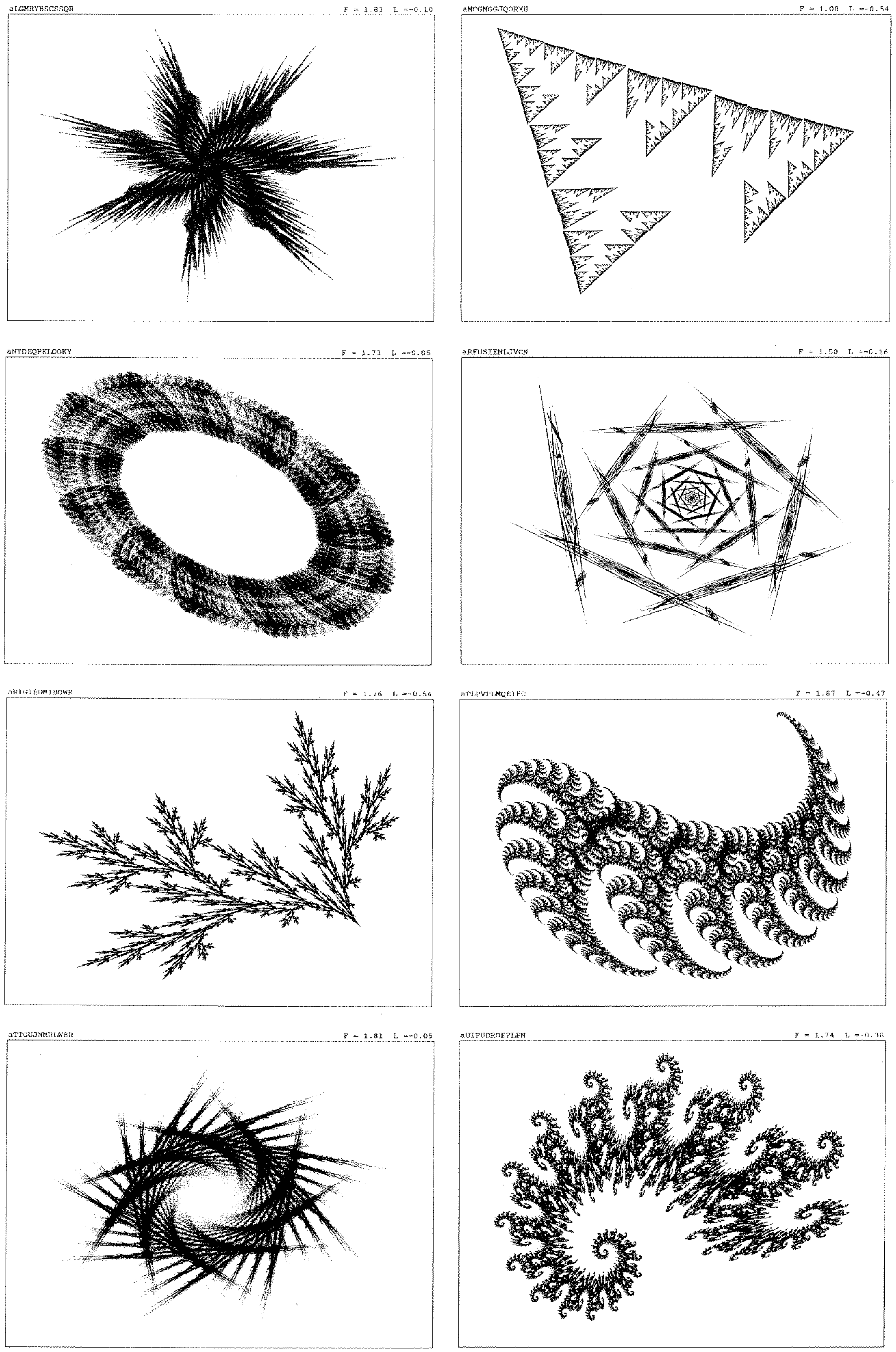
aLGMRYBSCSSQR
F = 1.83 L ≈ -0.10
aMCGMGGJQORXH
F = 1.08 L ≈ -0.54
aNYDEQPKLOOKY
F = 1.73 L ≈ -0.05
aRFUSIENLJVCN
F = 1.50 L ≈ -0.16
aRIGIEDMIBOWR
F = 1.76 L ≈ -0.54
aTLPVPLMQEIFC
F = 1.87 L ≈ -0.47
aTTGUJNMRLWBR
F = 1.81 L ≈ -0.05
aUIPUDROEPLPM
F = 1.74 L ≈ -0.38

