Higher-Dimensional Koch Curves

Yuliya Semibratova
NetID: semibra2

December 15, 2013

1 Introduction

The optimization of materials is a key part of materials science and engineering. Interestingly, the optimizing process often takes advantage of the natural problems present in materials, defects. Typically, these defects are classified according Euclidean dimensions (0, 1, 2, etc.)[1]. While this approach works for certain defects, such as those in Table 1.1: Euclidean Defects, it does not work for defects that are curvilinear or those that cannot be described using standard equations. The other defects, since they do not exist in any Euclidean (integer) dimension, must lie in fractional (fractal) dimension, denoted by D (see Table 1.2: Non-Euclidean Defects).

One such defect, the grain boundary, whose dimension is between one and two, can be used to measure the fracture toughness of the material.[2]As such, production processes that vary the fractal dimension of the grain boundary can be used to strengthen or weaken a material. According to studies by H. Khanbareh, J. H. Kruhl, and M. Nega[3], grain boundaries can be approximated by the Koch Curve fractal. The traditional Koch curve, which starts with a one-dimensional line segment and exists in the dimension log4/log3, repeats according to the following process:

1. Take an equilateral triangle.

[1]Callister, William D. & Rethwisch, David G. (2012) Hoboken, NJ: John Wiley & Sons, Inc.

[2]Khanbareh. H. (2011, December). Fractal Dimension Analysis of Grain Boundaries of 7XXX Aluminum Alloys and Its Relationship to Fracture Toughness. Retrieved from http://www.lr.tudelft.nlfileadmin/FaculteitLRImagesNovAMPictures/researchMSc/projects/Hamide thesis.pdf

[3]See 2, Kruhl, J.H. & Nega, M. (1996). Geologische Rundschau, 85, 38-43. DOI:10.1007 BF00192058

2. Split the sides into three equal lengths.
3. On any outward facing side (a side that is not connected to a previous triangle), take the central length and create an equilateral triangle from it.
4. For each line segment in the set, repeat from 1.[4]

Other versions of the Koch curve manipulate the initial line segment in different manners. Technically, the curve will remain a Koch curve fractal if it is self-similar and exists in a fractional dimension between one and two. With some extrapolation, the Koch process can be extended to higher dimensional starting points. In particular, if the starting point is a three-dimensional object, such as a regular tetrahedron or a cube (analogs of the triangular and square patterns for D between 1 and 2), then the resulting form will be a fractal with D between 2 and 3.

This project aimed to create fractals with tetrahedral and cubic starting points which would mirror the behavior of Koch curves with dimensions log4/log3 and log8/log4, respectively.

Table 1.1: Euclidean Defects

Dimension (d)	Defect
0	Point defect
1	Linear defect
2	Planar defect
3	Three-dimensional defect

Table 1.2: Non-Euclidean Defects

Dimension (D)	Defect
(1,2)	Grain boundary
(2,3)	Surface defect

2 Visualization: RTICAs

The Higher-Dimensional Koch Curves project had two main components, mathematical research and visualization. The visual aspect of the project was created in Python + OpenGL and was based largely on the oc.py[5] and blue.py[6] models. oc.py was used as an example of

[4]Weisstein, Eric W. (2013). http//:mathworld.wolfram.com/KochSnowflake.html
[5]First created by George Francis in IrisGL, April 1989
[6]George Francis, OpenGL, November 2013

creating and rotating three-dimensional objects in Python + OpenGL, whereas blue.py was used to enable forward and backward motion.

2.1 Running the RTICAs

2.1.1 Software

To function, the RTICAs require Python and OpenGL. The OpenGL files must be placed in the same folder that the RTICA is in, or Python will be unable to find them. As the Python files were written in Python 2.7, they may not be compatible with Python 3.

2.1.2 Keyboard Controls

Both RTICAs have the same set of keyboard controls. 'f' is used for forward motion, 'a' is used for backward (aft) motion, and 'y' is used to switch the rotation mode.

2.2 Tetrahedron: trikoch_[Stage].py

The first RTICA, trikoch_.py, is analogous to the traditional Koch curve[7]. The initial stage is a regular tetrahedron (and consequently, the sides are equilateral triangles). For each triangle, the fractal process is as follows:

1. Define the midpoints of the triangle.
2. Create an equilateral triangle (center triangle) whose vertices are the midpoints of the previous triangle.
3. Use the center triangle as the base for a regular tetrahedron.
4. In the next stage, perform the same process on the sides of the regular tetrahedron and on the triangles defined by two midpoints of the previous triangle and one vertex of the previous triangle (outer triangles).

