FRACTAL PLOT

PURPOSE

Generates a fractal plot.

DESCRIPTION

DATAPLOT generates Iterated Function Systems fractals as defined by Michael Barnsley. Barnsley defines an affine transformation as follows:

$$w(x) = w\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \qquad \textbf{(EQ 2-7)}$$

Fractal plots are generated by applying one or more affine transformations in an iterative fashion to an initial starting point (DATAPLOT uses (0,0) as the starting point). The points a, b, c, and d define rotation and scaling operations to be applied to the point. The e and f points define a translation to be applied to the point. An additional value is the probability weighting. These weights are applied to a uniform random number generator to determine which of the affine transformations (if there is more than one) to apply at a given step. The a, b, c, and d points are commonly expressed as follows:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} r_1\cos\alpha_1 & -r_2\sin\alpha_2 \\ r_1\sin\alpha_1 & r_2\cos\alpha_2 \end{bmatrix} \qquad \textbf{(EQ 2-8)}$$

This form makes the nature of the scaling and rotation more explicit. DATAPLOT can generate fractals expressed in either of these formats. In addition, DATAPLOT supports an alternate form for specifying the rotation and scaling (this algorithm is due to William Withers of the US Naval Academy). It performs the following rotation and scaling to obtain the a, b, c, and d points:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}\begin{bmatrix} p & 0 \\ 0 & q \end{bmatrix}\begin{bmatrix} \cos\beta & \sin\beta \\ \sin\beta & \cos\beta \end{bmatrix} \qquad \textbf{(EQ 2-9)}$$

This form specifies an initial rotation, a scaling, then a final rotation.

DATAPLOT currently supports each of these methods for specifying fractals. Note that both forms given in angles and scaling factors transform easily to the Barnsley form (i.e., a, b, c, d). If you have the a, b, c, and d points, you can get r_1, r_2, alpha$_1$, and alpha$_2$ (i.e., the alternate form for Barnsley’s definition) as follows:

$$r_1 = \sqrt{a^2 + c^2} \qquad \textbf{(EQ 2-10)}$$

$$r_2 = \sqrt{b^2 + d^2} \qquad \textbf{(EQ 2-11)}$$

$$\alpha_1 = \arccos\left(\frac{a}{\sqrt{a^2 + c^2}}\right) \qquad \textbf{(EQ 2-12)}$$

$$\alpha_2 = \arcsin\left(\frac{b}{\sqrt{b^2 + d^2}}\right) \qquad \textbf{(EQ 2-13)}$$

For each of the 3 formats, one row of the input variables defines a single affine transformation. A fractal can be generated from one or more affine transformations. The columns specify one of the elements (e.g., a, b, etc.).

If you do not wish to specify a probability factor, simply specify all probability weights to be 1 or leave it off. The translation variables are specified the same way for all 3 forms.

SYNTAX 1 (Wither’s format)

FRACTAL PLOT <y1> <y2> <y3> <y4> <y5> <y6> <y7> <SUBSET/EXCEPT/FOR qualification>

where <y1> is the variable containing the initial rotations (i.e., α);

<y2> is the variable containing the X scalings (i.e., p);

<y3> is the variable containing the Y scalings (i.e., q);

<y4> is the variable containing the final rotations (i.e., β);
<y5> is the variable containing the X translations (i.e., e);
<y6> is the variable containing the Y translations (i.e., f);
<y7> is a variable containing the probability weightings;
and where the <SUBSET/EXCEPT/FOR qualification> is optional and rarely used in this context.

SYNTAX 2 (Barnsley's format)

FRACTAL PLOT <a> <b> <c> <d> <e> <f> <y7> <SUBSET/EXCEPT/FOR qualification>
where <a> is the variable containing the a values;
<b> is the variable containing the b values;
<c> is the variable containing the c values;
<c> is the variable containing the d values;
<e> is the variable containing the e (i.e., translation);
<f> is the variable containing the f (i.e., translation) values;
<y7> is a variable containing the probability weightings;
and where the <SUBSET/EXCEPT/FOR qualification> is optional and rarely used in this context.

SYNTAX 3 (Barnsley's rotation matrix format)

FRACTAL PLOT <y1> <y2> <y3> <y4> <y5> <y6> <y7> <SUBSET/EXCEPT/FOR qualification>
where <y1> is the variable containing the α_1 values;
<y2> is the variable containing the r_1 values;
<y3> is the variable containing the r_2 values;
<y4> is the variable containing the α_2 values;
<y5> is the variable containing the X translations (i.e, e);
<y6> is the variable containing the Y translations (i.e., f);
<y7> is a variable containing the probability weightings;
and where the <SUBSET/EXCEPT/FOR qualification> is optional and rarely used in this context.

EXAMPLES

FRACTAL PLOT Y1 Y2 Y3 Y4 Y5 Y6
FRACTAL PLOT Y1 Y2 Y3 Y4 Y5 Y6 Y7

NOTE 1

The FRACTAL TYPE command is used to specify which of the three formats is used for the fractal data. The default is Barnsley's format (i.e., SYNTAX 2 above).

NOTE 2

The following sample data files in the DATAPLOT reference directory contain examples of fractal data sets. Just replace the READ section in the program example below. Two of these are shown in the sample programs. These data files are in the Wither's format.

