Chaos, Solitons and Fractals 42 (2009) 504–514

Contents lists available at ScienceDirect

Chaos, Solitons and Fractals

journal homepage: www.elsevier.com/locate/chaos

Approximating the connectedness of 3D quaternion Julia sets via isosurface polygonization

S. Halayka

Regina, SK Canada

ARTICLE INFO

Article history:
Accepted 19 January 2009

ABSTRACT

Approximations of the surface area, volume, specific surface area, and connectedness of 3D quaternion Julia sets are examined as various set parameters are run (e.g. C_x, lattice resolution).

1. Introduction

As discussed in [1], a 3D scalar field of quaternion magnitudes (e.g. $|Z|$) results from calculating a 3D quaternion Julia set [2–5] when using a finite 3D lattice of regularly spaced points.

Moving beyond the scope of [1], this magnitude field can then be used in conjunction with an isosurface polygonization algorithm [6–13] in order to approximate a set's surface area. For instance, the implementation discussed here uses Lorensen–Cline marching cubes, with a scalar domain of $0 \leqslant s \leqslant 1$ and an isovalue of $s_{surface} = 0.5$. In order to accommodate for this limited domain, each magnitude is truncated to twice the threshold value ($2t$), normalized, and then inverted before being passed into the isosurface polygonization algorithm as

$$s = \begin{cases} 1 - \frac{|Z|}{2t} & \text{if } |Z| < 2t, \\ 0 & \text{if } |Z| \geqslant 2t. \end{cases} \tag{1}$$

As such,

- $0 \leqslant |Z| < t$ is transformed into $1 \geqslant s > 0.5$, representing a point enclosed by the surface of a set.
- $|Z| = t$ is transformed into $s = 0.5$, representing a point along the surface of a set.
- $t < |Z| \leqslant 2t$ is transformed into $0.5 > s \geqslant 0$, representing a point outside of the surface of a set.

A set's total surface area A is approximated as the sum of the surface area of all triangles [14] produced by the entire set of marching cubes used during isosurface polygonization. Where $T_{x,y}$ represents the xth triangle's yth vertex

$$A = \sum_{x=1}^{total} \frac{1}{2} |(T_{x,1} - T_{x,2}) \times (T_{x,1} - T_{x,3})|. \tag{2}$$

Where e_{min}, e_{max} describe the spatial extent of the lattice, and ψ is the lattice resolution (a positive integer), the distance between each subsequent point along the lattice is

$$\delta = \frac{e_{max} - e_{min}}{\psi - 1}. \tag{3}$$

E-mail address: shalayka@gmail.com

0960-0779/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.chaos.2009.01.014

Where n_{set} is the number of points within a set (post-iteration, $0 \leqslant n_{set} \leqslant \psi^3$), a set's total volume V is approximated as

$$V = \delta^3 n_{set}. \tag{4}$$

Where $V \neq 0$ and $A \neq 0$, a set's specific surface area SA and connectedness σ are

$$SA = \frac{A}{V}, \tag{5}$$

$$\sigma = \frac{1}{SA} = \frac{V}{A}. \tag{6}$$

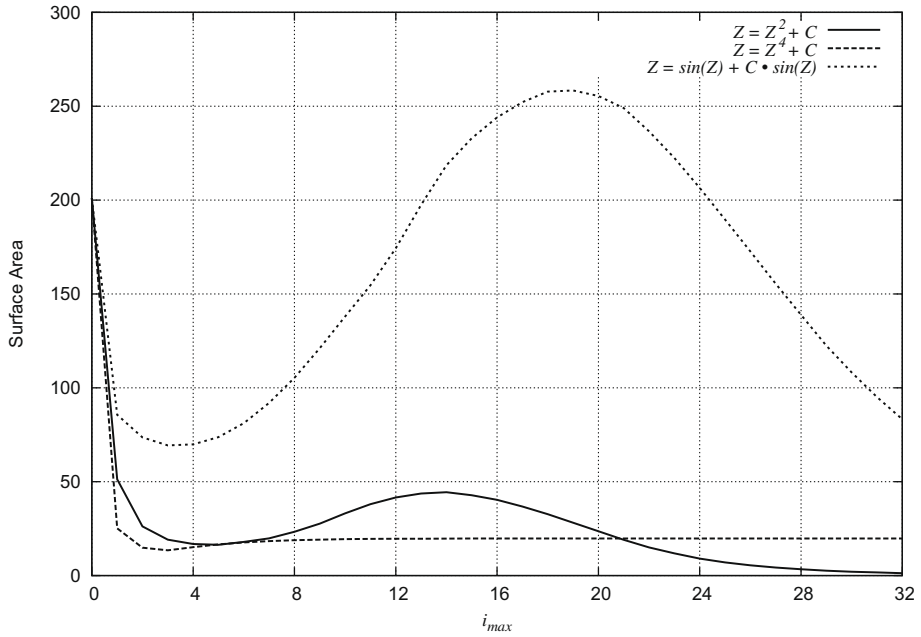

Fig. 1. Surface area for several iterative functions, where $t = 4,\ i_{max} = variable,\ Z_w = 0,\ C_{xyzw} = 0.3, 0.5, 0.4, 0.2,\ e_{min} = -4,\ e_{max} = 4$, and $\psi = 256$.

