Fractals: Recursion, Paradoxes, and Applications

Jorge Balbás

Institute for Pure and Applied Mathematics (UCLA)
jbalbas@ipam.ucla.edu

Dept. of Mathematics, Cal State University, Northridge
jorge.balbas@csun.edu

Curtis Center Math & Teaching Conference
UCLA
March 5th, 2016
Goal

I want to convince you that *Fractals* is a great topic to teach to high-school students and present some ideas.

Why?
Goal

I want to convince you that Fractals is a great topic to teach to high-school students and present some ideas

Why?

- It is an fascinating topic! – good to engage students
Goal

I want to convince you that Fractals is a great topic to teach to high-school students and present some ideas

Why?

- It is an fascinating topic! – good to engage students
- The level of difficulty ranges from the very accessible to the most complex ideas
Goal

I want to convince you that Fractals is a great topic to teach to high-school students and present some ideas.

Why?

- It is an fascinating topic! – good to engage students
- The level of difficulty ranges from the very accessible to the most complex ideas
- Great topic to motivate and introduce many ideas often taught in high school
Goal

I want to convince you that *Fractals* is a great topic to teach to high-school students and present some ideas

Why?

- It is an fascinating topic! – good to engage students
- The level of difficulty ranges from the very accessible to the most complex ideas
- Great topic to motivate and introduce many ideas often taught in high school
- Also to motivate and introduce very accessible ideas that are not usually taught in high school
Goal

I want to convince you that Fractals is a great topic to teach to high-school students and present some ideas.

Why?

- It is an fascinating topic! – good to engage students
- The level of difficulty ranges from the very accessible to the most complex ideas
- Great topic to motivate and introduce many ideas often taught in high school
- Also to motivate and introduce very accessible ideas that are not usually taught in high school, but could very well be
Goal

I want to convince you that Fractals is a great topic to teach to high-school students and present some ideas.

Why?

- It is an fascinating topic! – good to engage students
- The level of difficulty ranges from the very accessible to the most complex ideas
- Great topic to motivate and introduce many ideas often taught in high school
- Also to motivate and introduce very accessible ideas that are not usually taught in high school, but could very well be
- Connects Math with Computer Science and Programming
Goal

I want to convince you that Fractals is a great topic to teach to high-school students and present some ideas.

Why?

- It is an fascinating topic! – good to engage students
- The level of difficulty ranges from the very accessible to the most complex ideas
- Great topic to motivate and introduce many ideas often taught in high school
- Also to motivate and introduce very accessible ideas that are not usually taught in high school, but could very well be
- Connects Math with Computer Science and Programming
- Fractals are everywhere...
1 Intro
   - Motivation
   - Self Similarity

2 The Math
   - Recursive Processes
   - Some Programming
   - In Fractal Language
   - The Paradox

3 Other Examples
   - $L - systems$
   - Mandlebrot like Fractals

4 Applications
**Manmade Designs**

Whether it is our everyday surroundings,
Motivation

Manmade Designs

the most admired pieces of architecture,
or the most advanced engineering desings,

most of what we design consists of lines, circles, and smooth curves
Nature

But those smooth objects are not very convenient to describe other shapes we often see in nature.

These display a fractal geometry.
What Are Fractals?

**Definition:** A natural phenomenon or a mathematical set that exhibits a *repeating pattern* that displays at *every scale*. There are other definitions, but the key features are *self similarity* and *recursion*.
Recursion

- A powerful concept in mathematics and computer science characterized by:
  - A simple base case (or cases) – a terminating scenario (does not use recursion)
  - A set of rules that reduce all other cases toward the base case

Examples:
- Koch curve:
  - Given two points, replace the middle third of the line connecting them with two lines of the same length forming an equilateral triangle; repeat the process on the resulting four line segments $n$ times...
Recursion

- A powerful concept in mathematics and computer science characterized by:
  - A simple base case (or cases) – a terminating scenario (does not use recursion)
  - A set of rules that reduce all other cases toward the base case
Recursive Processes

Recursion

- A powerful concept in mathematics and computer science characterized by:
  - A simple base case (or cases) – a terminating scenario (does not use recursion)
  - A set of rules that reduce all other cases toward the base case

Examples:

- Factorial: \( n! = n \cdot (n - 1) \cdot (n - 2) \ldots 3 \cdot 2 \cdot 1 \)
  - Base case: \( 0! = 1 \)
  - Recursive rule: \( n = n \cdot (n - 1)! \)
Recursive Processes

