Title: Affine Transformations and the "Fractal Fern"

Link to Outcomes:

- **Problem Solving** Students will demonstrate problem solving abilities by determining the preimage and all subsequent images to be generated, deciding which points are used in an affine transformation. Students will find transformation formulas based on a set of preimage, and image coordinates.

- **Communication** Students will discuss mathematical concepts and solutions within groups as well as present the outcomes to other groups. Students will be able to define and explain affine transformations, preimage and image.

- **Reasoning** Students will analyze the projected images and reason to determine the best method to refine the final image. Students will determine how an arbitrary preimage will perform when transformed and predict the resulting images.

- **Connections** Students will conjecture the significance of the ability to generate figures, such as the fern, with minimum input data within other disciplines (e.g.,. - the arts, computer science, biology).

- **Estimation** Students will estimate point values on a Cartesian Coordinate Plane.

- **Geometry** Students will observe first-hand the application of the concept of self-similarity, which is fundamental to fractal geometry.

- **Algebra** Students will perform affine transformations, will plot points on the Cartesian Coordinate system as well as understand the concepts of functions, images, and preimages, and the algebraic concepts needed to determine transform coefficients.

- **Patterns/ Relationships** Students will observe how altering original data or IFS codes affects the projected image.

Brief Overview:

Affine transformations are used to generate fractal images because they allow complex graphic computer images to be created and stored with little input and memory. Students will define affine transformation verbally and mathematically and demonstrate how it is used by applying a given transformation to a specified point. Students will define and identify preimages and images and their corresponding points. Instructor will then explain how an affine transformation performs the operations of shrinking, stretching, rotating, and shearing.

Given specific points, students will use trial and error or other methods to find an affine transformation that will generate one point (or points) from some initial point. Students will then be asked to determine a system of equations that will generate the affine transformation. Students will then superimpose a fern-like fractal image on a Cartesian Coordinate Plane. They will extract data from the graph by identifying the coordinates significant points on the preimage and subsequent points on projected images. Students will then input the initial data into the designated computer spreadsheet and subsequent graphing programs (i.e., - FINDTRAN and ITERTRAN). Based on the output, students will refine their initial data until a satisfactory fern is generated. Students will then generate other images by choosing an arbitrary preimage and a set of images.

Grade/Level:

Grades 9 - 12; Geometry through Algebra II/Trigonometry

Duration/Length:

This lesson is expected to take three to four 45-minute class periods.

Prerequisite Knowledge:

Students should be able to graph on the Cartesian Coordinate Plane, should understand the concepts of self-similarity, fractals, transformations, preimages, and images. Students should preferably have some background in fundamental computer operating.

Objectives:

- Define self-similarity, strict self-similarity, and fractals.
- Perform transformations from preimage points to image points.
- Discuss methods for generating fractal images manually and by computer.
- Perform successive transformations to generate images from a given preimage using manual and computer methods.
- Plot points on the Cartesian Coordinate Plane.
- Develop a system of equations that will determine an affine transformation.
- Identify self-similar parts of a fractal image.
- Express conclusions verbally.
- Explain affine transformations and IFS codes and how they are performed and developed.

Materials/Resources/Printed Materials:

- Transparency of “fractal fern.” (reference: Peitgen, Heinz-Otto and Saupe, Dietmar (Eds.), The Science of Fractal Images, New York: Springer-Verlag 1988, p. 242.)
- Transparent quadrile with graduated x- and y-axes.
- Quadrile paper.
- Computers with TurboPascal software.

- IFS Coding charts.
- Affine/Iteration Transformation Program
- Overhead Projector

Development/Procedures:

DAY 1:

- Define “affine transformation” - the transformation of the form

  $$T(x,y) = (ax + by + c, dx + ey + f)$$

  For example:

  $$T(x,y) = (\tfrac{1}{2} x + y - 2, 2x - \tfrac{1}{2} y + 1)$$

- Demonstrate how to apply this transformation using the point (4, -7). ($T(4, -7) = (-7, 12 \tfrac{1}{2})$)

- Have students plot points on the Cartesian Coordinate System, connect them with line segments to create a simple figure (6 - 8 points), label the points with letter names A, B, C, etc., and record the coordinates of each point (all coordinates should be between -10 and 10 inclusive).

- Have students apply this transformation to each point in their figure. Name these image points A’, B’ C’, etc. (so that, for example, T(A) = A’).

- Students should then graph the image points and connect them in the same manner as they did the preimage points. The result should be a distorted version of the preimage. (The image should be a reflection about the y-axis, shifted vertically upwards, rotated clockwise and stretched.)

- Explain to the students that a single affine transformation will always perform the following operations: horizontal and vertical shrinking and/or stretching, rotation about the origin, shearing (i.e. - skewing so that one side of a rectangle remains fixed while the opposite side slides parallel to the fixed side), horizontal and vertical translation or sliding.

DAY 2:

- Place students in pairs and ask them to find the formula for an affine transformation T such that T(2, 1) = (3, 0). They can do this by trial and error. Many possible answers are possible; e.g., T(x, y) = (x + y, x - 2y).