Fig. 1. (Cont'd.)

each evaluation. Tests with other individuals show that the relative ratings are reasonably consistent.

Figure 2 shows the results using a gray scale on a plot in which the largest Lyapunov exponent (L) and correlation dimension (F) are the axes. The darkness of each box increases with the average rating of those attractors whose values of L and F fall within the box. There is a clear preference for patterns with correlation dimensions greater than 1 and large negative Lyapunov exponents. A similar result was found for strange attractors[1, 2], except that they are most appealing when their positive Lyapunov exponents are small.

The dimension preference is perhaps not surprising since many natural objects have dimensions in this range. The Lyapunov exponent preference is harder to understand. For a given dimension, the largest negative values of Lyapunov exponent correspond to cases in which the two exponents are equal, implying the same contraction in all directions and perfect self-similarity. The largest negative Lyapunov exponent and fractal dimension are bounded by a curve

$$-FL < \log O/\log D$$

where O is the number of mappings (2 in this case) and D is the dimension of the system of equations (2 in this case). Points on the boundary correspond to exact self-similarity. For the 76 cases that were rated five (best), the average correlation dimension was $F = 1.51 \pm 0.43$, and the average Lyapunov exponent was $L = -0.24 \pm 0.15$ bits per iteration, where the errors represent plus or minus one standard deviation. About 31% of the cases evaluated fall within the error bars.

These results suggest that the computer can be taught to select cases that have a high probability of aesthetic appeal. Various criteria have been used for this purpose. One such example is given in line 650 of the program listing in the Appendix. Another criterion that has been effective is

$$[(2 - F)/1.2]^2 + [(2 + L/\log O)/1.6]^2 < 1,$$

which eliminates about 98% of the two-dimensional cases. The 2% that remain are nearly all visually interesting.

8. SUGGESTIONS FOR FURTHER WORK

The method described above can be easily extended in a number of ways. There is nothing special about two-dimensional pairs of affine maps other than perhaps simplicity. This simplest case was chosen to emphasize the enormous diversity of patterns that can be produced. It is straightforward to apply the technique to cases with $O > 2$ and $D > 2$. The fraction of visually interesting cases decreases with both O and D unless some appropriate selection criterion is used. One could also examine the infinite variety of nonlinear mappings.

The probabilities of the mappings can be altered to change the density of points on various regions of the attractor. The probabilities could be included among the search parameters. Different probabilities do not alter the final appearance of the pattern, but they can significantly affect the early stages of its development. The maximum number of iterations is another parameter that can be adjusted. The development of the pattern is sensitive to the quality of the random numbers used to produce it[13], suggesting a possible test to distinguish chaos from randomness. Such tests have been applied to the analysis of DNA sequences[14].

Having found a visually appealing IFS, one can make small variations of the coefficients to optimize even further its appearance. The attractors can be animated by producing a succession of frames, each with a slightly different value of one or more of the parameters.