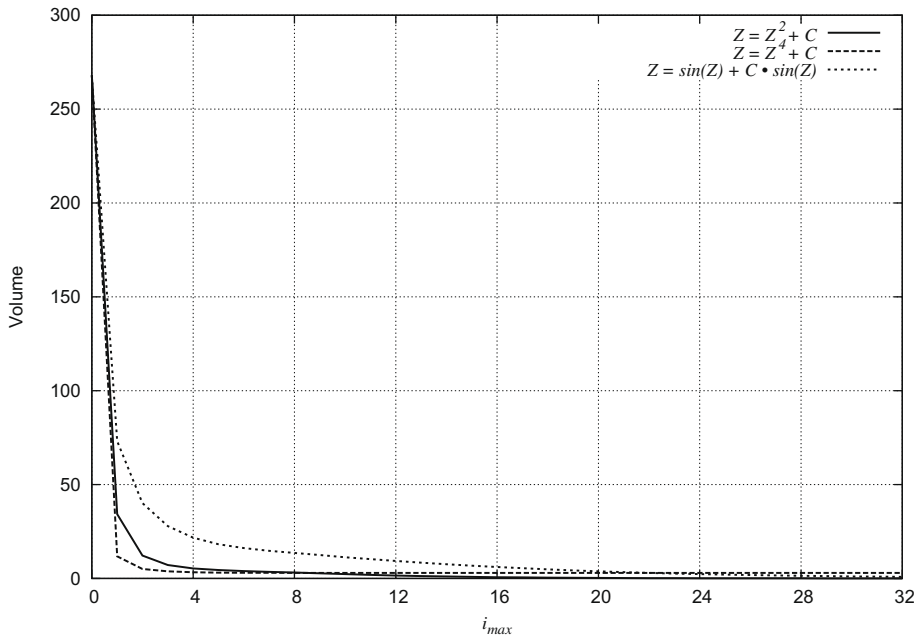

Fig. 2. Volume for several iterative functions, where $t = 4,\ i_{max} = variable,\ Z_w = 0,\ C_{xyzw} = 0.3, 0.5, 0.4, 0.2,\ e_{min} = -4,\ e_{max} = 4$, and $\psi = 256$.

The concept of specific surface area and connectedness [3,15,16] may be visualized by cutting a solid 3D object (e.g. a block of wood) in half, and then separating the two parts. The amount of solid substance (volume) remains roughly constant, but the surface area increases. Accordingly, the specific surface area also increases, ultimately accompanied by a decrease in connectedness. From this perspective, it is seen that specific surface area is synonymous with both surface complexity and disconnectedness.

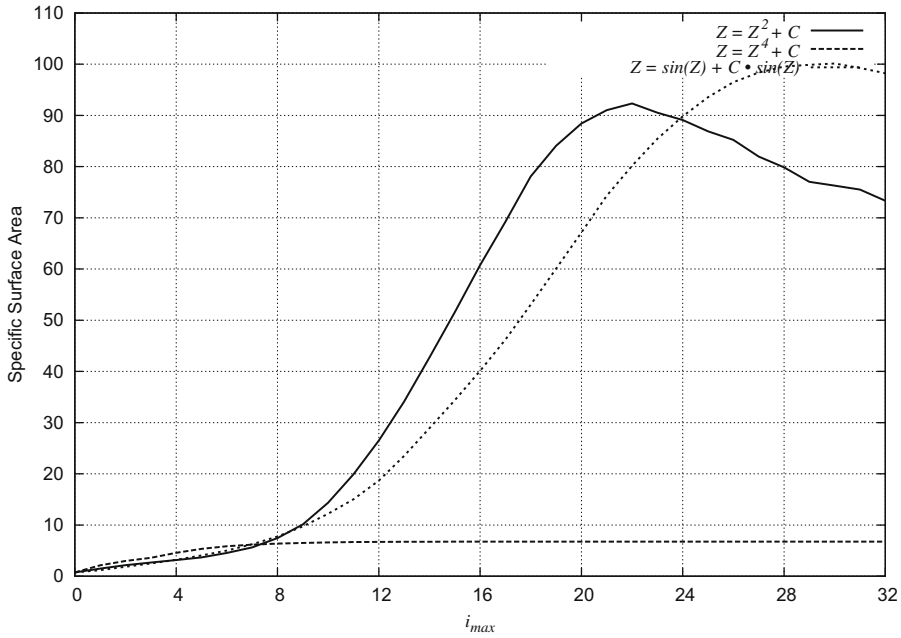

Fig. 3. Specific surface area for several iterative functions, where $t = 4,\ i_{max} = variable,\ Z_w = 0,\ C_{xyzw} = 0.3, 0.5, 0.4, 0.2,\ e_{min} = -4,\ e_{max} = 4$, and $\psi = 256$.

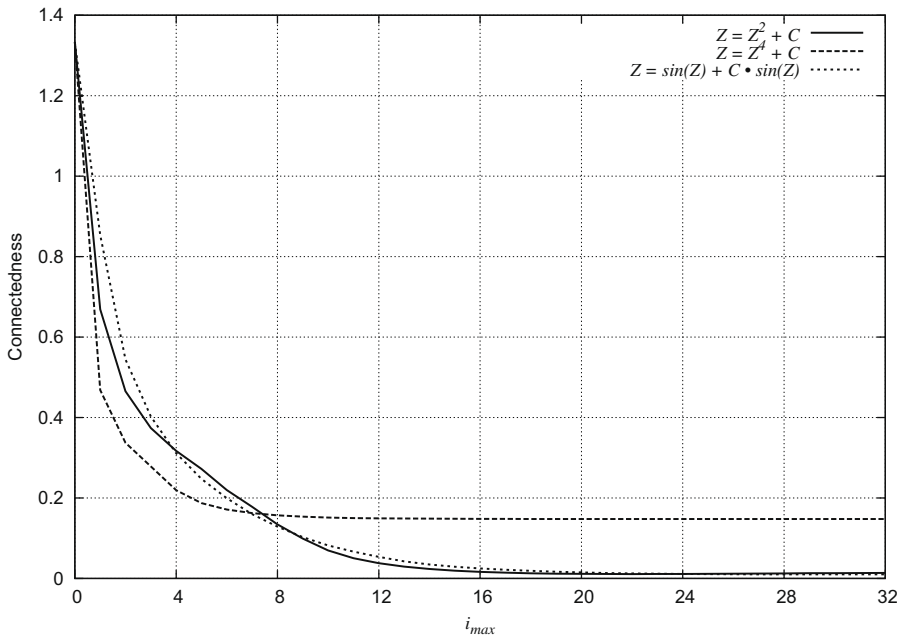

Fig. 4. Connectedness for several iterative functions, where $t = 4,\ i_{max} = variable,\ Z_w = 0,\ C_{xyzw} = 0.3, 0.5, 0.4, 0.2,\ e_{min} = -4,\ e_{max} = 4$, and $\psi = 256$.

