Iteration and Dynamic Geometry: Beyond Common Fractals with *The Geometer's Sketchpad*

Session 142
NCTM Philadelphia
22 April 2004

Nicholas Jackiw
KCP Technologies
njackiw@keypress.com
http://www.keypress.com/sketchpad

Nathalie Sinclair
nathsinc@math.msu.edu
Michigan StateUniversity

Contents

Review: What's New in GSP 4.0?

GSP 4.0 has many, many improvements over previous versions, and we will only be able to visit some of them in this workshop. Here are some of the important new developments we'll encounter.

1. **Selection** has changed: no more shift key!

2. **Animation** is now interactive. (You can continue working or dragging while objects are animating.) Use the **Motion Controller** to start and stop animations, and to adjust speeds and direction.

3. You can define **functions** using the Graph menu (**New Function**). Once you've defined a function, you can also plot it (as a Cartesian or polar graph), evaluate it (in the Calculator), or even differentiate it.

4. **Iteration** allows you to repeat geometric and analytic constructions, or numeric calculations, to examine sequences and series, fractals and dynamical systems.

5. **Split** and **Merge** allow you to redefine existing objects, both to correct mistakes and to investigate changing conditions.

6. **Tables** can now automatically collect data during an animation or iteration.

Here are several new features we specifically *won't* address in this workshop (unless there is time at the end during questions):

1. **Custom Tools** allow you to encapsulate constructions into new commands, as well as to create entire microworlds with your own tools.

2. **Parametric Color** allows you to use color as a "3^{rd}-dimension" of visualization, and to create surface plots.

3. **JavaSketchpad** allows you to save sketches you've designed in Sketchpad as interactive illustrations in your HTML web pages.

For a more detailed introduction to GSP 4.0's many new features, on your own be sure to investigate the sample sketch that comes with Sketchpad:

C:\Sketchpad\Samples\How To\HowTo_UpgradeFromV3.GSP

This sketch contains many examples of new GSP 4.0 functions and how to use them.

About Iteration

In mathematics, iteration refers to the act of performing of some mathematical process—a computation, algorithm, or construction—on some initial value, and then *repeating* that process on the result. Each repetition of the process is a single *iteration* of the process. For example, counting whole numbers—0, 1, 2, 3, 4—can be thought of as starting with the *initial value* of zero, and then *iterating* the operation of "adding one."

Iteration in Mathematics

Iteration appears across mathematics, from the simplest arithmetic operations (such as counting) to very recent ideas about fractals and chaos. In the school curriculum, iterative processes may show up in:

- Arithmetic Procedures (e. g. counting, long division, etc.)
- Sequences and Series
- Calculus — Limits and Approximations
- Fractals & Chaos
- Optimization
- Geometric Art & Design
- Recursion
- Dynamical Systems, Statistical Mechanics, etc.

In addition, iterative phenomena and processes frequently appear in physics, biology, and chemistry.

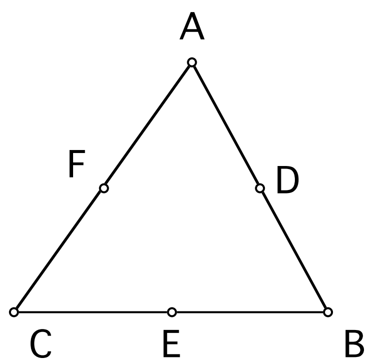
Iteration and Mathematical Thinking / Problem Solving

The repetitive nature of iterative processes makes them appropriate to a variety of problem contexts and problem solving strategies:

- **Extrapolation**:
  If you can't find an answer, can you find a process that leads to an answer? (If so, iterate that process until you reach the solution.)
- **Incremental Approximation:**
  If you can't find an answer, can you find a process that allows you to improve an arbitrary guess? (If so, iterate that process until your approximation is "close enough" for the purposes of the problem.)
- **Divide & Conquer:**
  If a problem is too hard to solve, can it be split into two smaller problems which are easier to solve? If so, iteratively "divide" the problem into small enough parts that the parts can be solved, and then assemble your result from the "conquered" problem parts.

Iteration in Sketchpad

Mathematicians use many notations to describe iterative processes. As a geometry tool, Sketchpad uses a geometric notation for describing iterations. In such a notation, *pre-image objects* (such as points or numbers) are iterated toward *image* objects. These image objects must *depend* on the pre-image objects, such that when the pre-images change, the images change. Once you "map" pre-images to images in the **Iterate** dialog box, Sketchpad repeats the mapping iteratively. You can set the number of iterations to control how many times Sketchpad repeats your construction or calculation. Here's an example.