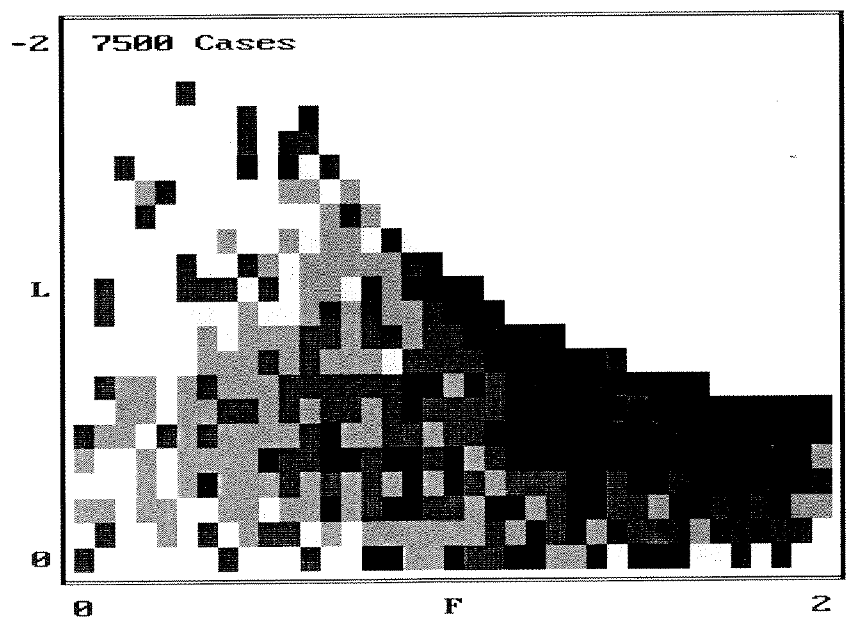

Fig. 2. Results of evaluating 7500 iterated function systems, showing that the most visually appealing cases are those with large negative Lyapunov exponents (L) and with correlation dimensions (F) greater than one.

Three-dimensional iterated function systems can be explored using mappings of the form

$$x_{new} = a_1x + a_2y + a_3z + a_{10}$$

$$y_{new} = a_4x + a_5y + a_6z + a_{11}$$

$$z_{new} = a_7x + a_8y + a_9z + a_{12}$$

with a Jacobian

$$J = |a_1a_5a_9 + a_2a_6a_7 + a_3a_4a_8 - a_3a_5a_7 - a_2a_4a_9 - a_1a_6a_8|$$

Adding a third dimension raises interesting possibilities for new display modes. The simplest case is to plot x and y, but to ignore z, which is equivalent to viewing the projection (or shadow) of the attractor on the xy plane. Alternately, the attractor can be projected onto the yz or zx plane or rotated through an arbitrary angle. A gray scale can be used to represent the number of iterates that fall on a given screen pixel[15], thereby restoring some of the resolution lost by the finite resolution of the screen.

Another possibility is to code the third dimension in color. Examples of three-dimensional iterated function systems using 16 colors are shown in Fig. 3. These