[7] D=log4/log3

In effect, for each triangle, the subdivision process is the same as that of the Sierpinski Gasket. In order to make the process like that of a Koch Curve, which is an additive fractal, trikoch_.py also includes extrusions.

The Python program is written recursively. The recursive method calculates the midpoints of the previous triangle (which is passed to the method) and passes the three outer triangles back to the recursive method. In addition, the recursive method finds the normal vector through the centroid of the center triangle and scales it to have the same magnitude as the side length of the center triangle. Then, the method defines three more triangles, whose vertices are two vertices of the center triangle and the tip of the normal vector. These triangles (extruded triangles) are also passed to the recursive method.

The drawing of these figures is done separately. The original tetrahedron is drawn within the display method, as it is the zero-th stage. Within the recursive method, the vertices of the extruded triangles are added to an array, to be stored. Then, within the display callback, these vertices are drawn using the glVertex3fv() process. The figure is flat shaded through placing one of three colors (determined by a modulus) at each vertex of the tetrahedron.

In order to increase the speed at which the program runs, it uses display lists; at the outset of the program, the points the given stage of the fractal are calculated and put into a display list.

Previous versions of the program could switch between stages, but the current form cannot, as it can use only one display list. If the project is picked up in the future, hopefully this issue will be fixed.

2.3 Cube: cubekoch_[Stage].py

The second RTICA, cubekoch_.py, is analogous to a Koch Curve with D=log8/log4=1.5. In a planar view, this fractal appears to be made up of added squares. Therefore, when converted to a higher dimension, the starting point was made into a cube. For each square face, the fractal process is as follows:

1. Define the points at one third and two thirds of the way along each side of the square (side length L).
2. Define the center square as that which is formed from the points (0, 1/3L, 1/3L) (0, 1/3L, 2/3L) (0, 2/3L, 1/3L) (0, 2/3L, 2/3L), where the top left corner of the previous

square is (0, 0, 0).

3. Use the center square as the base of a cube with side length 1/3L. Call the sides of this cube the extruded faces.

4. In the next stage, perform the same process on the extrude faces and on the squares of side length 1/3L which can be formed from the area surrounding the center square.

Like the tetrahedron stellation RTICA, cubekoch_.py functions recursively. The recursive method is passed a set of four points, from which it calculates the center square and the eight outer squares. The eight outer squares are passed back to the recursive method. From the center square, the recursive method calculates the four points "above" the center square. Then, the sides of the extruded cube are defined and passed back to the recursive method.

The RTICA's display method is built in the same manner as that of trikoch_.py. However, since there are more points to calculate for each extruded cube, the program fails earlier than trikoch_.py. Even using display lists, it is only capable of running three stages before crashing Python.

3 Results

3.1 Fractal Dimension

The fractal dimension can be defined as:

$$D=\ln(N)/\ln(r)$$

where N is the number of pieces created from each single piece in the previous stage and r is the factor by which is piece is smaller in length than the previous single piece.[8] According to Paul Meakin, the same process can be applied to a fractal which increases in surface area, rather than length. Then, the equation becomes:

$$D=\ln(S)/\ln(r).$$

[8]Meakin, Paul. (1986). "Fractal scaling in thin film condensation and material surfaces" Critical Reviews in Solid State and Materials Sciences, 13(2), 148-189. DOI:10.1080/01611598608241265

In this case, S is the number of surface areas which replace the single surface in the previous stage and r is defined the same way as before.

3.1.1 Tetrahedron

In the tetrahedral fractal, each triangle is replaced by six triangles (three outer triangles on the base and three extruded triangles) whose side lengths are 1/2 of the side length of the triangle in the previous stage. Therefore, N=6 and r=2.

The fractal dimension is:

$$D=\ln(6)/\ln(2)=2.58.$$

3.1.2 Cube

In the cubic fractal, each square is replaced by thirteen squares (eight on the base and five extruded squares) whose side lengths are 1/3 of the side length of the square in the previous stage. Therefore, N=13 and r=3.

The fractal dimension is:

$$D=\ln(13)/\ln(3)=2.33.$$

In accordance with the tendency visible in fractals with dimensions between one and two, the cubic fractal is less "rough" than the tetrahedral fractal.

