Statistics of Trials and Placements: Circles and Squares.

The Parameters *c* and *f*.

1. Introduction.

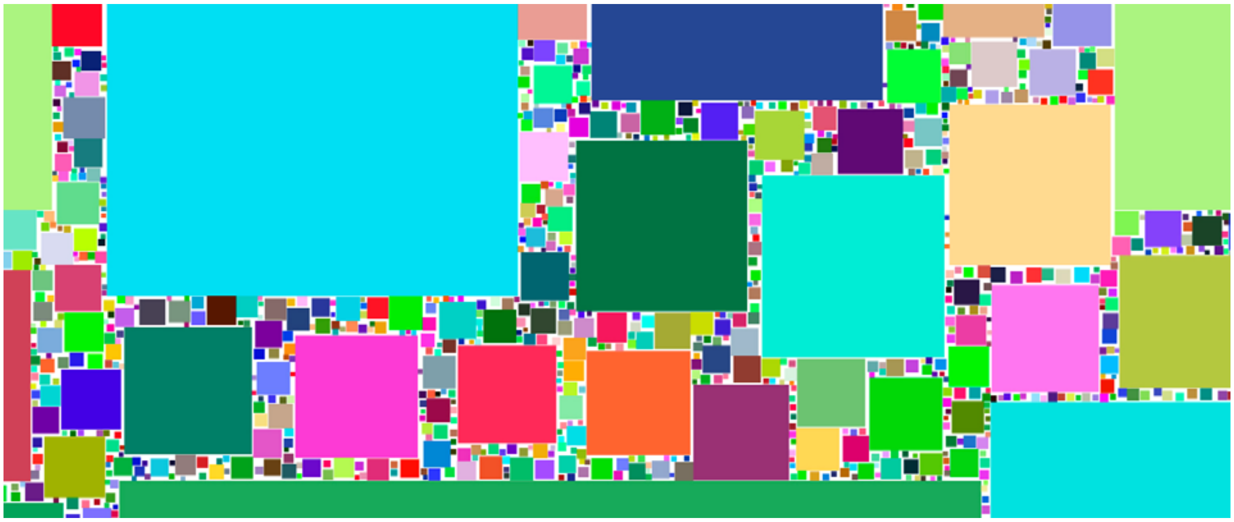

Fig. 1. Part of a square fractal. Random color.

A primary feature of the statistical geometry algorithm (Appendix A) is the power law for the areas of the fractalized shapes:

$$A_i = \frac{const.}{(i+N)^c} \qquad \text{Eq. (1)}$$

where A_i is the area of the the *i*-th shape (circles in the present case). The constant in the numerator is chosen so that the sum of all the areas A_i (to infinity) equals the total area *A* of the region where the shapes are to be placed. When the algorithm is run it is found that the cumulative number of trials n_{cum} needed to place *n* shapes is given by another power law:

$$n_{cum}(n) = Kn^f \qquad \text{Eq. (2)}$$

for large *n*. It should be kept in mind that Eq. (1) is a *definition* involved in setting up a run, while Eq. (2) is an observed result. The *f* parameter is found by fitting a power law to somewhat noisy data and is subject to the statistical variation involved in such situations, i.e., the *f* values given here are statistical estimates while *c* values are exact quantities, chosen in the setup procedure.

There is a relationship between *f* and *c* and an important goal here is to obtain data on this relationship for circles and squares.

