

The curvature and dimension of non-differentiable surfaces

S. Halayka*

September 19, 2019

Abstract

The curvature of a surface can lead to fractional dimension. In this paper, the properties of the 2-sphere surface of a 3D ball and the 2.x-surface of a 3D fractal set are considered. Tessellation is used to approximate each surface, primarily because the 2.x-surface of a 3D fractal set is otherwise non-differentiable.

1 Tessellation of surfaces

Approximating the surface of a three-dimensional shape via triangular tessellation (a mesh) allows us to calculate the surface's dimension D , somewhere between 2.0 and 2.9999.

First we calculate, for each triangle, the average dot product of the triangle's normal \hat{n}_i and its three neighbouring triangles' normals $\hat{o}_1, \hat{o}_2, \hat{o}_3$:

$$d_i = \frac{\hat{n}_i \cdot \hat{o}_1 + \hat{n}_i \cdot \hat{o}_2 + \hat{n}_i \cdot \hat{o}_3}{3}. \quad (1)$$

Because we assume that there are three neighbours per triangle, the mesh must be closed.

Then we calculate the normalized measure:

$$m_i = \frac{1.0 - d_i}{2.0}. \quad (2)$$

Once m has been calculated for all triangles, we can then calculate the average normalized measure x , where t is the number of triangles:

$$x = \frac{m_1 + m_2 + \dots + m_t}{t}. \quad (3)$$

The dimension of the surface is

$$D = 2.0 + x. \quad (4)$$

Analogous is the dimension of a line $D = 1.0 + x$, from 1.0 to 1.9999. See Figures 1 and 2.

*sjhalayka@gmail.com

In this paper, Marching Cubes [1] is used to generate the triangular tessellations. The full C++ code can be found at [2].

For a 2-sphere, the *local* curvature vanishes as the size of the triangles decreases. This results in a dimension of 2.0, which is to be expected from a non-fractal surface. See Figures 3, 4, and 5.

On the other hand, for the surface of a three-dimensional fractal set, the local curvature does not vanish. This results in a dimension greater than 2.0, but not equal to or greater than 3.0, which is to be expected from a fractal surface. See Figures 6, 7, 8, and 9.

As far as we know, this method of calculating the fractal dimension of a surface is novel.

References

[1] Bourke P. <http://paulbourke.net/geometry/polygonise/>

[2] Full C++ source code <https://github.com/sjhalayka/meshdim>



Figure 1: The average dot product of neighbouring line segments is $d_i = (0.0 + 0.0)/2 = 0.0$. This leads to a normalized measure of $m_i = (1.0 - d_i)/2.0 = 0.5$, which in turn leads to an average normalized measure of $x = 0.5$. The dimension is $D = 1.0 + x = 1.5$.

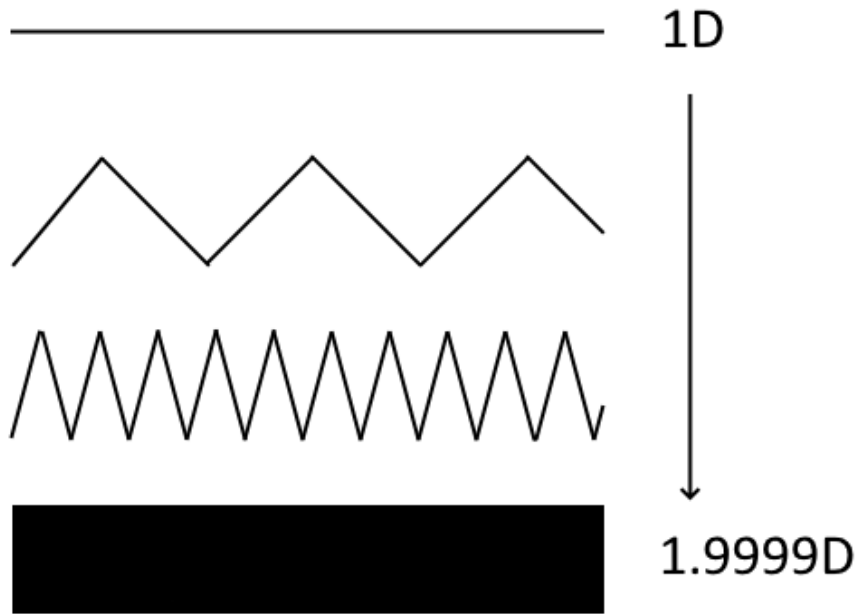


Figure 2: A line as it goes from dimension 1.0 (at top) to 1.9999 (at bottom). In the end, where the dimension is 1.9999, the result is practically a rectangle.