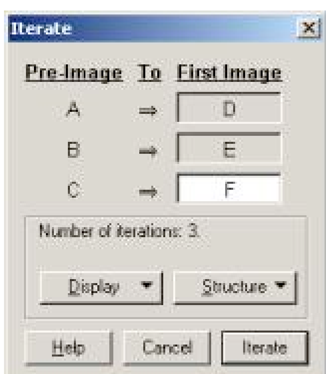


1. Construct the figure at left. D, E, and F are the midpoints of the sides of ΔABC.

Now you are going to "map" ΔABC to ΔDEF and iterate that mapping. This will repeat your triangle and its midpoints on the midpoints of the original triangle, giving you a smaller triangle inside the larger one.

Iterating this process will give you a sequence of inner "midpoint triangles."

2. Select **A**, **B**, and **C**—the pre-image (independent points) that you want to iterate. Then choose **Iterate** from the **Transform** menu.



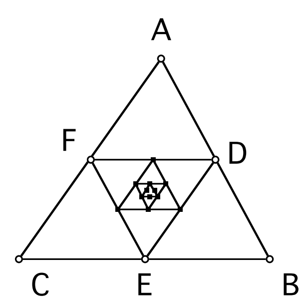
3. Fill out the dialog box as shown, choosing points D, E**, and** F **(by clicking them in your sketch) as the "first images" of points** A, B**, and** C**.**

As soon as you begin defining mappings, Sketchpad shows you the second, third, and fourth images of your iterated map in the sketch.

Adjust the number of iterations in the Display menu, and click Iterate when done.

Sketchpad displays the iterated result. To change the number of iterations after leaving the **Iterate** dialog box, select the entire figure and press

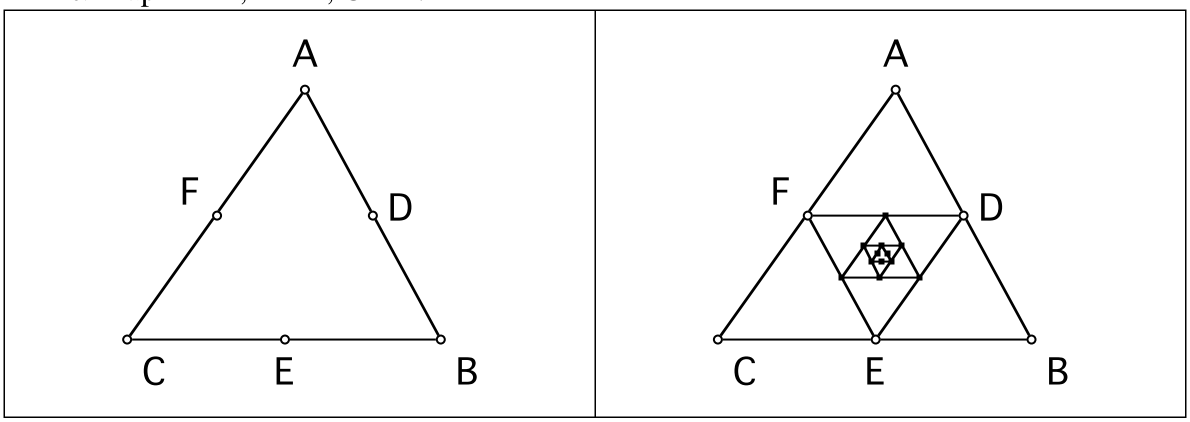
+: to increase the iterations by 1
-: to reduce the iterations by 1