2. Sets that form a solid 3D ball

A demonstration of the surface area and volume approximations given here may be started by analyzing the special sets produced where the maximum number of iterations is $i_{max} = 0$. Since no iterations are performed, these special sets are independent of the iterative function (e.g. $Z = Z^2 + C$), are always in the form of a solid 3D ball centred at the origin, and have a radius determined solely by the threshold and Z_w parameters

$$r = \begin{cases} \sqrt{t^2 - |Z_w|^2} & \text{if } |Z_w| < t, \\ 0 & \text{if } |Z_w| \geqslant t. \end{cases} \tag{7}$$

As pointed out by Douglas Sweetser [17], this equation is also inherent to the theory of special relativity [18], where t stands in for the uniform speed of light in vacuum and $|Z_w|$ stands in for the uniform speed of a massive body.

Given that the surface area and volume of a solid 3D ball [14] are

$$A = 4\pi r^2, \tag{8}$$

$$V = \frac{4}{3}\pi r^3, \tag{9}$$

the corresponding specific surface area and connectedness simplify to

$$SA = \frac{3}{r}, \tag{10}$$

$$\sigma = \frac{r}{3}. \tag{11}$$

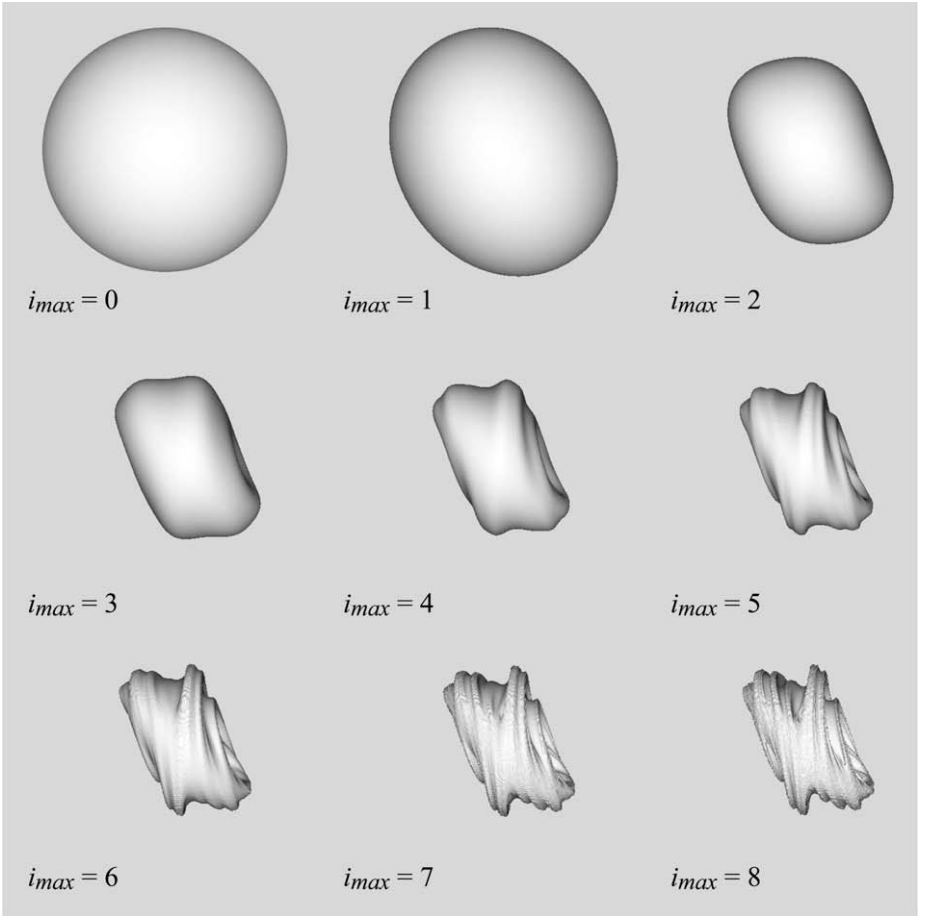

Fig. 5. Meshes of $Z = Z^2 + C$, where $t = 4$, $i_{max} = variable$, $Z_w = 0$, $C_{xyzw} = 0.3, 0.5, 0.4, 0.2$, $e_{min} = -4$, $e_{max} = 4$, and $\psi = 512$. The top-left mesh shown here (the surface of a solid 3D ball) has been scaled down from actual size in order to fit within this image. All remaining meshes are of the same rotational orientation and scale.

For example, where $i_{max} = 0$, $t = 4$, and $Z_w = 0$, a solid 3D ball of $r = 4$ is produced, where $A \approx 201$, $V \approx 268$, $SA = 3/4 = 0.75$, and $\sigma = 4/3 \approx 1.333$.

In order to guarantee that a solid 3D ball is produced in its entirety, but without sacrificing resolution unnecessarily, its radius can be used to specify the spatial extent of the lattice

$$e_{min} = -r, \tag{12}$$

$$e_{max} = r. \tag{13}$$

As expected, the surface area of a solid 3D ball is approximated well by isosurface polygonization [8] when using reasonably large lattice resolutions (e.g. $\psi \geqslant 128$).

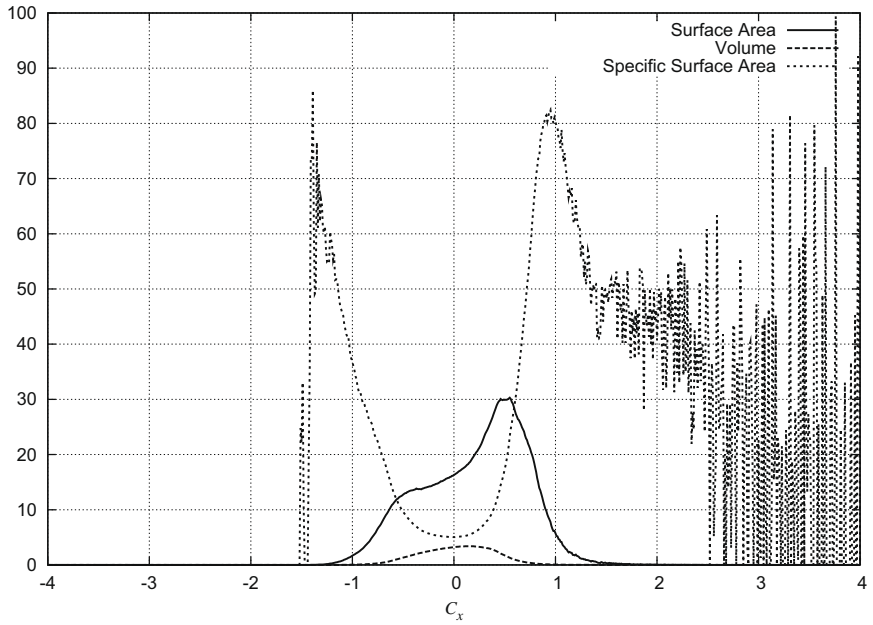

Fig. 6. Surface area, volume, and specific surface area for $Z = Z^2 + C$, where $t = 4$, $i_{max} = 8$, $Z_w = 0$, $C_{xyzw} = variable, 0.5, 0.4, 0.2$, $e_{min} = -4$, $e_{max} = 4$, and $\psi = 256$.