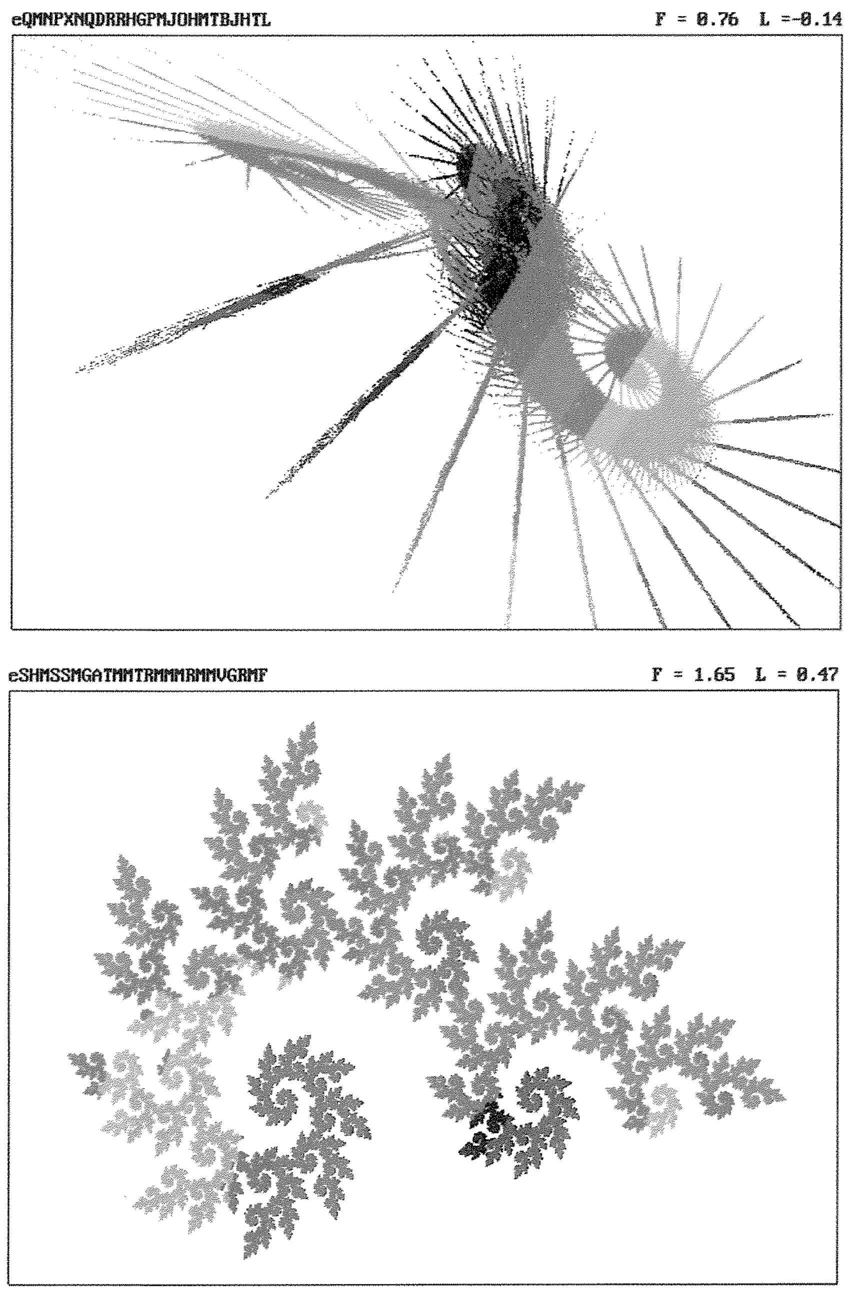

Fig. 3. Examples of iterated function systems produced by pairs of three-dimensional affine maps in which the color is determined by one of the variables.

figures were produced directly from VGA screen images using a color ink-jet printer. Some computer languages allow one to cycle through a variety of color palettes to find the most pleasing combination of colors or to produce an animated color display. Modern versions of BASIC have such a PALETTE command. In Fig. 3 the palette was adjusted to provide a rainbow sequence of colors.

It is also possible to produce an anaglyph[16] in which each (x, y) value is plotted twice, once in red and once in cyan, displaced horizontally by a distance proportional to z to produce a three-dimensional monochrome image when viewed through red/blue glasses. Color three-dimensional images can be produced by plotting the two colored views side-by-side and either viewing them cross-eyed or through an inexpensive prism stereoscope.† The attractors can be rotated to provide a view from the most pleasing angle or animated with successively rotated images.

Three-dimensional attractors can also be displayed using shadows to convey depth or contour bands as with topographic maps. One can slice the attractor like a loaf of bread and display the slices in an array or as an animated sequence. Combinations of these techniques permit visualization in dimensions higher than three. The attractors can be projected onto spheres, cylinders, tori, or other surfaces.

Iterated function systems can also be used to produce a crude kind of computer music. For a two-dimensional map, x might be used to control the pitch and y the duration of each note. The result is a not-displeasing, though alien-sounding, form of music that might appeal to those with exotic musical tastes.