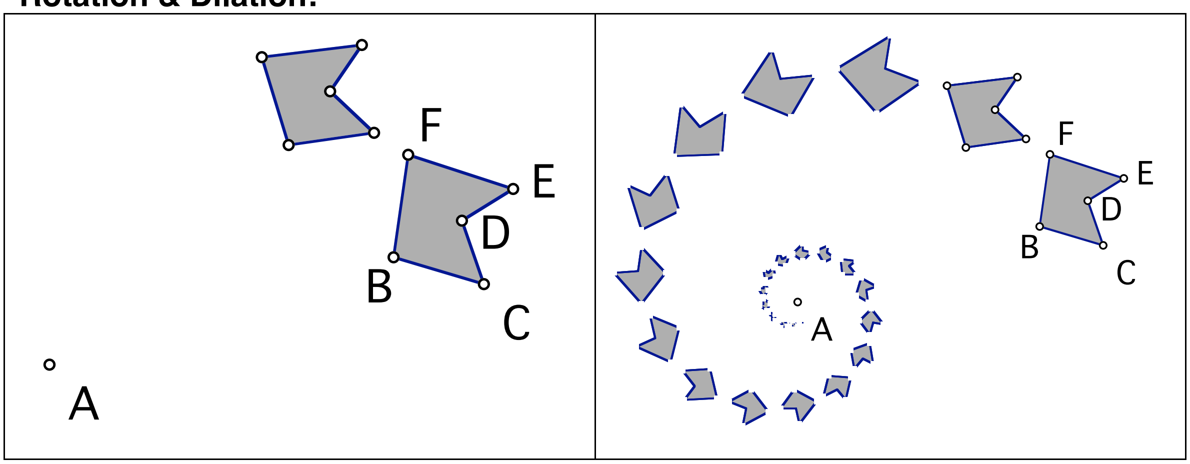
Geometric Iteration Challenges

For **geometric iteration**, use the **Iterate** dialog box to map pre-image *points* to image *points*. Sketchpad repeats your construction as these iterated points move to their images. After iterating, experiment with the level of iteration by pressing **+** or **–** when your iterated construction is selected.

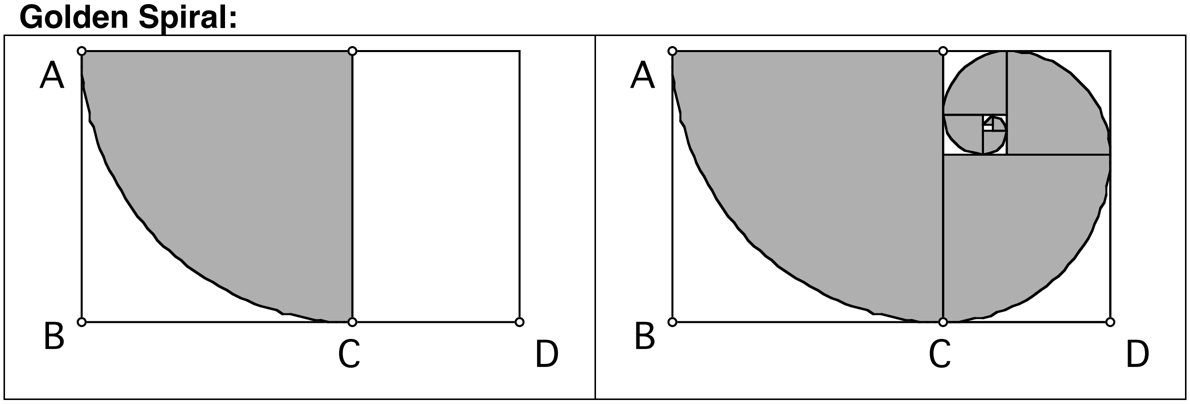
Midpoint Polygon
Hint: Map A⇒E, B⇒F, C⇒D.



Rotation & Dilation:



Golden Spiral:



Numeric Iteration Challenges

For **numeric iterations,** use the **Iterate** dialog box to map one "parameter" pre-image to a calculation based on that parameter. As you increase the level of iteration (*n*), Sketchpad displays a table containing the values of all iterated measurements and calculations. Numeric iteration can be used to explore numeric sequences and series.

Basic Function Iteration

Hint: First define pre-image seed (using **Graph | New Parameter**); then define *f*(x) (**New Function**), then calculate *f*(seed) (**Calculate**). Iterate seed⇒*f*(seed). Then try changing your function *f* (by double-clicking it) to explore other iterated functions.

$$\text{seed} = 1.00$$
$$f(x) = 2 \cdot x$$
$$f(\text{seed}) = 2.00$$

⇒

n	f(seed)
0	2.00
1	4.00
2	8.00
3	16.00
4	32.00
5	64.00

Approximating the Square Root of *p* (Hero's Method)

$$p = 5.00000$$
$$guess_0 = 1.00000$$
$$\frac{guess_0 + \frac{p}{guess_0}}{2} = 3.00000$$

n	$\frac{guess_0 + \frac{p}{guess_0}}{2}$
0	3.00000
1	2.33333
2	2.23810
3	2.23607
4	2.23607
5	2.23607

$$\sqrt{p} = 2.23607$$

Fibonnaci Numbers ($f_n = f_{n-2} + f_{n-1}$)