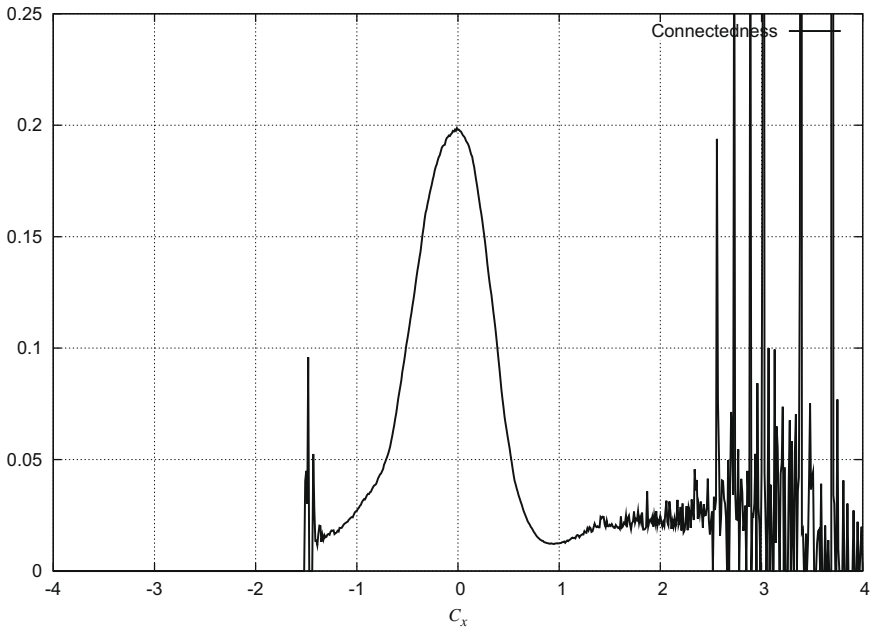

Fig. 7. Connectedness for $Z = Z^2 + C$, where $t = 4$, $i_{max} = 8$, $Z_w = 0$, $C_{xyzw} = variable, 0.5, 0.4, 0.2$, $e_{min} = -4$, $e_{max} = 4$, and $\psi = 256$.

3. Sets that deviate from a solid 3D ball

Now that the properties of a solid 3D ball produced by $i_{max} = 0$ have been outlined, they can be readily used as a baseline reference for comparing the properties of sets produced by any given iterative function where $i_{max} > 0,\ C \neq 0, 0, 0, 0$.

See Figs. 1–4 for plots of the evolving set properties produced by three different iterative functions as i_{max} is varied from 0 to 32. See Fig. 5 for an accompanying visualization of $Z = Z^2 + C$ as i_{max} is varied from 0 to 8.

See Figs. 6 and 7 for a plot of the evolving set properties produced by the iterative function $Z = Z^2 + C$ as C_x is varied from -4 to 4. See Fig. 8 for an accompanying visualization of $Z = Z^2 + C$ as C_x is varied from -1 to 1. It is important to note (both analytically and visually) that not only does the set's volume reduce as the absolute value of C_x increases, but that the set also becomes increasingly disconnected.

It is taken that the large spikes near the edges of the specific surface area and connectedness plots in Figs. 6 and 7 are due to a limitation inherent to the simple approximation of volume given here. That is, the approximated surface area is allowed to vanish in a continuous fashion, but the approximated volume is not. It seems that in the case of sets which are sparse and disconnected that this produces an approximated volume that is disproportionately large for the respective approximated surface area, resulting in a measure of connectedness that is much larger than expected. However, this manuscript was intentionally designed to focus only on core concepts, and so the implementation of more sophisticated methods of volume approximation (e.g. directly from the polygon mesh [19,20]) are left as future work.

One other aspect of the approximation that deserves attention is how the set properties change with increasing lattice resolution (e.g. $dA/d\psi$, etc.). This gives heed to Mandelbrot's original question [21,22] of 'how long is the coast of Britain'? Here, the question is reworded as 'how long and wide is the surface of a 3D quaternion Julia set' when approximated via isosurface polygonization?

See Figs. 9–12 for plots of the set properties produced by the iterative functions $Z = Z^2 + C$ and $Z = \sinh(Z^3) + C$ as the set parameters and lattice extent are held constant while the lattice resolution is varied from 1024 to 4096. Fig. 13 visualizes the difference in polygon mesh complexity for $\psi = 8$ and $\psi = 64$. It is important to note (both analytically and visually) that the