Recursion

- A powerful concept in mathematics and computer science characterized by:
  - A simple base case (or cases) – a terminating scenario (does not use recursion)
  - A set of rules that reduce all other cases toward the base case

Examples:

- Fibonacci numbers: 1, 2, 3, 5, 8, 13, 21, …
  - Base case: $F_0 = 1$, $F_1 = 2$
  - Recursive rule: $F_n = F_{n-1} + F_{n-2}$
Recursion

- A powerful concept in mathematics and computer science characterized by:
  - A simple base case (or cases) – a terminating scenario (does not use recursion)
  - A set of rules that reduce all other cases toward the base case
- Examples:
  - Koch curve:
    Given two points, replace the middle third of the line connecting them with two lines of the same length forming an equilateral triangle; repeat the process on the resulting four line segments \( n \) times...
The above description of the Koch curve is *iterative*: draw a line segment, replace middle third with two lines, remove the middle third of each of the resulting four line segments... repeat this $n$ times...

This is hard and too long to implement

Much easier to do it *recursively*
The above description of the Koch curve is *iterative*: draw a line segment, replace middle third with two lines, remove the middle third of each of the resulting four line segments... repeat this $n$ times...

This is hard and too long to implement

Much easier to do it *recursively*

- Base case: draw the line between the two given points
- Recursive rule: find the points delimiting the middle third and the new point where the two new lines meet (not on the line), and repeat $n - 1$ times for each of the resulting four segments...
function koch_curve($p_1$, $p_2$, $k$):
    if $k == 0$: -- base case
        plot line connecting $p_1$ and $p_2$
    else:
        $A = \frac{\sqrt{3}}{6} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$ -- rotation + scaling matrix!
        $p_3 = (2p_1 + p_2)/3$
        $p_5 = \frac{1}{2}(p_1 + p_2) + A(p_1 - p_2)$
        $p_4 = (p_1 + 2p_2)/3$
        koch_curve($p_1$, $p_3$, $k - 1$)
        koch_curve($p_3$, $p_5$, $k - 1$)
        koch_curve($p_5$, $p_4$, $k - 1$)
        koch_curve($p_4$, $p_2$, $k - 1$)
We now apply the recursive algorithm to the three sides of a triangle...
Snowflake Curve – Remarks

- Recursion makes it easy, but too many recursive calls slow down the process, takes up too much memory!
Snowflake Curve – Remarks

- Recursion makes it easy, but too many recursive calls slow down the process, takes up too much memory!
- This fractal belongs to a more general class of fractals known as \( L - \text{systems} \), which consist of:
Snowflake Curve – Remarks

- Recursion makes it easy, but too many recursive calls slow down the process, takes up too much memory!

- This fractal belongs to a more general class of fractals known as \( L – \text{systems}, \) which consist of:
  
  - An *alphabet* (a set of variables): \( V = \{ A, B, \ldots \} \)
  - A set of constants \( \omega = \{ \alpha, \beta, \gamma, \ldots \} \)
  - A set of *production* rules: combinations of variables and constants to replace previous variables
Snowflake Curve – Remarks

- Recursion makes it easy, but too many recursive calls slow down the process, takes up too much memory!
- This fractal belongs to a more general class of fractals known as \( L - systems \), which consist of:
  - An alphabet (a set of variables): \( V = \{A, B, \ldots\} \)
  - A set of constants \( \omega = \{\alpha, \beta, \gamma, \ldots\} \)
  - A set of production rules: combinations of variables and constants to replace previous variables
- In our case:
  - \( V = F \) – move forward
  - \( \omega = \{+, -\} \) – turn 60° left (+) or right (−)
  - \textit{axiom} (initial conditions): \( F \)
  - rules: \( F \rightarrow F + F - -F + F \)
Properties of this Snowflake

What is the area of the snowflake after $n$ iterations?
Properties of this Snowflake

- What is the area of the snowflake after \( n \) iterations?
  - initial area: \( A_o = a_o = \frac{\sqrt{3}}{4} \)
Properties of this Snowflake

- What is the area of the snowflake after $n$ iterations?
  - initial area: $A_o = a_o = \frac{\sqrt{3}}{4}$
  - in each iteration we add $T_n = 3 \cdot 4^{n-1}$ new triangles of area
    $a_n = \frac{a_{n-1}}{9} = \frac{a_o}{9^n}$
Properties of this Snowflake