Figure 3: Low resolution surface for the iterative equation is $Z = Z^2$. The surface's dimension is 2.02.

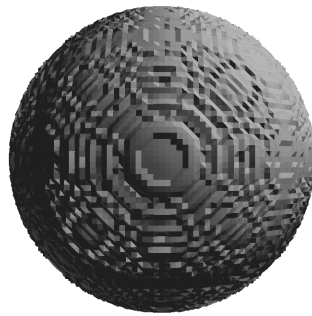


Figure 4: Medium resolution surface for the iterative equation is $Z = Z^2$. The surface's dimension is 2.06.

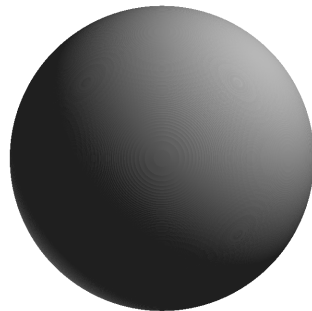


Figure 5: High resolution surface for the iterative equation is $Z = Z^2$. The surface's dimension is 2.0.

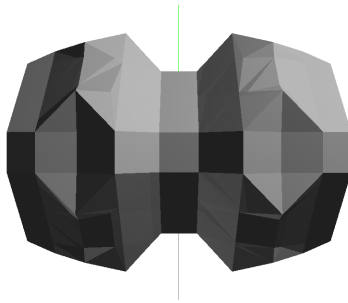


Figure 6: Low resolution surface for the iterative equation is $Z = Z \cos(Z)$. The surface's dimension is 2.05.

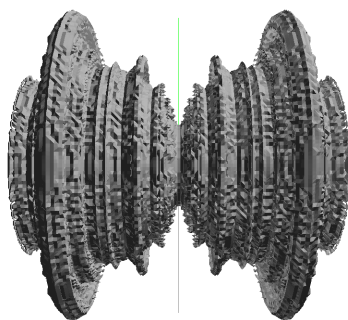


Figure 7: Medium resolution surface for the iterative equation is $Z = Z \cos(Z)$. The surface's dimension is 2.11.

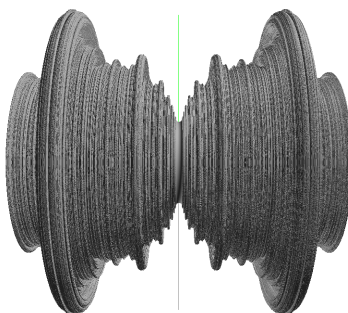


Figure 8: High resolution surface for the iterative equation is $Z = Z \cos(Z)$. The surface's dimension is 2.08.

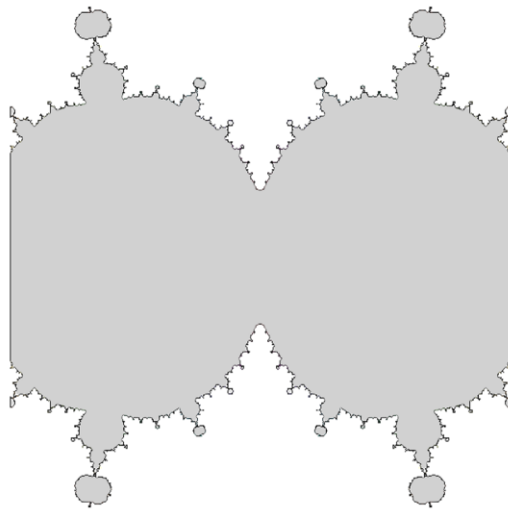


Figure 9: A two-dimensional slice of $Z = Z \cos(Z)$, showing the fractal nature of the set.