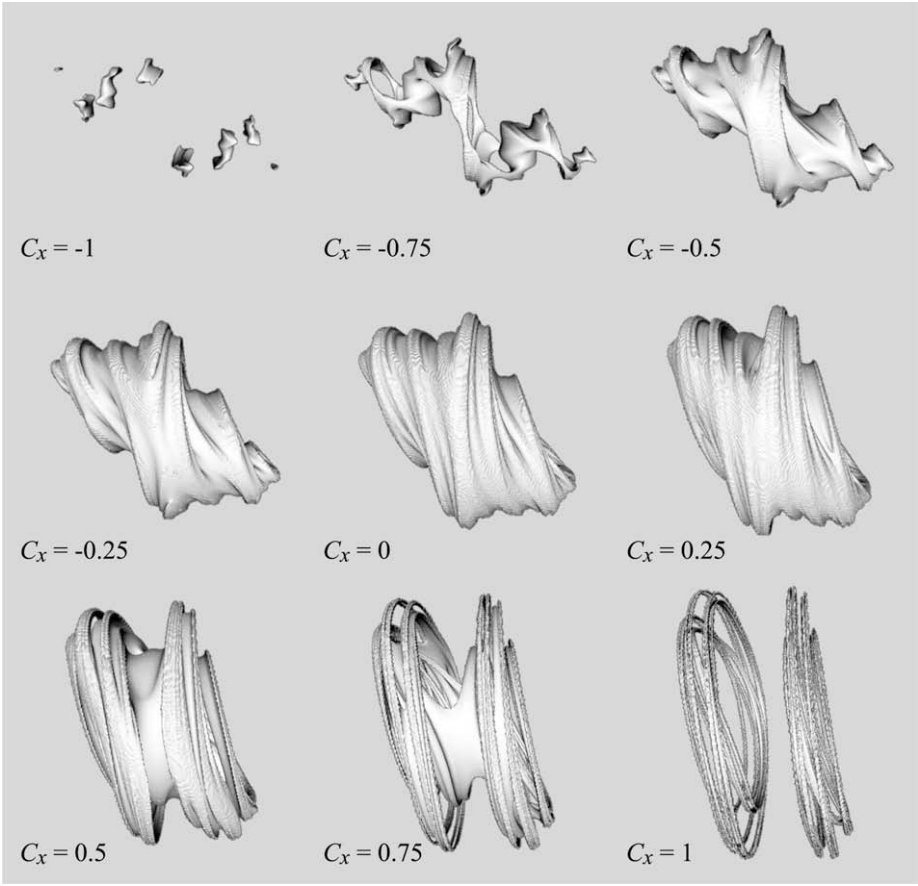

Fig. 8. Meshes of $Z = Z^2 + C$, where $t = 4,\ i_{max} = 8,\ Z_w = 0,\ C_{xyzw} = variable, 0.5, 0.4, 0.2,\ e_{min} = -4,\ e_{max} = 4$, and $\psi = 768$. All meshes shown here are of the same rotational orientation and scale.

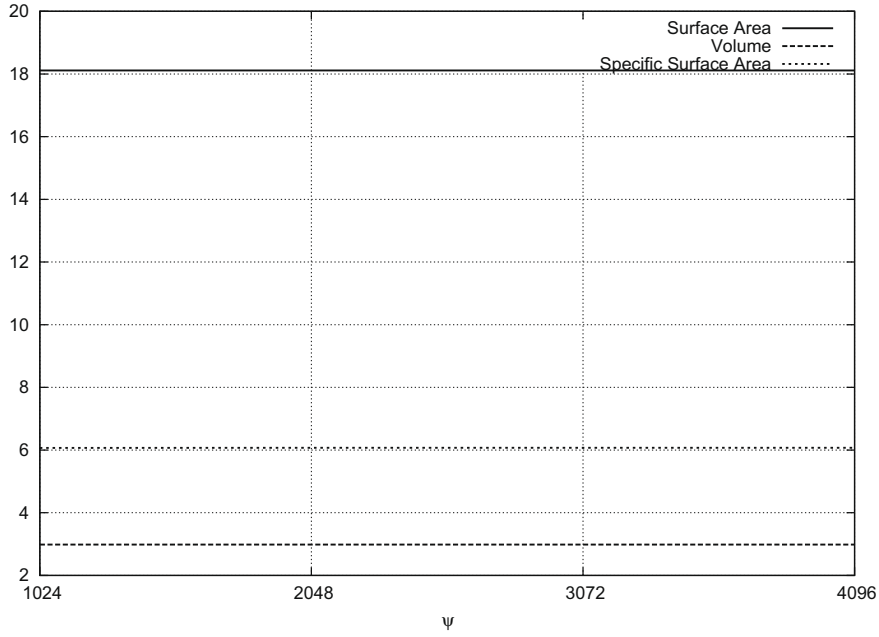

Fig. 9. Surface area, volume, and specific surface area for $Z = Z^2 + C$, where $t = 4,\ i_{max} = 8,\ Z_w = 0,\ C_{xyzw} = 0.3, 0.5, 0.4, 0.2,\ e_{min} = -1,\ e_{max} = 1$, and $\psi = variable$.

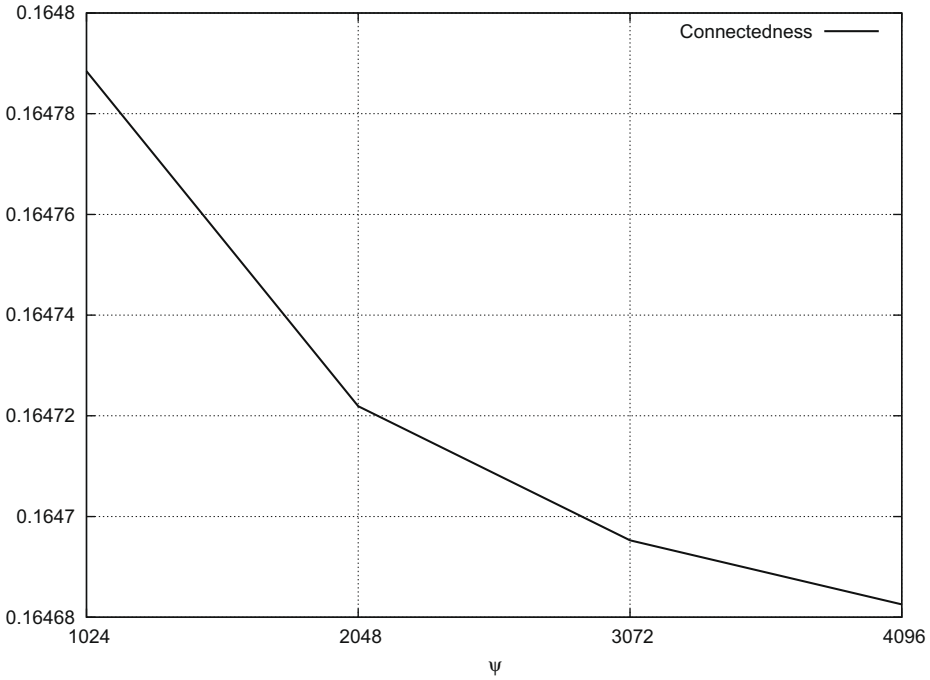

Fig. 10. Connectedness for $Z = Z^2 + C$, where $t = 4,\ i_{max} = 8,\ Z_w = 0,\ C_{xyzw} = 0.3, 0.5, 0.4, 0.2,\ e_{min} = -1,\ e_{max} = 1$, and $\psi = variable$.

sets produced by $Z = \sinh(Z^3) + C$ are much more inherently disconnected than the sets produced by $Z = Z^2 + C$, even though they all share the same set and lattice parameters.