Much more could be done with correlating the aesthetic appeal of the attractors with the various numerical quantities that characterize them. The Lyapunov exponent and correlation dimension are only two such quantities; there are infinitely many others[17]. One could determine if there are discernible differences between the preferences of scientists and artists. Preliminary indications suggest that complexity might appeal more to artists than to scientists, who tend to see beauty in simplicity. There may be discernible cultural differences. One could determine whether the results are the same for more complicated systems of equations and for different methods of displaying the results, such as color versus monochrome.

If quantifiable measures of aesthetics can be deduced for patterns produced by a computer[18], then perhaps the same could be done for patterns produced by humans. Fractal compression schemes attempt to represent arbitrary images by their IFS parameters. From these parameters, many quantities such as the spectrum of contractions can be easily deduced. It would make an interesting study to determine whether there are correlations between these quantities and the artistic quality of the original image. Perhaps such schemes could be used to detect forgeries if there are discernible differences in the fractal characteristics of works by various artists.

† Stereoscopes and other 3-D supplies are available from Reel 3-D, P.O. Box 2368, Culver City, CA 90231.

REFERENCES

1. J. C. Sprott, Automatic generation of strange attractors. *Comp. & Graph.* **17**, 325–332 (1993).
2. J. C. Sprott, *Strange Attractors: Creating Patterns in Chaos*, M&T Books, New York (1993).
3. M. Barnsley and S. Demko, Iterated function systems and the global construction of fractals. *Proc. Royal Soc. London A* **399**, 243–275 (1985).
4. J. Hutchinson, Fractals and self similarity. *Indiana Univ. J. Math.* **30**, 713–747 (1981).
5. M. F. Barnsley, V. Ervin, D. Hardin, and J. Lancaster, Solution of an inverse problem for fractals and other sets. *Proceedings of the National Academy of Sciences* **83**, 1975 (1986).
6. M. F. Barnsley and L. P. Hurd, *Fractal Image Compression*, A. K. Peters, Wellesley, MA (1993).
7. M. F. Barnsley, *Fractals Everywhere*, Academic Press, Boston (1988).
8. H. G. Schuster, *Deterministic Chaos*, Springer-Verlag, New York (1984).
9. P. Grassberger and I. Procaccia, Characterization of strange attractors. *Phys. Rev. Lett.* **50**, 346–349 (1983).
10. G. Benettin, L. Galgani, A. Giorgilli, and J. Strelcyn, Lyapunov characteristic exponents for smooth dynamical systems and for Hamiltonian systems; a method for computing all of them. *Meccanica* **15**, 9–20 (1980).
11. A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, Determining Lyapunov exponents from a time series. *Physica* **16D**, 285–317 (1985).
12. W. Lauterborn and J. Holzfuss, Evidence for a low-dimensional strange attractor in acoustic turbulence. *Phys. Lett. A* **115**, 369–372 (1986).
13. I. N. Stewart, Order within the chaos game? *The Dynamics Newsletter* **3**, 4–9 (1989).
14. H. J. Jeffrey, Chaos game visualization of DNA sequences. *Comp. & Graph.* **16**, 25–33 (1992).
15. C. Pickover, A note on rendering 3-D strange-attractors. *Comp. & Graph.* **12**, 263–267 (1988).
16. J. C. Sprott, Simple programs produce 3D images. *Comp. Phys.* **6**, 132–138 (1992).
17. J. D. Farmer, E. Ott, and J. A. Yorke, The dimension of chaotic attractors, *Physica* **7D**, 153–170 (1983).
18. H. W. Franke, *Computer Graphics, Computer Art*, Springer-Verlag, New York (1985).