FRACBRAN.DAT	-	generate a branch
FRACCHRI.DAT	-	generate a Christmas tree
FRACCLOU.DAT	-	generate a cloud
FRACFERN.DAT	-	generate a fern
FRACFRON.DAT	-	generate a frond
FRACGALA.DAT	-	generate a galaxy
FRACPENT.DAT	-	generate a pentagon
FRACSPIR.DAT	-	generate a spiral
FRACSQUA.DAT	-	generate a square
FRACTRIA.DAT	-	generate a triangle

NOTE 3

The appearance of the plot is controlled by the LINE and CHARACTER settings. Typically, you want to set the LINE to blank and the CHARACTER to a "." or some other character. It is also recommended that you set the character small. This is demonstrated in the example programs below.

NOTE 4

DATAPLOT continues to generate the fractal plot until the maximum number of points for a plot has been reached. This is 20,000 or 40,000 on most current implementations. The FRACTAL ITERATIONS command can be used to set this to a smaller number (it is currently not possible to set it to a larger number).

NOTE 5

The 2 angle variables can be given in either radians or degrees. If they are given in degrees, be sure to enter an ANGLE UNITS DEGREES command before the FRACTAL PLOT command.

DEFAULT

None

SYNONYMS

None

RELATED COMMANDS

LINES	=	Sets the type for plot lines.
CHARACTER	=	Sets the type for plot characters.
CHARACTER FONT	=	Sets the font for plot characters.
ANGLE UNITS	=	Specifies whether angles are given in degrees or radians.
PLOT	=	Generates a data or function plot.
MULTIPLOT	=	Allows multiple plots per page
FRACTAL (LET)	=	Generate fractal data (of the kind used to create Koch snowflakes).

REFERENCE

DATAPLOT uses an algorithm provided by Douglass Withers of the U.S. Naval Academy.

“Fractals Everywhere,” Michael Barnsley, Academic Press, 1988.

“Chaos, Fractals, and Dynamics: Computer Experiments in Mathematics,”, Robert Devaney, Addison-Wesley, 1990.

“Chaos and Fractals: New Frontiers of Science,”, Petigen, Jurgens, and Saupe, Springer-Verlang, 1993.

APPLICATIONS

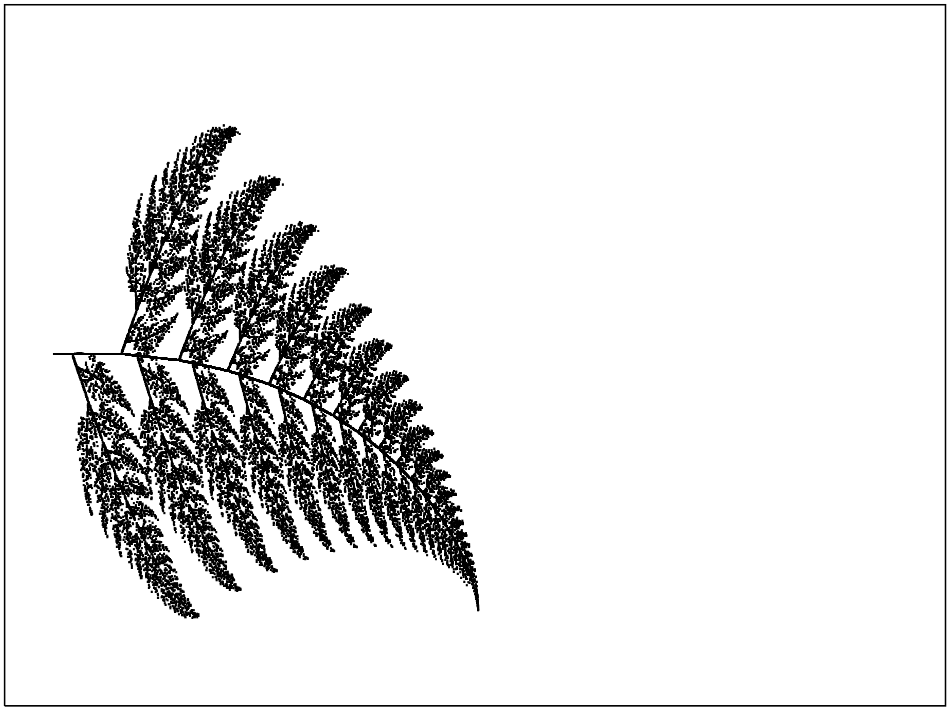
Fractals

IMPLEMENTATION DATE

88/12

PROGRAM 1

```
. Generate a fractal fern
READ Y1 TO Y7
180.000 0.160 0.001 180.000 0.000 0.000 1
 0.000 0.850 0.850 -2.500 1.600 0.000 15
180.000 0.340 0.300 229.000 1.600 0.000 2
109.709 -0.288 0.379 235.233 0.440 0.000 2
END OF DATA
FRAME OFF
FRAME COORDINATES 5 5 95 95
ANGLE UNITS DEGREES
CHARACTER FONT SIMPLEX
CHARACTER .
LINE BLANK
FRACTAL TYPE WHITHERS
FRACTAL PLOT Y1 Y2 Y3 Y4 Y5 Y6 Y7
```



PROGRAM 2

```
READ Y1 TO Y7
 0.000 0.400 -0.400 90.000 0.000 0.000 1
-35.578 0.111 0.759 -28.559 0.070 0.070 1
 35.578 0.759 -0.111 241.441 -0.070 -0.070 1
-17.912 0.641 1.956 -17.951 0.000 0.000 1
END OF DATA
FRAME OFF
FRAME COORDINATES 5 5 95 95
ANGLE UNITS DEGREES
CHARACTER JUSTIFICATION LEBO
CHARACTER .
LINE BLANK
FRACTAL TYPE WHITHERS
FRACTAL PLOT Y1 Y2 Y3 Y4 Y5 Y6 Y7
```