As expected, some of the set properties produced by $i_{max} > 0,\ C \neq 0, 0, 0, 0$ are found to change drastically as lattice resolution increases (e.g. volume is expected to remain roughly constant, while surface area is not). Although the change related to $Z = Z^2 + C$ is all but negligible at this particular range of lattice resolutions, the change related to $Z = \sinh(Z^3) + C$ is much

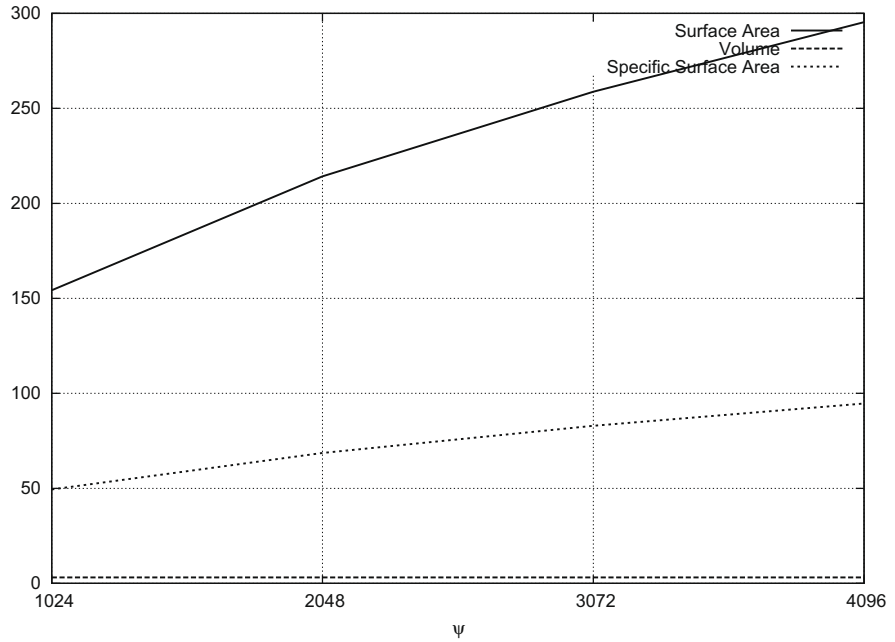

Fig. 11. Surface area, volume, and specific surface area for $Z = \sinh(Z^3) + C$, where $t = 4,\ i_{max} = 8,\ Z_w = 0,\ C_{xyzw} = 0.3, 0.5, 0.4, 0.2,\ e_{min} = -1,\ e_{max} = 1$, and $\psi = variable$.

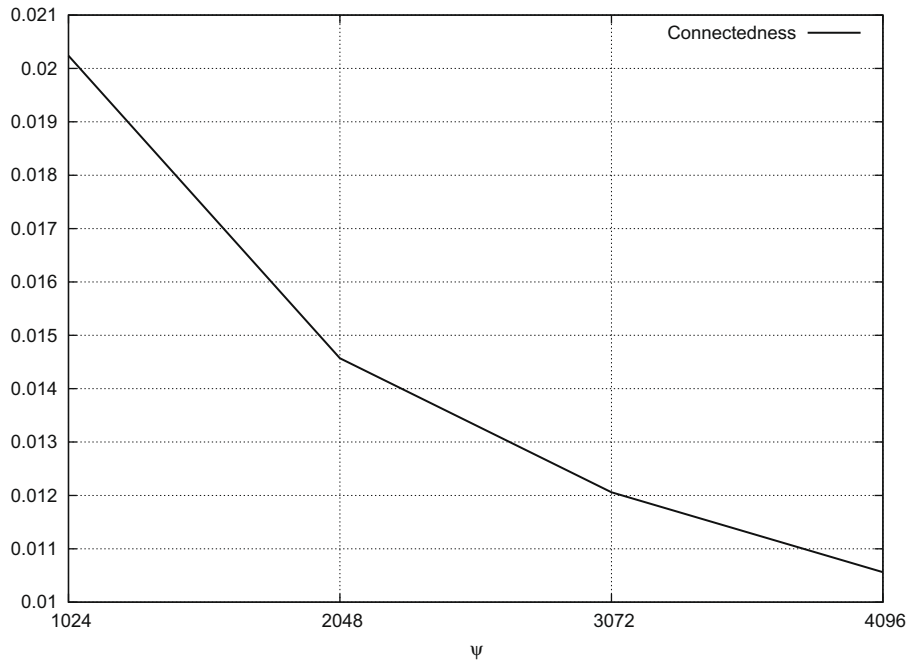

Fig. 12. Connectedness for $Z = \sinh(Z^3) + C$, where $t = 4,\ i_{max} = 8,\ Z_w = 0,\ C_{xyzw} = 0.3, 0.5, 0.4, 0.2,\ e_{min} = -1,\ e_{max} = 1$, and $\psi = variable$.

more pronounced (e.g. surface area nearly doubled in value). This is quite unlike the case of a solid 3D ball, where all of the set properties are expected to remain settled upon their respective constant values (e.g. $A \approx 201$) even as lattice resolution is increased without limit. In fact, it is this difference that draws the distinction between fractal (e.g. $i_{max} > 0,\ C \neq 0, 0, 0, 0$) and traditional (e.g. $i_{max} = 0$) geometry, insomuch that the surface of a 3D quaternion Julia set is expected to present new scale-dependent complexity via disconnectedness while undergoing zoom (e.g. reduced lattice extent, or increased resolution), but the surface of a solid 3D ball is not.