- What is the area of the snowflake after \( n \) iterations?
  - initial area: \( A_0 = a_0 = \frac{\sqrt{3}}{4} \)
  - in each iteration we add \( T_n = 3 \cdot 4^{n-1} \) new triangles of area
    \[ a_n = \frac{a_{n-1}}{9} = \frac{a_0}{9^n} \]
  - the area added in iteration \( n \) is \( T_n \cdot a_n = \frac{3}{4} \cdot \left( \frac{4}{9} \right)^n \cdot a_0 \)
Properties of this Snowflake

- What is the area of the snowflake after \( n \) iterations?
  - initial area: \( A_0 = a_0 = \frac{\sqrt{3}}{4} \)
  - in each iteration we add \( T_n = 3 \cdot 4^{n-1} \) new triangles of area
    \[
    a_n = \frac{a_{n-1}}{9} = \frac{a_0}{9^n}
    \]
  - the area added in iteration \( n \) is
    \[
    T_n \cdot a_n = \frac{3}{4} \cdot (\frac{4}{9})^n \cdot a_0
    \]
  - so, after \( n \) iterations, \( A_n = a_0 \left( 1 + \frac{3}{4} \sum_{k=1}^{k=n} \left( \frac{4}{9} \right)^k \right) \)
Properties of this Snowflake

What is the area of the snowflake after \( n \) iterations?

- initial area: \( A_o = a_o = \frac{\sqrt{3}}{4} \)
- in each iteration we add \( T_n = 3 \cdot 4^{n-1} \) new triangles of area \( a_n = \frac{a_{n-1}}{9} = \frac{a_o}{9^n} \)
- the area added in iteration \( n \) is \( T_n \cdot a_n = \frac{3}{4} \cdot (\frac{4}{9})^n \cdot a_0 \)
- so, after \( n \) iterations, \( A_n = a_o \left( 1 + \frac{3}{4} \sum_{k=1}^{n} (\frac{4}{9})^k \right) \)
- and calculating the geometric sum, we obtain:

\[
A_n = \frac{a_o}{5} \cdot \left( 8 - 3 \left(\frac{4}{9}\right)^n \right)
\]
Properties of this Snowflake

- What is the area of the snowflake after \( n \) iterations?
  - initial area: \( A_o = a_o = \frac{\sqrt{3}}{4} \)
  - in each iteration we add \( T_n = 3 \cdot 4^{n-1} \) new triangles of area
    \[
    a_n = \frac{a_{n-1}}{9} = \frac{a_o}{9^n}
    \]
  - the area added in iteration \( n \) is \( T_n \cdot a_n = \frac{3}{4} \cdot (\frac{4}{9})^n \cdot a_0 \)
  - so, after \( n \) iterations, \( A_n = a_o \left( 1 + \frac{3}{4} \sum_{k=1}^{n} (\frac{4}{9})^k \right) \)
  - and calculating the geometric sum, we obtain:
    \[
    A_n = \frac{a_o}{5} \cdot \left( 8 - 3 \left( \frac{4}{9} \right)^n \right)
    \]

- Similarly, we can find the perimeter, \( P_n \), after \( n \) iterations:
Properties of this Snowflake

- What is the area of the snowflake after \( n \) iterations?
  - initial area: \( A_o = a_o = \frac{\sqrt{3}}{4} \)
  - in each iteration we add \( T_n = 3 \cdot 4^{n-1} \) new triangles of area
    \[ a_n = \frac{a_{n-1}}{9} = \frac{a_o}{9^n} \]
  - the area added in iteration \( n \) is \( T_n \cdot a_n = \frac{3}{4} \cdot \left( \frac{4}{9} \right)^n \cdot a_0 \)
  - so, after \( n \) iterations, \( A_n = a_o \left( 1 + \frac{3}{4} \sum_{k=1}^{n} \left( \frac{4}{9} \right)^k \right) \)
  - and calculating the geometric sum, we obtain:
    \[ A_n = \frac{a_o}{5} \cdot \left( 8 - 3 \left( \frac{4}{9} \right)^n \right) \]

- Similarly, we can find the perimeter, \( P_n \), after \( n \) iterations:
  - each side of the triangle has initial length: \( s_o = 1 \)
Properties of this Snowflake

- What is the area of the snowflake after $n$ iterations?
  - initial area: $A_o = a_o = \frac{\sqrt{3}}{4}$
  - in each iteration we add $T_n = 3 \cdot 4^{n-1}$ new triangles of area
    $$a_n = \frac{a_{n-1}}{9} = \frac{a_o}{9^n}$$
  - the area added in iteration $n$ is $T_n \cdot a_n = \frac{3}{4} \cdot \left(\frac{4}{9}\right)^n \cdot a_0$
  - so, after $n$ iterations, $A_n = a_o \left(1 + \frac{3}{4} \sum_{k=1}^{n} \left(\frac{4}{9}\right)^k\right)$
  - and calculating the geometric sum, we obtain:
    $$A_n = \frac{a_o}{5} \cdot \left(8 - 3 \left(\frac{4}{9}\right)^n\right)$$