APPENDIX

```
110 DEFDBL A-Z: DIM A(12)
120 RANDOMIZE TIMER              'Reseed random number generator
130 SCREEN 12                    'Assume VGA graphics
140 GOSUB 300                    'Set parameters
150 GOSUB 400                    'Iterate equations
160 GOSUB 500                    'Display results
170 GOSUB 600                    'Test results
180 ON T% GOTO 130, 140, 150
190 END
```

```
300 REM Set parameters
310 X = .05: Y = .05: XE = X + .00001: YE = Y
320 FOR I% = 1 TO 12: A(I%) = .1 * (INT(25 * RND) - 12): NEXT I%
330   J0 = ABS(A(1) * A(4) - A(2) * A(3))
340   J1 = ABS(A(7) * A(10) - A(8) * A(9))
350 IF J0 + J1 = 0 OR J0 > 1 OR J1 > 1 THEN GOTO 320  'Not contracting
360 P = J0 / (J0 + J1)
370 T% = 3: LSUM = 0: N = 1: N1 = 0: N2 = 0
380 XMIN = 1000000#: XMAX = -XMIN: YMIN = XMIN: YMAX = XMAX
390 RETURN

400 REM Iterate equations
410 IF X < > XE THEN IF RND > P THEN R% = 6 ELSE R% = 0
420 XNEW = A(1 + R%) * X + A(2 + R%) * Y + A(5 + R%)
430 YNEW = A(3 + R%) * X + A(4 + R%) * Y + A(6 + R%)
490 RETURN

500 REM Display results
510 IF N < 100 OR N > 1000 THEN GOTO 560
520   IF X < XMIN THEN XMIN = X
530   IF X > XMAX THEN XMAX = X
540   IF Y < YMIN THEN YMIN = Y
550   IF Y > YMAX THEN YMAX = Y
560 IF N = 1000 THEN GOSUB 800
570 IF X > XL AND X < XH AND Y > YL AND Y < YH AND N > 1000 THEN PSET (X, Y)
590 RETURN

600 REM Test results
610 GOSUB 700                  'Calculate Lyapunov exponent
620 GOSUB 900                  'Calculate correlation dimension
630 IF N > 21000 THEN T% = 2   'Candidate IFS found
640 IF ABS(XNEW) + ABS(YNEW) > 1000000# THEN T% = 2  'Unbounded
650 IF N > 998 AND (F < 1 OR L > -.2) THEN T% = 2     'Uninteresting
660 IF LEN(INKEY$) THEN T% = 0 'User key press
670 X = XNEW: Y = YNEW: N = N + 1
690 RETURN

700 REM Calculate Lyapunov exponent
710 XSAVE = XNEW: YSAVE = YNEW: X = XE: Y = YE
720 GOSUB 400        'Reiterate equations
730 DLX = XNEW - XSAVE: DLY = YNEW - YSAVE: DL2 = DLX * DLX + DLY * DLY
740 DF = 10000000000# * DL2: RS = 1# / SQR(DF)
750 XE = XSAVE + RS * (XNEW - XSAVE): YE = YSAVE + RS * (YNEW - YSAVE)
760 XNEW = XSAVE: YNEW = YSAVE
770 LSUM = LSUM + LOG(DF): L = .721347 * LSUM / N
790 RETURN

800 REM Resize the screen (and discard the first thousand iterates)
810 DX = .1 * (XMAX - XMIN): DY = .1 * (YMAX - YMIN)
820 XL = XMIN - DX: XH = XMAX + DX: YL = YMIN - DY: YH = YMAX + DY
830 IF XH - XL < .000001 OR YH - YL < .000001 THEN GOTO 890
840   WINDOW (XL, YL)-(XH, YH): CLS
850   LINE (XL, YL)-(XH, YH), , B
890 RETURN

900 REM Calculate fractal dimension
910 IF N < 200 THEN GOTO 990      'Wait for transient to settle
920 IF N = 200 THEN D2MAX = (XMAX - XMIN) ^ 2 + (YMAX - YMIN) ^ 2
930 IF N = 200 OR RND < .02 THEN XS = X: YS = Y 'New reference point
940 DX = XNEW - XS: DY = YNEW - YS
950 D2 = DX * DX + DY * DY
960 IF D2 < .001 * D2MAX THEN N2 = N2 + 1
970 IF D2 < .00001 * D2MAX THEN N1 = N1 + 1: F = .434294 * LOG(N2 / N1)
990 RETURN
```