$$f_0 = 1.00$$
$$f_1 = 1.00$$
$$f_0 + f_1 = 2.00$$

⇒

n	f_0+f_1
0	2.00
1	3.00
2	5.00
3	8.00
4	13.00

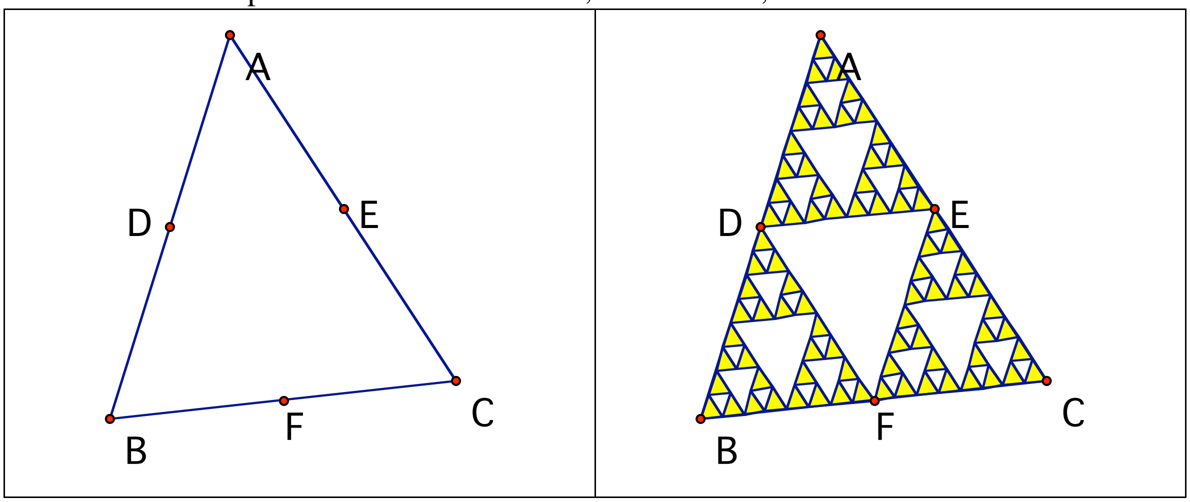
Numeric iteration can be combined with geometric visualization: **Plot** your individual calculated values before iterating, and Sketchpad will plot the entire series after iterating.

Fractal Iteration Challenges

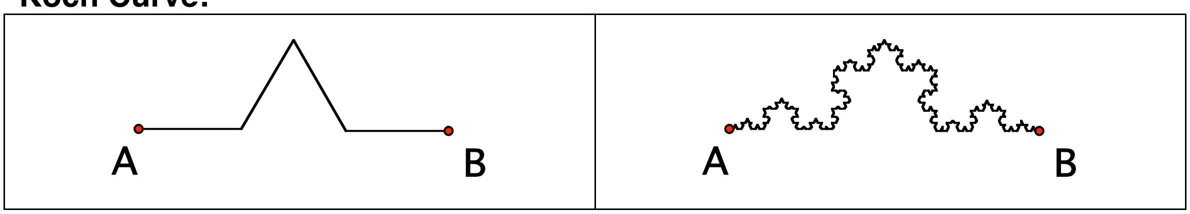
Fractal iterations frequently involve more than one map—that is, the same pre-image or set of pre-images maps to two or more sets of post-images simultaneously. To create multiple mappings in the **Iterate** dialog box, use **Structure: Add New Map**.

Sierpinski Gasket:

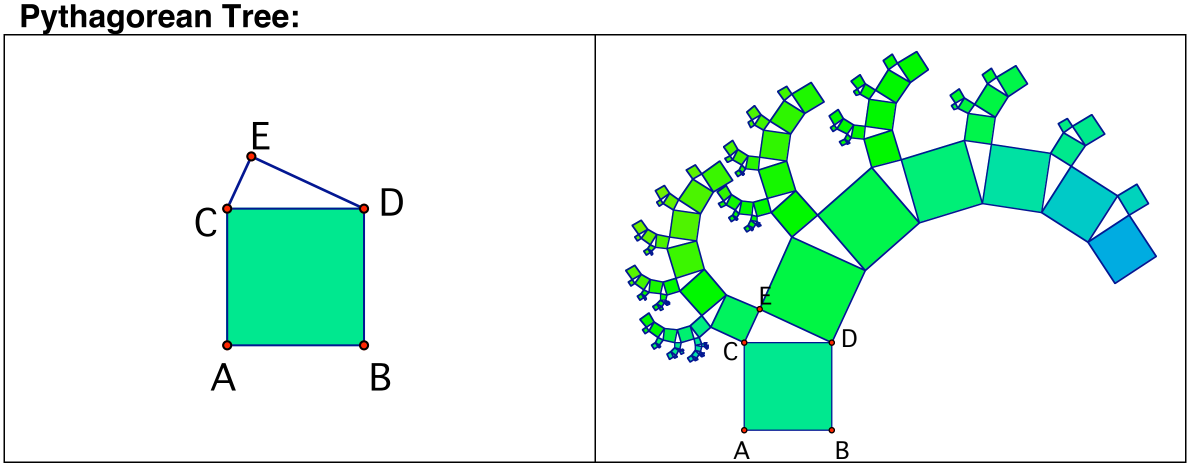
Hint: Make **3** Maps of ΔABC: ABC⇒ADE, ABC⇒DBF, ABC⇒ECF.