2. The $n_{cum}(n)$ Plots

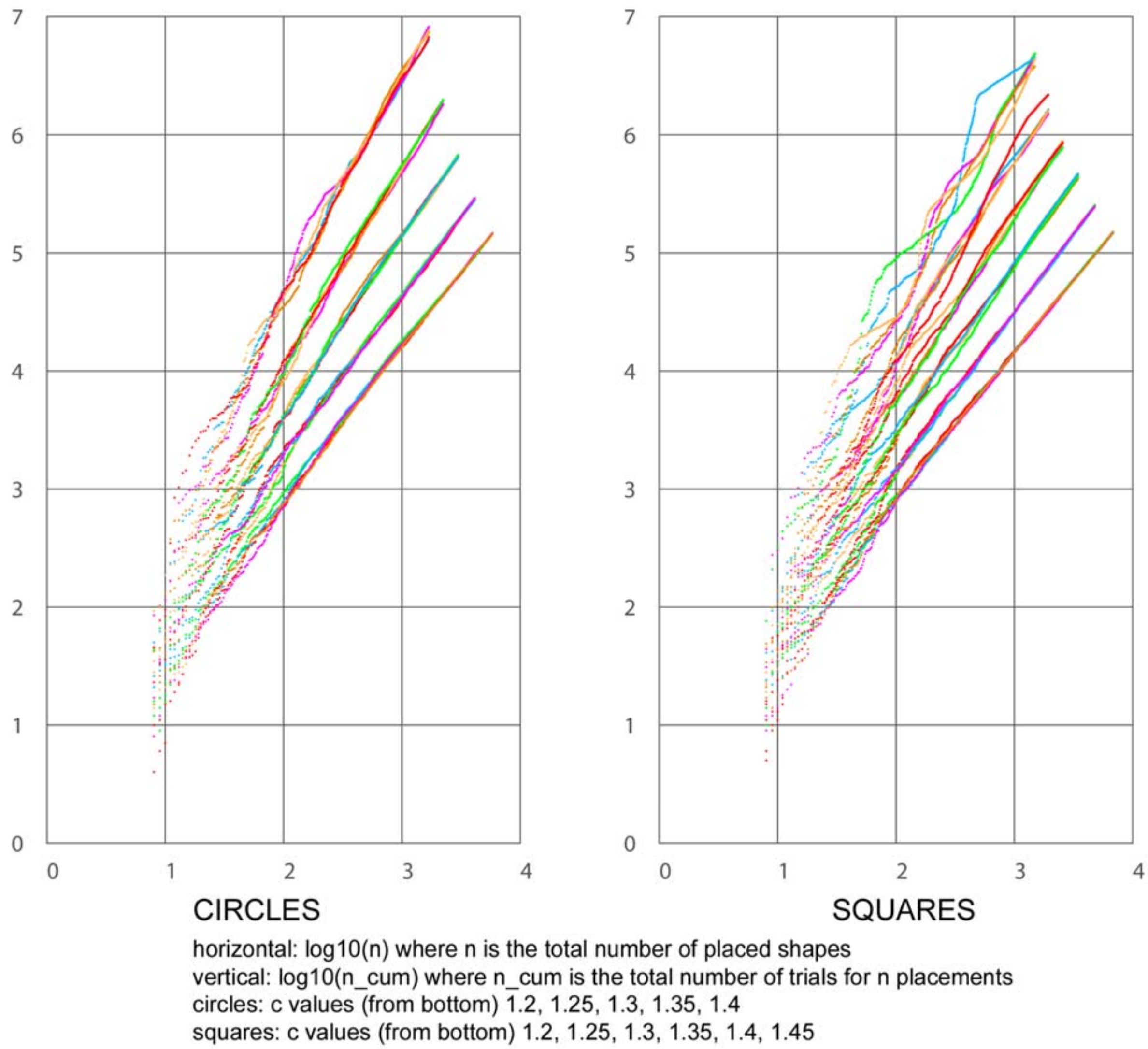

Fig. 2. Log-log run plots for statistical geometry fractals. There are 5 runs for each *c* value. Note the extremely wide range of the vertical scale (7 orders of magnitude).

A plot of the cumulative number of trials n_{cum} versus the number of placed shapes n is the best way to represent what happens at run time. Because of the large range of n_{cum} values and the fact that the large-n behavior of $n_{cum}(n)$ is a power law in n, the best way to present the data is in log-log coordinates, i.e., $\log_{10}(n_{cum}(n))$ versus $\log_{10}(n)$. In such a plot the slope of a straight line is the exponent of a power law.

