```
1
   1 e. 0 2 3 4
0
   onein=: 1 e. ,
```
test if one is in the array

```
   (2 2,:2 2) onein;._3 b
0 0 1 1 0 0
0 0 1 1 1 0
0 1 1 1 1 0
0 1 1 1 1 1
1 1 1 0 1 1
1 1 1 1 1 1
```
apply `onein` to the tesselation;

```
   N=: [: +/@, (,:~@,~)@[ onein;._3 ]
   2 N b
25
   eps=: % #
```
epsilon

```
   2 eps b
0.153846
   fd=: (N %&^. %@eps)"0 2
```
estimate based upon this tesselation

```
   2 fd b
1.71967
```

Next we consider a larger version of the Sierpinski triangle.

```
   VRAWH=:1000 1000
   rtiter 10000
100
   $b=:|.VRA
1000 1000
   1 2 3 4 5 6 7 8 9 10 fd b
1.64822 1.65963 1.66552 1.67062 1.66397 1.67517 1.67752 1.68224 1.68849
1.67705
```

The fractal dimension is computed to be near 1.68 which is not too far from the theoretic value near 1.58. Larger versions of the image give slightly more accurate estimates.

As a second example, consider the fern shown in Figure 4.4.1. If `b` denotes the 1000 by 1000 binary matrix giving that image, we compute the following estimates of fractal dimension.

```
   1 2 3 4 5 6 7 8 9 10 fd b
1.73172 1.72922 1.72449 1.71946 1.71589 1.71194 1.7081 1.70432 1.70172
1.69914
```

It appears that the fractal dimension of that image is near 1.7.

4.8 Exercises

1. Implement the following functions using agenda to handle the cases. The domain is real numbers except (a) has integer domain.

(a) $f(x)=\begin{cases} x^2 & \text{if } x \text{ is even} \\ 1-5x & \text{if } x \text{ is odd} \end{cases}$ (b) $g(x)=\begin{cases} x^2 & \text{if } x \le 3 \\ 1-5x & \text{if } 3<x \end{cases}$ (c) $g(x)=\begin{cases} x^2 & \text{if } x \le 3 \\ 1+x^2 & \text{if } 3<x\le 7 \\ 2+x^2 & \text{if } 7<x \end{cases}$