- Next ask students to find an affine transformation that meets the above condition and also maps (4, 0) to (3, 2). This is more challenging, although many answers are still possible; e.g., T(x, y) = (x + 2y - 1, x - 2).

- Now add the condition that T(-6, 2) = (-3, -5). Have students discuss whether they can find the transformation in a systematic way. It can be done by substituting the coordinates of preimage and corresponding image points into the form of an affine transformation. This results in the following 3 by 3 systems of equations:

$$\begin{aligned} 2a + 1b + c &= 3 \\ 4a + 0b + c &= 3 \\ -6a + 2b + c &= -3 \end{aligned} \qquad \begin{aligned} 2d + 1e + f &= 0 \\ 4d + 0e + f &= 2 \\ -6d + 2e + f &= -5 \end{aligned}$$

- Have students solve these systems. The first one yields a = 1, b = 2, and c = -1. The second systems yields d = ½, e = -1, and f = 0, so the formula for T is T(x, y) = (x + 2y - 1, ½ x - y). Have students check this with the three given points.

- Emphasize that three non-collinear preimage points and their corresponding images are necessary to determine the affine transformation.

- Demonstrate the spreadsheet FINDTRAN by typing in the preimage and image coordinates above. It should yield the same results.

- For practice with the spreadsheet, have students find the affine transformation that maps (2, 5) to (-1, 3), (0, -4) to (2, 0), and (-6, 2) to (-3, -2). It should give T(x, y) = (.409x - .424y + .303, .545x + .212y + .848)

DAY 3:

- Place students in pairs. Distribute transparencies of quadrile paper (the smaller the squares the better) along with photocopies of the “fractal fern”.

- Have students identify the self similar parts of the fern. Then have them identify the preimage and three subsequent images. Determine whether the fern is “strictly” self-similar. (It is not. The stem of the fern from the endpoint up to the second branch is not similar to the whole.) Discuss results as a class.

- Students will then choose three points on the preimage and label them A, B and C. Have the students lay the quadrile transperency over the photocopy of the fern and draw the x- and y-axes. Students will then find the coordinates of the three selected points (Different pairs will get different results due to different positions of the ferns on the coordinate plane. This is okay because the transformation is irrespective of the points, only their relationship to the image points, which is consistent.) Have students record coordinates on the Transformation Worksheet.

- Students will then choose three points on all four subsequent images and find their coordinates. (The fourth image will be the stem and will be generated by three collinear points along the segment from the end point to the second branch). These points will be labeled T_1(A,B), T_2(A,B), T_3(A,B), and T_4(A,B). The students then record the results on the transformation worksheet.

- Have the students run the FINDTRAN program. At the appropriate places in the program, students will input their recorded coordinates. The program will generate Iterating Function System (IFS) codes. These numbers should be recorded on the bottom of their worksheets.

- Once the IFS codes have been recorded, students should exit the FINDTRAN program and open and run the Pascal program ITERTRAN. At the appropriate prompts, students should input appropriate maximum and minimum x and y values based on the scale that they recorded their initial points, and the resultant IFS codes. The program will generate an image on the screen that should closely resemble the original “fractal” fern.

- Students may then discuss in pairs or as a class what can be done to refine the screen image of the fern.

Evaluation:

Students will be asked to identify a fractal image and its self similar parts. Then they will be asked to select portions of said image that would effectively be used in an affine transformation. Students will then be asked to define or explain affine transformations, describing the procedure used to obtain the IFS codes from a given fractal image. Then students will be required to generate that image from the IFS codes.

Extension/Follow Up:

Have students create their own fractal image, by first creating a simple image on quadrile paper and transforming the image two to four times using specific transformation formulas. Students would then be asked to input the resulting preimage and image points, generate the IFS codes using the FINDTRAN program, input the IFS codes into the ITERTRAN program, and generate a fractal image. Students might be asked to predict how their image will look before they run the ITERTRAN program. More advanced students might determine the method for finding the IFS codes without the computer program.

Authors:

Mr. Geoffrey Birky
Greenwood Mennonite School
Sussex County (DE)

Mrs. Brooke Abercrombie
Friendly High School
Prince George’s County

Fractal Attraction
Barnsley's IFS Coding System

Iterated Function System -- set of geometric functions (transformations) that are iterated to produce a fractal attractor.

The result is determined by the transformations (how the figure is reduced, rotated, sheared, and slid) only. The initial figure has no effect.

Image

Find pieces similar to whole, determine transformations

Transformations
(IFS Code) Stored and/or transmitted

Iterate

Image reconstructed

Affine Transformations

T(preimage) =	image
T(x,y)	= (.5x + y - 2, 2x - .5y + 1)
preimage	image
A = (,)	A' = T(A) = (,)
B = (,)	B' = T(B) = (,)
C = (,)	C' = T(C) = (,)
D = (,)	D' = T(D) = (,)
E = (,)	E' = T(E) = (,)
F = (,)	F' = T(F) = (,)

Affine transformations are of the form:

$$(T(x,y) = (ax + by + c, dx + ey + f)$$

The "code" is just the coefficients a, b, c, d, e, f

Find an affine transformation T such that T(2,1) = (3,0)

Find T so that T (4,0) = (3,2) **and** T(2,1) = (3,0)

Now add the requirement that T (-6, 2) = (-3, -5) and find an affine transformation.