Each such plot is a record of a particular run as a series of dots, and because of the random nature of the algorithm it will be different every time the algorithm is run. For large n the dots merge into a continuous line.

One of the most interesting questions that arises is "Does the statistical geometry algorithm halt?" These plots provide the best test of this, and given the power-law behavior for large *n* the evidence suggests that the algorithm does not halt[1] (at least not for "low" c values). The total number of trials needed for placement of *n* shapes is quite predictable.

Increasing *c* increases the number of trials needed to place *n* shapes by a huge amount.

There is a lot of noise in these traces due to the random nature of the placement algorithm. In general they become noisier for higher *c* and higher *n*. This graph can give the reader some idea of how much variation to expect in the execution of the algorithm. Some observations:

- For low *c* the relationship between $n_{cum}(n)$ and *n* goes over to a power law in *n* for large *n*.
- As *c* increases there is more noise in $n_{cum}(n)$, but the main trend is still a power law.
- For both circles and squares the $n_{cum}(n)$ behavior shows *oscillations* about the mean for large *c*. For the circles these oscillations are sinusoidal-looking. For squares one sees the rather surprising feature of sawtooth-like oscillations in some of the curves. The magnitude of such oscillations varies greatly from one run to another. It is possible that the upper limit of *c* values for which the algorithm works is set by the growth in these fluctuations.
- For *c* values higher than about 1.2, squares are much more "packable" than circles and require fewer trials under given conditions.
- The plots get much noisier for high *c* values, indicating that the amount of computation needed for a given *c* and *n* is much more variable.
- As a rule *a **given percentage fill** is most rapidly achieved with a high c value*. The many trials per placement is more than offset by the much faster increase in fill percentage.
- As a rule *a **given number of placed shapes** is most rapidly achieved with a low c value*. The percentage fill, however, will be low.

---

[1] What one means by "halt" depends on the assumptions made about the numbers used. For ordinary floating-point digital numbers the algorithm will eventually halt because of the finite precision of the numbers. (In practice it would take an immensely long time to actually reach this limit.) For a mathematician's numbers, which have infinite precision, the evidence is that the process does not halt, but the evidence of computational experiments falls far short of a rigorous proof. In all of the author's statistical geometry computations the algorithm is computed with standard-precision floating-point numbers . The rasterized images are created from the high-precision data thus found.

3. The Relationship Between *c* and *f*.

Values of *f* were found for each run trace. Since they vary, they were averaged to produce mean values as follows:

	circles	squares
c = 1.10	f = 1.11	
c = 1.15	f = 1.19	
c = 1.20	f = 1.27	f = 1.22
c = 1.25	f = 1.34	f = 1.29
c = 1.30	f = 1.49	f = 1.35
c = 1.35	f = 1.63	f = 1.46
c = 1.40	f = 1.86	f = 1.54
c = 1.45		f = 1.65
c = 1.47	f ~ 2.3	