- Similarly, we can find the perimeter, $P_n$, after $n$ iterations:
  - each side of the triangle has initial length: $s_o = 1$
  - in each iteration we add $N_n = 4 \cdot N_{n-1} = 3 \cdot 4^n$ sides of length
    $$s_n = \frac{s_{n-1}}{3} = \frac{s_o}{3^n}$$
Properties of this Snowflake

- What is the area of the snowflake after $n$ iterations?
  - initial area: $A_o = a_o = \frac{\sqrt{3}}{4}$
  - in each iteration we add $T_n = 3 \cdot 4^{n-1}$ new triangles of area
    $a_n = \frac{a_{n-1}}{9} = \frac{a_o}{9^n}$
  - the area added in iteration $n$ is $T_n \cdot a_n = \frac{3}{4} \cdot (\frac{4}{9})^n \cdot a_0$
  - so, after $n$ iterations, $A_n = a_o \left(1 + \frac{3}{4} \sum_{k=1}^{n} \left(\frac{4}{9}\right)^k\right)$
  - and calculating the geometric sum, we obtain:
    $$A_n = \frac{a_o}{5} \cdot \left(8 - 3 \left(\frac{4}{9}\right)^n\right)$$

- Similarly, we can find the perimeter, $P_n$, after $n$ iterations:
  - each side of the triangle has initial length: $s_o = 1$
  - in each iteration we add $N_n = 4 \cdot N_{n-1} = 3 \cdot 4^n$ sides of length
    $s_n = \frac{s_{n-1}}{3} = \frac{s_o}{3^n}$
  - and, after $n$ iterations, we obtain:
    $$P_n = N_n \cdot s_n = 3 \cdot s_o \cdot \left(\frac{3}{4}\right)^n$$
Properties of this Snowflake

If we repeat the process over and over, i.e., let $n \to \infty$, we’ll find:

- **Area:**
  \[
  \lim_{n \to \infty} A_n = \lim_{n \to \infty} \frac{a_o}{5} \cdot \left(8 - 3 \left(\frac{4}{9}\right)^n\right) = \frac{8}{5} \cdot a_o = \frac{2\sqrt{3}}{5}
  \]
Properties of this Snowflake

If we repeat the process over and over, i.e., let $n \to \infty$, we’ll find:

- **Area:**
  \[
  \lim_{n \to \infty} A_n = \lim_{n \to \infty} \frac{a_o}{5} \cdot \left( 8 - 3 \left( \frac{4}{9} \right)^n \right) = \frac{8}{5} \cdot a_o = \frac{2 \sqrt{3}}{5}
  \]

- **Perimeter:**
  \[
  \lim_{n \to \infty} P_n = \lim_{n \to \infty} 3 \cdot s_o \cdot \left( \frac{3}{4} \right)^n = \infty
  \]

That is: the area remains finite, but we need an infinitely long curve to enclose it!!! or we can draw an infinitely long curve within a finite area!!!
Properties of this Snowflake

If we repeat the process over and over, i.e., let $n \to \infty$, we’ll find:

- **Area:**
  \[
  \lim_{n \to \infty} A_n = \lim_{n \to \infty} \frac{a_o}{5} \cdot \left( 8 - 3 \left( \frac{4}{9} \right)^n \right) = \frac{8}{5} \cdot a_o = \frac{2\sqrt{3}}{5}
  \]

- **Perimeter:**
  \[
  \lim_{n \to \infty} P_n = \lim_{n \to \infty} 3 \cdot s_o \cdot \left( \frac{3}{4} \right)^n = \infty
  \]

- That is:
  - the area remains finite, but we need an infinitely long curve to enclose it!!!
  - or
  - we can draw an infinitely long curve within a finite area!!!
Sierpinski Gasket

Q: Can you write the pseudo-code use to generate these?
L – systems

Branching

Q: How would you define this as an \( L - system \)? Alphabet?, Constants?, Rules?
Mandlebrot and Newton Fractals

These are generated using complex numbers and evaluating a function iteratively.
Applications

(From fractalfoundation.org)

- Heat exchanger
- Cell phone antenna
- Cable design
- Fluid mixer
Thank you very much!