Koch Extrusions on Regular Polyhedra

IGL Mathviz 2019
Project Narrative

Joshua Holder
jholder3@illinois.edu

December 1, 2019

1 Abstract

This project re-implements and extends Yulyia Semibratova's 2013 project Higher Dimensional Koch Curve's[1]. I have used Yuliya's code for koch extrusions on a tetrahedron in Python and re-implemented it in an interactive online web page using Javascript/WebGL. I further extended functionality to two more cases, the octahedron and icosahedron. After seeing koch extrusions on a tetrahedron seemingly converge into a cube and extrusions on a cube seemingly converge into an octahedron, it was conjectured that the other platonic solids would also converge to a platonic solid. However, after completing the octahedron and icosahedron it was apparent that neither converge into a platonic solid. A remaining possible explanation for the seeming convergence of the tetrahedron and cube are the dihedral angles of the polyhedra. The extrusions on a tetrahedron produce a tetrahedron on each face, and it's possible that the dihedral angle between the extrusions create the cube shape.

2 Mathematics

This project sought to explore the properties of fractals made by extrusions on regular polyhedra. A common fractal is the Koch Curve. This is made by:

- Create an equilateral triangle
- Remove the middle third from each line segment
- Create a new equilateral triangle using the length and location of the removed piece
- Repeat the process for each line segment

This creates an object of dimension $\frac{ln4}{ln3}$. The analogue implemented in this project is the following process:

- Create a regular polyhedron
- Remove the middle third from each face
- From the removed section, extrude a new regular polyhedron
- Repeat the process for each face

Yuliya discovered for the cases of a tetrahedron and cube the dimensions are $\frac{ln(6)}{ln(2)}$ and $\frac{ln(13)}{ln(3)}$ respectively.

This has been applied to the tetrahedron, octahedron, and icosahedron. In each of these three cases the polyhedron extruded is a tetrahedron. This was chosen because the faces of each of these polyhedra are equilateral triangles and naturally forms a tetrahedron. Similarly, this could be applied to a cube with the extrusions being smaller cubes. The only remaining platonic solid is the dodecahedron of which there is no clear polyhedron that could be extruded.

3 Programming

This project began by re-implementing Yuliya's Python/OpenGL code for the tetrahedron into Javascript/WebGL. David Eck's WebGL library glsim.jseck was used to help render the scene as it provides a similar syntax to OpenGL. The application can be run in any modern web browser with Javascript enabled.

The the main process calculates the midpoints of a triangle. This gives the inner triangle as well as the outter three. The inner triangle is to be removed and the process recursively iterates on the outter triangle. This is done up to the desired level and produces the vertices to be drawn later.

The drawing is done first by manually creating the original polyhedron. It then loops through the list of vertices created in the aforementioned process.

However, there is an error in the construction. In the original python code, the middle third of the face, or the base of the extruded tetrahedra, are only removed for the first level of recursion. Further extrusions on the extrusions have all four faces. This remains to be fixed and would require a significant rework of the recursive algorithm currently implemented while likely decreasing performance. There is no visual difference from outside the polyhedron, but there may be significant difference in what is seen inside if it were properly implemented.

The controls implemented use keyboard buttons in a similar fashion to Professor George Francis' other projects along with clickable buttons located beneath the display. A slider for the height of extrusions is added for further interactivity as well. The camera movement is implemented using "jazz track" provided by Professor Francis which creates smooth continuous movement with the cursors position.

4 Conclusions and Future Directions

The conjecture has been proven false using the icosahedron and octahedron constructions. The reason why the tetrahedron and cube seemingly converge to other polyhedra still remains hidden. Possible answers may lie in examining the dihedral angles of the shapes involved.

Further improvements could be made by implementing the process on the cube and dodecahedron. As mentioned in the programming section, there remains a flaw in the algorithm which only affects the visual appearance inside the object.

References

[1] http://new.math.uiuc.edu/math198/MA198-2013/semibra2/

[2] http://math.hws.edu/graphicsbook/source/glsim/glsim-doc.html