There are also differences to be noted between the Koch surfaces–the tetrahedral and cubic fractals–and the Koch curves. For the Koch curves, the analog of the cubic fractal has a higher dimension than the analog of the tetrahedral fractal. The fact that the surfaces have fractal dimensions that are arranged conversely in size is slightly surprising.

A possible explanation would be that the scaling factor, r, is different for the surfaces than for the curves. However, for both the surfaces and the curves, r is larger for the square-shaped fractals than for the triangle-shaped fractals. Therefore, the reason for this difference is unclear.

4 Conclusions

4.1 Convergence to a Cube

From the fourth stage of the tetrahedral fractal, it is clear that the fractal is converging to a three-dimensional solid, namely a cube. As the extruded tetrahedra become smaller in size, their adds less and less "roughness" to the fractal. If the RTICA were to go to a higher stage, presumably the sides of the cube would be filled in with smaller and smaller tetrahedra, eventually creating a completely flat face.

A similar example of the convergence of a repeated pattern to another form is Robert Fathauer's "Fractal Crystal Sculpture," which stellates a half-sized cube onto each side of the previous cube.[9] Unfortunately, the similarity is not complete. Fathauer's form is not a true fractal. Although it is self-similar, it exists in a Euclidean dimension. This fact is evident when computing the D of the object.

If the stellated cube were moved to the side of the previous cube, then there would be three equal-sized surface areas left over, in addition to the five extruded faces of the cube. Then, N=8 and r=2, due to the halfway decrease in size.

$$D=\ln(8)/\ln(2)=3.$$

Since the structure exists in a Euclidean dimension, it cannot be considered a fractal. Nonetheless, this structure appears to be the closest analog to the cubic convergence available in modern study.

4.2 Further Research

Further research would be beneficial in the fields of RTICA optimization and fractal convergence. If the RTICA could be optimized such that further stages could be generated, then it would be possible to discern whether the cubic fractal converges to another three-dimensional object. It would also be interesting to determine how other forms[10] stellate.

[9] Fathauer, Robert. "Fractal Crystal Sculpture" (2007). Retrieved from http://mathartfun.com/shopsite_sc/store/html/Art/FractalCrystalArt.html

[10] Perhaps octahedra and dodecahedra, to keep with the theme of platonic solids.

5 Bibliography

1. Baker, Martin John. (2012). 3D Theory - OpenGL Rendering. Retrieved from http://www.euclideanspace.com/threed/rendering/opengl/

2. Burns, Joe. (2012). So You Want A Page Jump, Huh?. Retrieved from http://www.html goodies.com/tutorials/getting_started/article.php/3479511

3. Callister, William D. & Rethwisch, David G. (2012) Hoboken, NJ: John Wiley & Sons, Inc.

4. Fathauer, Robert. Fractal Crystal Sculpture (2007). Retrieved from http://mathartfun.com/ shopsite_sc/store/html/Art/FractalCrystalArt.html

5. Ivanova, V.S., I.J Bunin and V.I. Nosenko. (1998). JOM, 50 (1), 52-54. Retrieved from http://link.springer.com/article/10.1007

6. Khanbareh. H. (2011, December). Fractal Dimension Analysis of Grain Boundaries of 7XXX Aluminum Alloys and Its Relationship to Fracture Toughness. Retrieved from http://www.lr.tudelft.nl//fileadmin/Faculteit/LR/Images/NovAM/Pictures_research/MSc projects/Hamide_thesis.pdf

7. Kruhl, J.H. & Nega, M. (1996). Geologische Rundschau, 85, 38-43. DOI:10.1007/ BF00192058

8. Meakin, Paul. (1986). Fractal scaling in thin film condensation and material surfaces Critical Reviews in Solid State and Materials Sciences, 13(2), 148-189. DOI:10.1080/ 01611598608241265

9. Sweeney, J. (2008, October). 3D Koch Snowflake. Retrieved from http://www.3dvision.com/ wordpress/2008/10/30/3d-koch-snowflake/

10. Wahl, Bernt Rainer. Chapter 4: Calculating Fractal Dimensions. Retrieved from http://www.wahl.org/fe/HTML_version/link/FE4W/c4.htm

11. Weisstein, Eric W. (2013). Koch Snowflake. Retrieved from http://mathworld. wolfram.com/KochSnowflake.html