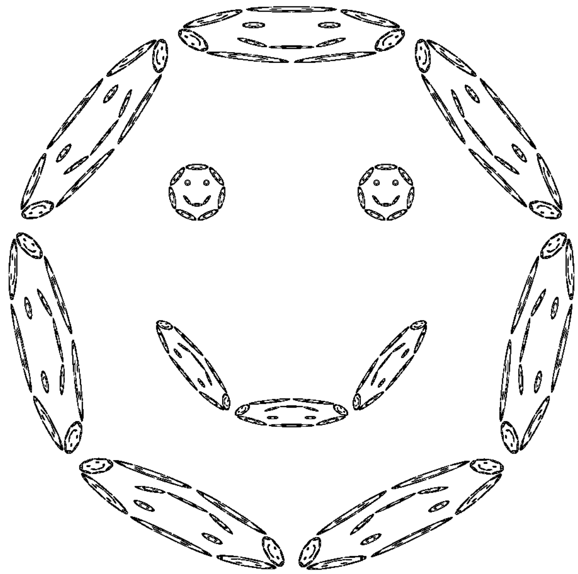

Figure 4.8.1 Smile of Smiles

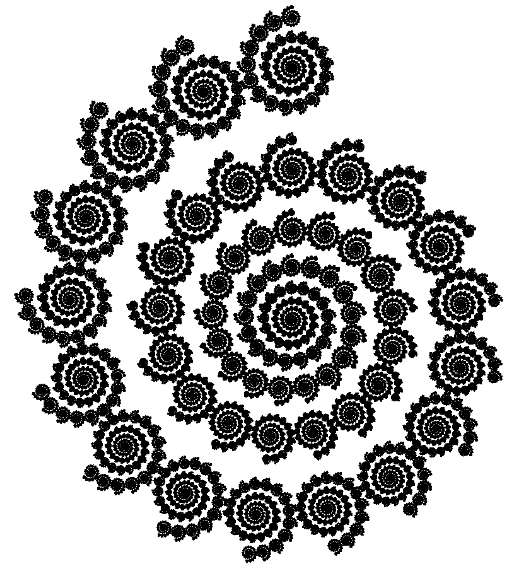

Figure 4.8.2 A Spiral of Spirals

2. Implement the functions in the previous exercise using explicit control words.

3. Consider the $3x+1$ function $t(x)$ from Section 4.1.
(a) Plot the path to 1 for each of the integers from 100 and 109.
(b) Implement a function `tn` which gives the number of elements in the path to 1 for a given number. Plot `tn` for all the integers from 1 to 200.

4. The classic $3x+1$ is defined as follows: $h(x) = \begin{cases} x/2 & if\ x\ is\ even \\ 3x+1 & if\ x\ is\ odd \end{cases}$.

Implement that function and create an analog of Figure 4.1.1 for the path to 1 generated by $h(x)$ on 27.

5. Load the script *~addons/graphics/fvj4/smiles.ijs*. It defines 12 matrices and transformations. Duplicate the smiles of smiles image in Figure 4.8.1 by defining a random transformation function `rt` in the following ways.
(a) Select the random transformations with equal probability.
(b) Find weights to bias the selection so that the image is balanced.

6. A spiral of spirals as in Figure 4.8.2 can be created using the transformations associated with `m0` and `m1` .

```
   m0                                         m1
0.2  0    0                              0.904498 0.350404  0
0    0.2  0                             _0.350404 0.904498  0
0.4  0.8  1                              0.222953 _0.12745  1
```

(a) Create an image of the iterated function system selecting those transformations with equal probability.
(b) Find weights to bias the selection so that the image is balanced.

7. Identify transformations needed to approximately replicate the spiral of smiles image shown in Figure 4.8.3. Use the transformations from the previous exercise and the smile transformations in Exercise 5 to build the ones you need.

8. Identify transformations that allow you to approximately duplicate Figure 4.2.2.

9. We are not restricted using affine maps in iterated function systems. Figure 4.10.3 illustrates a fractal version of the word CHAOS from [Reiter, B et al, 1998] which uses curved strokes for parts of some of the letters.
(a) Approximately duplicate Figure 4.8.4 using transformations of your own design.
(b) Create a curved stroke fractal version of the word "WORD".

10. Randomly chosen iterated function systems do not have the same appearance as ones that use hand crafted affine transformations. Figure 4.8.5 shows a fractal resulting from a randomly selected set of six affine

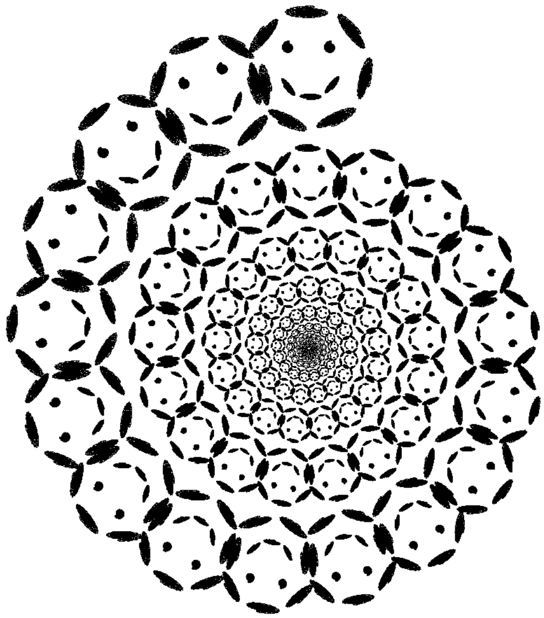

Figure 4.8.3 A Spiral of Smiles of Same

transformations. The six coefficients of the matrix in the left two columns were selected at random from between -1 and 1. Such a matrix was acceptable if it mapped the unit square back into the unit square. Six acceptable matrices were used to create the random transformation function `rt`.
(a) Implement that strategy for creating random fractals. Find several different images that way.
(b) Use box counting to estimate the fractal dimension of the fractals you produced in (a).