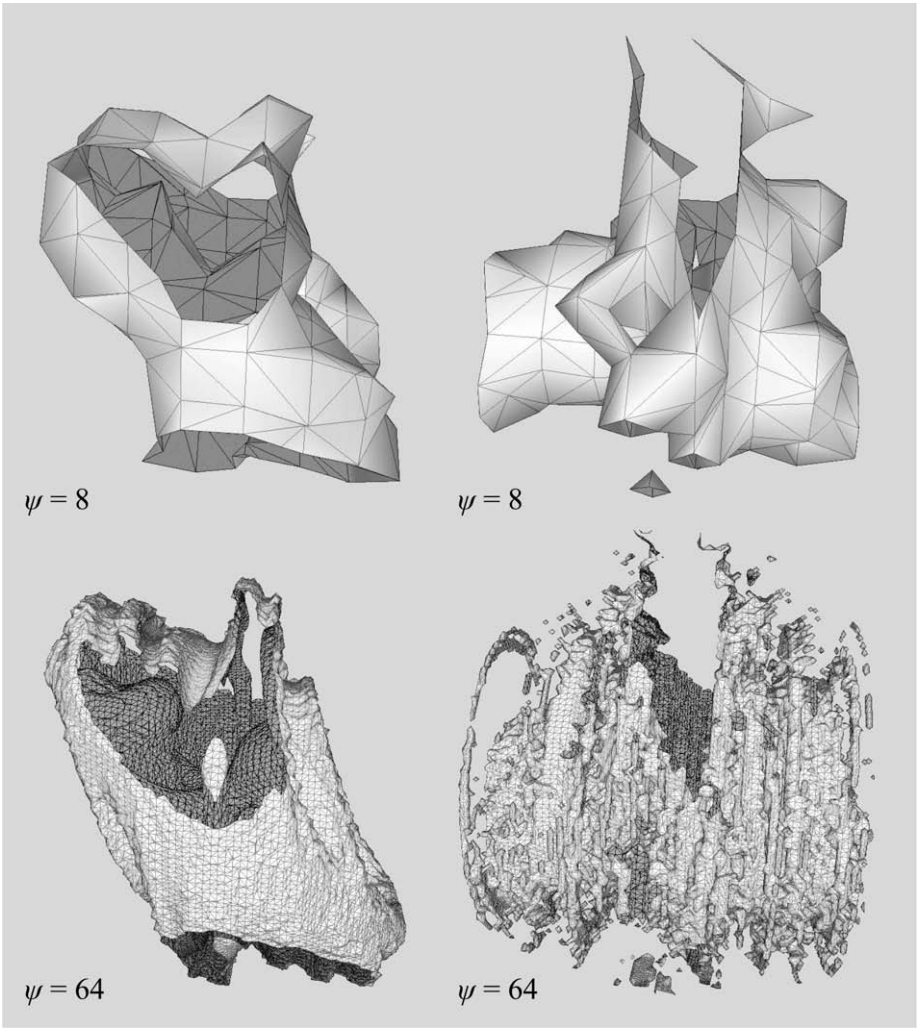

Fig. 13. Meshes with wireframes of $Z = Z^2 + C$ (left) or $Z = \sinh(Z^3) + C$ (right), where $t = 4,\ i_{max} = 8,\ Z_w = 0,\ C_{xyzw} = 0.3, 0.5, 0.4, 0.2,\ e_{min} = -1,\ e_{max} = 1$, and $\psi = 8$ (top) or $\psi = 64$ (bottom). All meshes shown here are of the same rotational orientation and scale. All meshes shown here are clipped by the finite lattice extent, revealing extra surface detail (e.g. dark backfaces, wiggly wireframe edges along the clipping planes).

4. Using local surface normal disparity to identify potential regions of visual interest

The relationship between curvature and disconnectedness can be demonstrated in a straightforward way by the definitively non-fractal solid 3D ball. For instance, mean curvature H [23] and disconnectedness (e.g. specific surface area) both vanish hand-in-hand as a solid 3D ball's radius is increased without limit:

$$H = \frac{SA}{3} = \frac{1}{r}. \tag{14}$$

It seems then that even though the concept of curvature is not directly applicable to the actual (non-approximated) surface of a 3D quaternion Julia set (it is non-differentiable [24]), it is still reasonable to expect that the highly curved local regions of a set's *approximated* surface are the regions most likely to present new scale-dependent complexity via disconnectedness while undergoing zoom. That said, a simple method for identifying these highly curved local regions is given here.

In terms of the implementation used here, the number of marching cubes required to polygonize a 3D quaternion Julia set's entire magnitude field is

$$n_{mc} = (\psi - 1)^3. \tag{15}$$

The centre points of the entire set of marching cubes also form their own regularly spaced 3D lattice, but with a reduced resolution ψ_{mc} and reduced spatial extent e_{mcmin}, e_{mcmax} of

$$\psi_{mc} = \psi - 1, \tag{16}$$

$$e_{mcmin} = e_{min} + \frac{\delta}{2}, \tag{17}$$

$$e_{mcmax} = e_{max} - \frac{\delta}{2}. \tag{18}$$

This secondary lattice may then be assigned scalar values, forming a 3D scalar field that is suitable for producing a mesh via isosurface polygonization in the usual fashion. The foremost use for this secondary mesh is to enclose the primary mesh's highly curved local regions, thus identifying them as potential targets for further manual investigation.

A crude approximation of the local curvature k of the hth marching cube is produced by comparing the unit surface normals $\vec{N}$ of each unique pair of triangles associated with h (e.g. T_i, T_j) via dot product

$$k_h = \sum_{i=1}^{(total_h - 1)} \sum_{j=i+1}^{total_h} \frac{1 - \vec{N}_i \cdot \vec{N}_j}{2}. \tag{19}$$

And so, as the amount of disparity amongst the direction of the local surface normals increases, so does the approximated measure of local curvature. For instance, a single pair of triangles forming a flat plane (identical surface normal direction) would provide a measure of $k_h = 0$, but a single pair of triangles forming a ridge or valley of negligible thickness (opposing surface normal direction) would provide a measure of $k_h \approx 1$.

