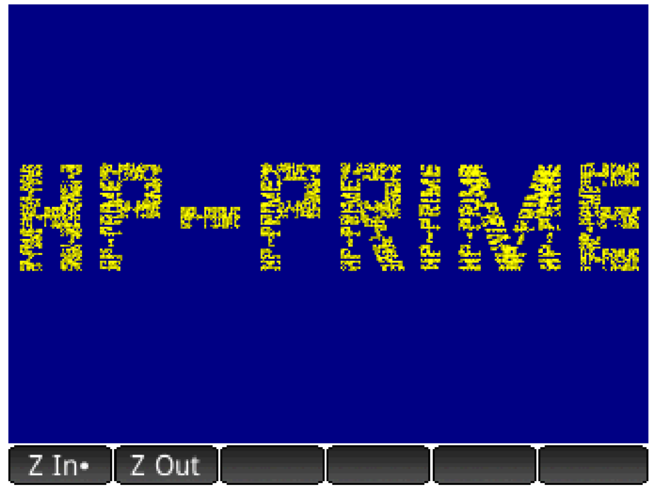
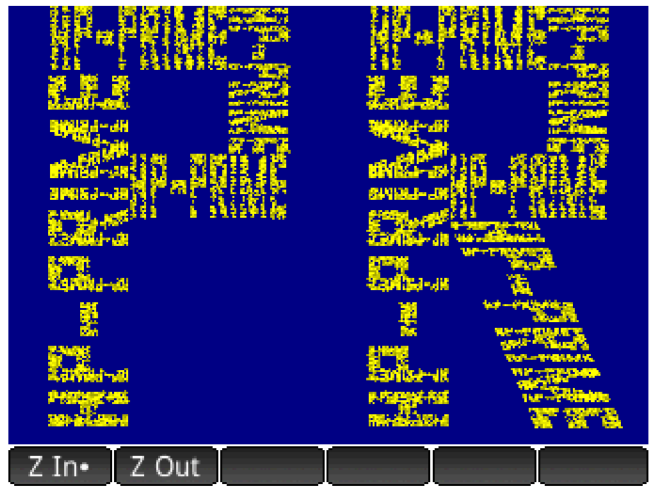
Wes Loewer's
HP-Prime
IFS Program



Introduction:

This program generates Iterated Function System fractals. These intricate patterns are produced by starting with a random (*x*,*y*) coordinate then repeatedly applying randomly selected linear transformations chosen from a list carefully crafted transforms. The resulting point, (*x'*, *y'*), is then plotted and then the process is reiterated over and over.

As you zoom into the image by tapping the screen, you'll notice that the image is self-similar. For example, as you zoom in you see that the letters making up **HP-PRIME** is made up of smaller **HP-PRIME**'s which in turn are made up of even smaller **HP-PRIME**'s, and so on.



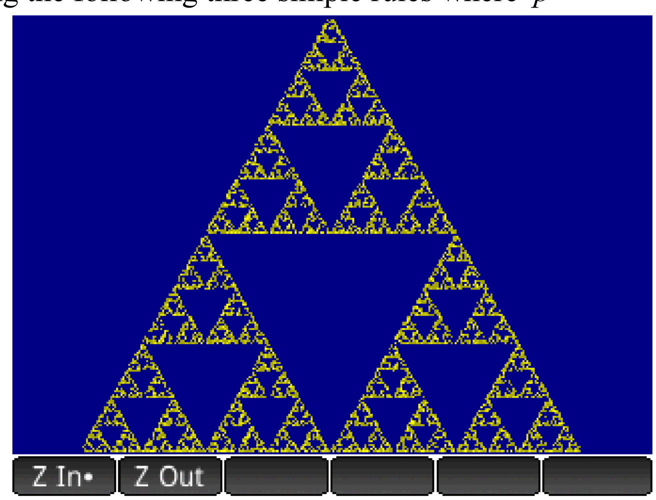
An Example:

The famous Sierpinski Triangle can be generated using the following three simple rules where p indicates the probability of that rule being selected.

Rule #1) $$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{2} \end{bmatrix}, \quad p = \frac{1}{3}$$

Rule #2) $$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \frac{\sqrt{3}}{4} \\ -\frac{1}{4} \end{bmatrix}, \quad p = \frac{1}{3}$$

Rule #3) $$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} -\frac{\sqrt{3}}{4} \\ -\frac{1}{4} \end{bmatrix}, \quad p = \frac{1}{3}$$



Starting with a random (*x*,*y*) point, one of the three rules is randomly selected and applied to (*x*,*y*), generating a new (*x*,*y*) point. This new point is plotted and the process is repeated. After hundreds of points are plotted, the pattern looks like this. Since the rules are randomly selected, the individual points will be in different locations each time the program is run. However, the over all pattern is always the same.

Details:

The linear transformation rules are stored in the program as a list.

```
        {
          a1, b1, c1, d1, e1, f1, p1,
          a2, b2, c2, d2, e2, f2, p2,
          a3, b3, c3, d3, e3, f3, p3,
          ...
        }
```

where the rows of a, b, c, d, e, and f are the values in the transformation:

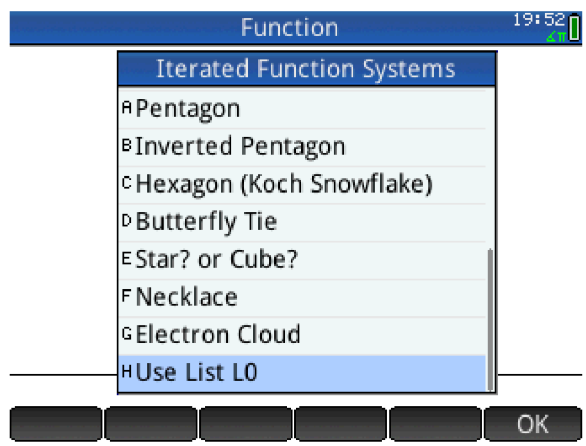
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}, \quad p$$

For example, the Sierpinski Triangle shown above is stored as a list of 21 numbers, 3 rules of 7 values.

```
        {
          0.5,   0.0,   0.0,   0.5,        0.0,    0.5, 0.333333,
          0.5,   0.0,   0.0,   0.5,   0.433013, -0.25, 0.333333,
          0.5,   0.0,   0.0,   0.5,  -0.433013, -0.25, 0.333333
        }
```

Besides the 16 built-in examples, you can generate your own shapes. An internet search of "IFS fractal galleries" will garner a plethora of definitions.

You can save your own IFS definitions in list **L0** and select the **Use List L0** option.



For example, copy and paste this to the calculator and select **Use List L0.**

```
L0 := {
        0.5,   0.0,   0.0,   0.5,   0.0,        0.5,   0.25,
        0.5,   0.0,   0.0,   0.5,   0.433013, -0.2,   0.25,
        0.5,   0.0,   0.0,   0.5,  -0.433013, -0.2,   0.25,
        0.5,   0.0,   0.0,   0.5,   0.0,        0.0,   0.25
        }
```

Enjoy,

Wes Loewer
wjltemp-prog@yahoo.com