11. In Section 4.4 we created a weighted random index function which was used to define the function `rt` based upon transformation and also an alternate approach that used matrices.
(a) Write a tacit version of the function `rmt` from the matrix approach.
(b) Use the timing facilities (`6!:2`) to order the time required by (i) the transformation based approach in Section 4.4 (ii) the matrix based approach in Section 4.4 (iii) approach (ii) modified to use the new `rmt` from (a).

12. Apply the chaos game (a) to an isosceles right triangle and (b) to a triangle with a large obtuse angle.

13. (a) Create a variant of the chaos game where you move two thirds of the way toward the selected vertex. What does the image look like when this is applied to a triangle?
(b) Create a variant of the chaos game where you move either one third or two thirds (selected at random) of the way toward the selected vertex. What does the image look like when applied to a triangle?

14. (a) Find the fractal dimension of the X fractal appearing in Figure 4.8.6.
(b) Find the fractal dimension of the J fractal appearing in Figure 4.8.7.

15. Use the sum of powers equation at the end of Section 4.6 to compute the fractal dimension of the image in Figure 4.8.8.

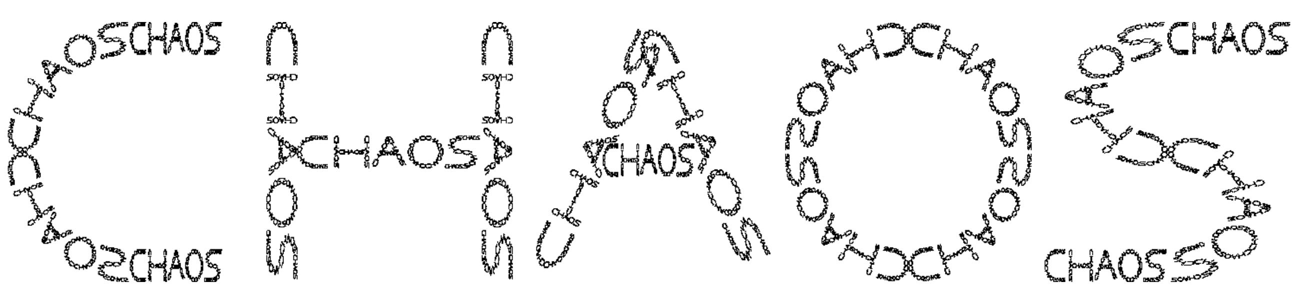

Figure 4.8.4 Chaos within Chaos

Figure 4.8.5 A Random IFS

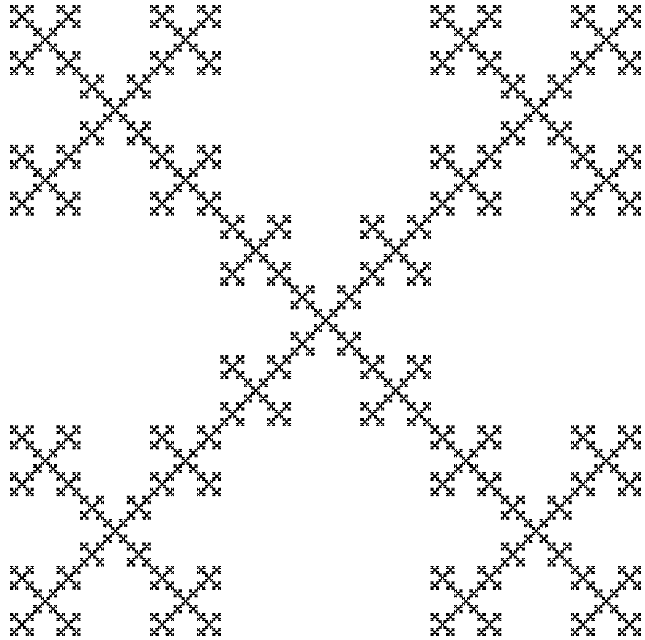

Figure 4.8.6 A Fractal X

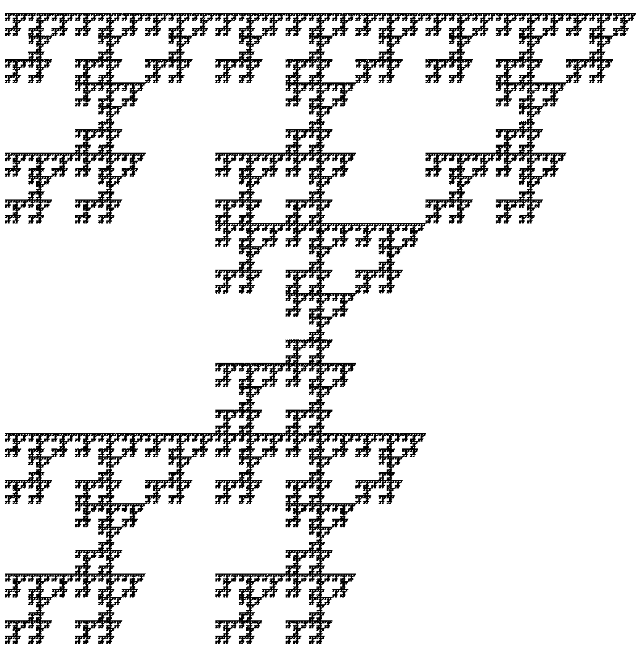

Figure 4.8.7 A Fractal J

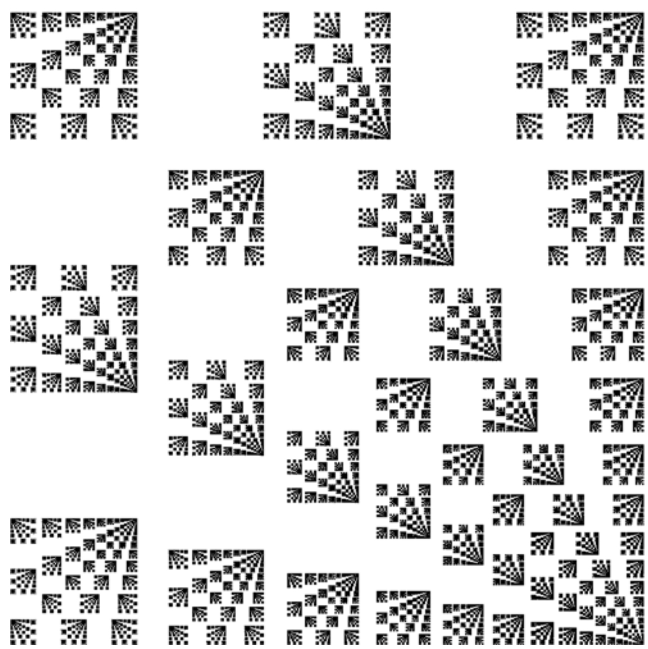

Figure 4.8.8 A Fractal from Similitudes

16. Consider the tessellations
`(3 3,:3 3)<;._3 i.6 6` and
`(1 1,:3 3)<;._3 i.6 6`
(a) How would you describe the second one?
(b) Can you now predict the result of
`(1 2,:3 3)<;._3 i.6 6`

17. In Section 4.7 non-overlapping tesselations were used to estimate fractal dimension. Use the constructions in the previous exercise to guide estimation of fractal dimension using overlapping tesselations.