Koch Curve:



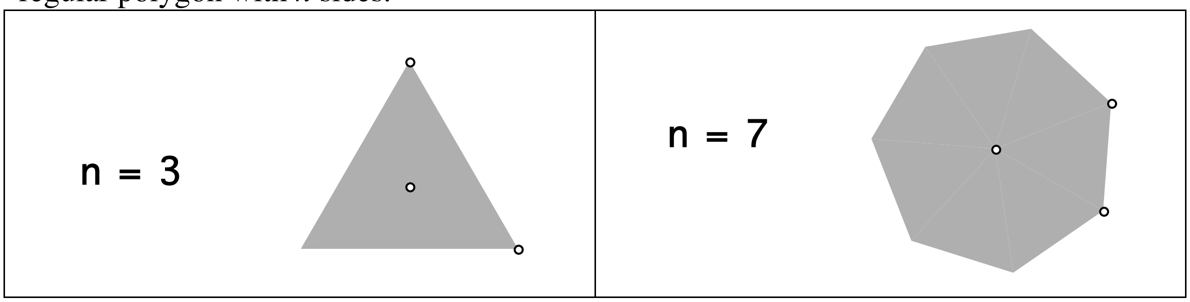
Pythagorean Tree:



Advanced Iteration Problems

The following modeling challenges are harder, and may require you to combine numeric and geometric iterative techniques. You may not have time to try them all, so choose one that interests you. Even if you can't completely model the system in Sketchpad, discuss with your partners how you would approach the problem mathematically. We will review several of them in detail at the end of the workshop.

1. The Regular n-Gon
Given a parameter *n* whose value can be changed to 3, 4, 5 …, use iteration to construct a regular polygon with *n* sides.

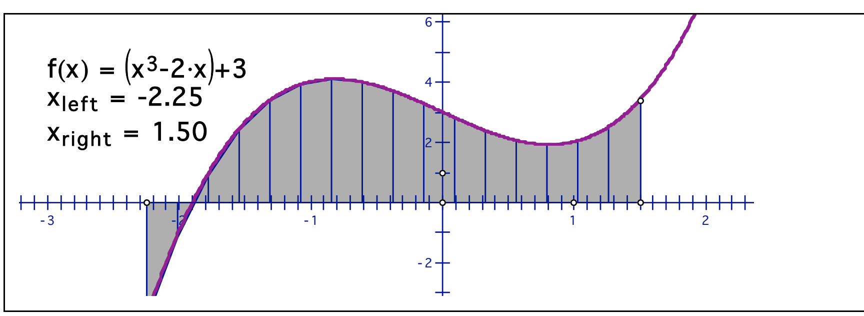


Hint: How does the vertex angle of an *n*-gon depend on *n*? **Extensions**: (1) Also calculate the area and perimeter of your constructed n-gon. (2) Also construct a star-polygon with *n* vertices.

2. Projectile Motion
Given a projectile traveling at a given velocity, use iteration to model the flight path of the projectile.

Hints: Use a vector (a directed segment) to represent initial velocity and another to represent gravity, and a point to represent initial position. How do velocity and gravity affect position and velocity?

3. Numeric Integration
Given a function *f* and the left and right coordinates of an interval on the domain, approximate the area under the curve with quadrilaterals. Use an iterated construction so that the number of quadrilaterals increases as you increase the level of iteration.



Hint: Start with the quadrilateral through (x1, 0), (x1, *f*(x1)), (x2, 0), and (x2, *f*(x2)). How could you cut this quadrilateral into two parts?