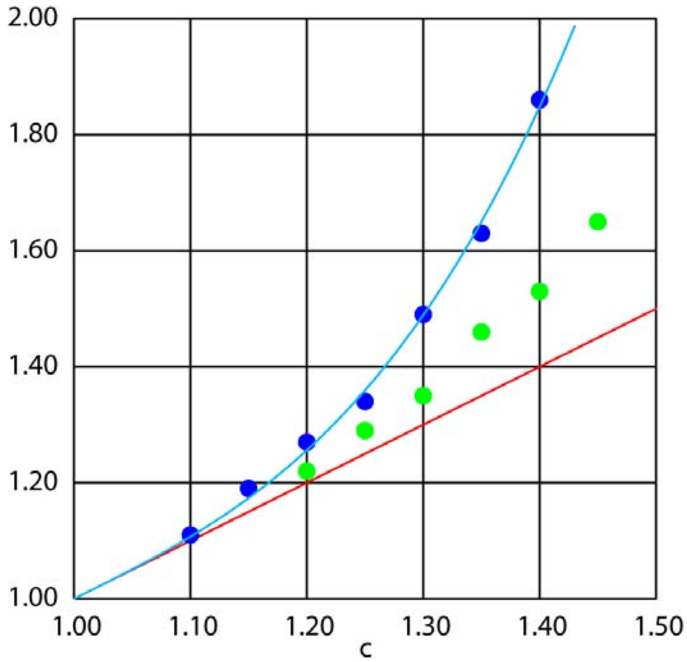

Fig. 3. The relationship between the power-law exponent c and the power-law exponent f (the power law for cumulative trials versus the number of placed shapes). The blue points are for circles and the green ones for squares. The red line is $f = c$. Note that $f > c$ for all data points. The blue line is an approximate curve-fit to the circle data, and is given by $f = c + 7(c\text{-}1)^3$.

It can be seen that f increases very steeply with c beyond about c = 1.2 which shows that the amount of computation for a given number of placed shapes increases very rapidly for high c values.

That data suggests that the upper limit of usable c values is higher for squares than for circle.

The data fitting scheme for finding *f* is described in Appendix B. The statistical error in *f* is about ±.02 to ±.04 (higher for higher *c*).

4. References.

[1] Statistical geometry article on the web site john-art.com.

[2] John Shier, "Hyperseeing", Summer 2011 issue, pp. 131-140, published by ISAMA. Available by download at the web site john-art.com.

[3] "Statistical Geometry", John Shier, July 2011. A colorful self-published fractal art picture book available at lulu.com.

John Shier

September 2011

Appendix A. The Statistical Geometry Algorithm.

It has been found [1]-[3] that it is possible to create fractal patterns of a wide variety of geometric shapes by the following algorithm:

1. Create a sequence of areas A_i equal to $\frac{1}{N^c}, \frac{1}{(N+1)^c}, \frac{1}{(N+2)^c}, \frac{1}{(N+3)^c}, \ldots$. Choose an area (square, rectangle, circle, …) A to be filled.

2. Sum the areas A_i to infinity, using the Hurwitz zeta[2] function

$$\zeta(c,N) = \sum_{i=0}^{\infty} \frac{1}{(N+i)^c}$$

3. Define a new set of areas S_i by $S_i = \frac{A}{\zeta(c,N)(N+i)^c}$. It will be seen that the sum of all these redefined areas is just A.

4. Let $i = 0$. Place a shape having area S_0 in the area A at a random position x,y such that it falls entirely within area A. Increment i. This is the "initial placement".

5. Place a shape having area S_i entirely within A at a random position x,y such that it falls entirely within A. If this shape overlaps with any previously-placed shape repeat step 5. This is a "trial".

6. If this shape <u>does not</u> overlap with any previously-placed shape put x,y and the shape dimensions in the "placed shapes" data base, increment i, and go to step 5. This is a "placement".

---

[2] The definitions of the Riemann and Hurwitz zeta functions can be found in Wikipedia.

7. Stop when *i* reaches a set number, percentage filled area reaches a set value, or other.

One will note that the dimensions of the shapes are nowhere specified. They are calculated from the areas. A very wide variety of shapes have been found to be "fractalizable" in this way.

The parameters c and N can have a variety of values. The parameter c is often in the range 1.2-1.4 with a largest usable value around 1.51. N can be 1 or larger, and need not be an integer.

By construction the result is a space-filling random fractal -- if the process never halts. Available evidence [1]-[3] says that it does not. The power law area sequence ensures that it has the fractal "statistical self-similarity" (scale-free) property. And the random search ensures that no two circles will ever touch, so that the "gasket" is a single continuous object.

Appendix B. Data Fitting.

The raw data were converted to $\log_{10}$ values. These values were then fitted by least squares adjustment to a straight line with the points weighted as the *y* value (the cumulative number of trials required). Note that the weighting was done with the n_{cum} value, not its logarithm. This weighting had the effect of emphasizing the right-hand side of the curves in Fig. 2 so that the slopes (exponents) found relate to the "steady state" part of the data (large *n*).