Affine transformations for the Fractal Fern

A = (,) B = (,) C = (,)

T_1(A) = (,) T_1(B) = (,) T_1(C) = (,)
T_1(x,y) =

T_2(A) = (,) T_2(B) = (,) T_2(C) = (,)
T_2(x,y) =

T_3(A) = (,) T_3(B) = (,) T_3(B) = (,)
T_3(x,y) =

T_4(A) = (,) T_4(B) = (,) T_4(C) = (,)
T_4(x,y) =

IFS code:	a	b	c	d	e	f
T_1						
T_2						
T_3						
T_4						

Spreadsheet FINDTRAN for Microsoft Works
(Uses Cramer's Rule to solve two 3x3 systems of linear equations to find the coefficients of an affine transformation given three preimage points and their corresponding images.)

	A	B	C	D	E	F
1		Preimage x	Preimage y		Image x	Image y
2	Point A					
3	Point B					
4	Point C					
5						
6	D =	=b2*(c3-c4)+b3*(c4-c2)+b4*(c2-c3)			D =	=b2*(c3-c4)+b3*(c4-c2)+b4*(c2-c3)
7	Da =	=e2*(c3-c4)+e3*(c4-c2)+e4*(c2-c3)			Dd =	=f2*(c3-c4)+f3*(c4-c2)+f4*(c2-c3)
8	Db =	=b2*(e3-e4)+b3*(e4-e2)+b4*(e2-e3)			De =	=b2*(f3-f4)+b3*(f4-f2)+b4*(f2-f3)
9	Dc =	=b2*(c3*e4-c4*e3)+b3*(c4*e2-c2*e4)+b4*(c2*e3-c3*e2)			Df =	=b2*(c3*f4-c4*f3)+b3*(c4*f2-c2*f4)+b4*(c2*f3-c3*f2)
10	IFS	code:				
11	a =	=b7/b6			d =	=f7/f6
12	b =	=b8/b6			e =	=f8/f6
13	c =	=b9/b6			f =	=f9/f6

```
{Program to generate fractal image by iterating affine
transformations using IFS code.
Written for Turbo Pascal by Geoffrey Birky 7/10/95}

program itertransf;
uses graph;
const
  xmin = -30.0; xmax = 70.0; {left and right edges of screen}
  ymin = 0.0; ymax = 80.0; {bottom and top edges of screen}
  ntransf = 4; {number of different affine transformations}
  niters = 32000;
var
  driver,mode,iter,transf: integer;
  x,y,xnew: real;
  a,b,c,d,e,f: array[1..ntransf] of real;

{scaling functions convert arbitrary data coordinates to pixel numbers}
function scalex(x: real): integer;
  begin scalex := trunc((x-xmin)/(xmax-xmin)*getmaxx+0.5) end;
function scaley(y: real): integer;
  begin scaley := trunc((ymax-y)/(ymax-ymin)*getmaxy+0.5) end;

begin {itertransf}  {Enter IFS code here}
a[1] := -0.151; b[1] := 0.254;  c[1] := 17.3;
d[1] := 0.269;  e[1] := 0.240; f[1] := -0.5;
a[2] := 0.200; b[2] := -0.232;  c[2] := 12.0;
d[2] := 0.249;  e[2] := 0.222; f[2] := 8.3;
a[3] := 0.853; b[3] := 0.033;  c[3] := 2.2;
d[3] := -0.032;  e[3] := 0.848; f[3] := 12.5;
a[4] := 0.0; b[4] := 0.0;  c[4] := 15.0;
d[4] := -0.060;  e[4] := 0.173; f[4] := 0.9;

{Initialize graphics. Start at random point.}
randomize;
x := random(1000)/1000*(xmax-xmin)+xmin;
y := random(1000)/1000*(ymax-ymin)+ymin;
driver := detect;
initgraph(driver,mode,'c:\tp\bgi');
putpixel(scalex(x),scaley(y),15);

{iterate the randomly chosen transformations}
for iter := 1 to niters do
  begin
  transf := trunc(random(ntransf))+1;
{To make the fern look good, uncomment the following two lines}
```

```
{ transf := trunc(random(ntransf+4))+1;
   if transf>ntransf then transf := 3;   }
   xnew := a[transf]*x+b[transf]*y+c[transf];
   y := d[transf]*x+e[transf]*y+f[transf];
   x := xnew;
   putpixel(scalex(x),scaley(y),15)
   end; {for iter}

 {Draw frame to indicate completion, wait for enter key.}
 rectangle(scalex(xmin),scaley(ymin),scalex(xmax),scaley(ymax));
 readln;
 closegraph
 end. {itertransf}
```