However, since a marching cube may contain more than a single pair of triangles, all k_h are then re-normalized using the global maximum k_{max}, so that the isosurface polygonization algorithm domain is limited to $0 \leqslant s \leqslant 1$, as was preferred in Section 1:

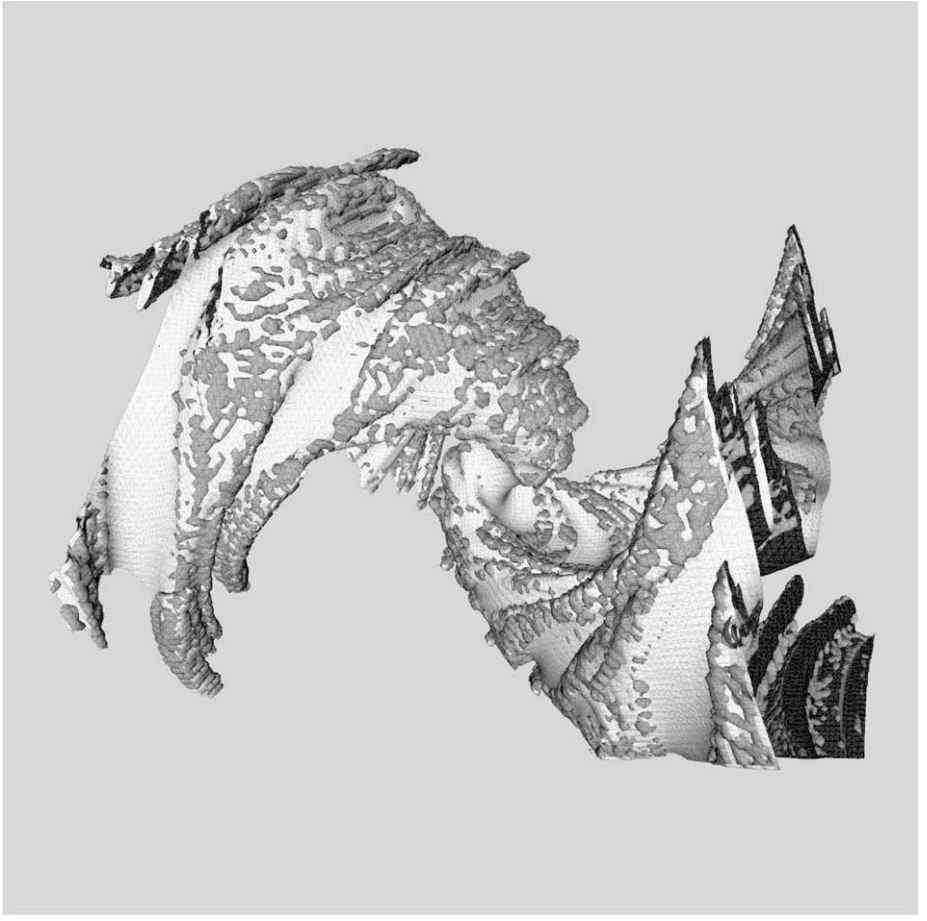

Fig. 14. Mesh with wireframe of $Z = \sin Z + C \cdot \sin Z$, where $t = 4,\ i_{max} = 8,\ Z_w = 0,\ C_{xyzw} = 0.3, 0.5, 0.4, 0.2,\ e_{min} = -1,\ e_{max} = 1$, and $\psi = 128$. The mesh is clipped by the finite lattice extent, revealing extra surface detail. As usual, an isovalue of $s_{surface} = 0.5$ was used to produce the white primary mesh. However, an arbitrary isovalue of $s_{surface} = 0.01$ was used to produce the gray secondary mesh. The volumetric character of the secondary mesh is demonstrated by the fact that it also occludes some of the primary mesh's dark backfaces.

$$s_h = \frac{k_h}{k_{max}}. \tag{20}$$

The result of this approximation of local curvature is visualized in Fig. 14. It is important to note that the volume defined by the gray secondary mesh encloses the highly curved local regions of the white primary mesh well enough, ultimately avoiding the flatter local regions. Of course a better, more complex approximation of local curvature would take into account triangle-based adjacency (edge sharing) rather than marching cube volume-based adjacency, but is seemingly unnecessary for this simple demonstration of core concepts.

5. Notes

The ISO standard C++ source code used to generate the meshes and set property data for the figures given here is public domain [25]. The portion that implements the marching cubes algorithm relies on the lookup tables and some of the public domain code presented by Paul Bourke and Cory Bloyd [8].

Future work related to specific surface area and connectedness in general will focus on the analysis of the various scalar components of the various field types produced in computational fluid dynamics [26–28] and general relativity [29].

It should also prove interesting to explore the specific surface area and connectedness produced by the 4-volume of the quaternion Julia set (e.g. where Z_w is no longer an arbitrary constant) and various different projections of it into 3D (e.g. the projected 3-volume would serve as the 'surface area'). Not only would this illustrate the properties of the sets, but also the properties of each different projection method. As suggested by Godwin Vickers [30], Hopf fibration should be a good starting point for this line of investigation.

Acknowledgements

Thank you to B. Mandelbrot, A. Naït-Merzouk, and A. Norton for correspondence regarding the surface of 3D quaternion Julia sets. Thank you to M.S. El Naschie for his words of encouragement, and for guiding my attention toward some of the works referenced here. Special thanks to H. Godai and S. Uesaka for their inspiring work related to lattices and polygon meshes.

References

[1] Halayka S. Some visually interesting non-standard quaternion fractal sets. Chaos, Solitons & Fractals 2009;41(5):2842–6.
[2] Norton A. Generation and display of geometric fractals in 3-D. In: SIGGRAPH'82: Proceedings of the 9th annual conference on computer graphics and interactive techniques; 1982. ISBN: 0-89791-076-1.
[3] A. Douady, J.H. Hubbard. On the dynamics of polynomial-like mappings. Annales scientifiques de l'École Normale Supérieure Sér.4, 18, no. 2; 1985, p. 287–343.
[4] Mandelbrot BB. The fractal geometry of nature; 1983. ISBN: 978-0716711865.
[5] Tan L, et al. The Mandelbrot set, theme and variations; 2000. ISBN: 978-0521774765.
[6] Lorensen WE, Cline HE. Marching cubes: a high resolution 3D surface construction algorithm. ACM SIGGRAPH Comput Graph 1987;21(4).
[7] Bloomenthal J. Polygonization of implicit surfaces. Comput Aided Geom Des 1988.
[8] Bourke P. Polygonising a scalar field. <http://local.wasp.uwa.edu.au/pbourke/geometry/polygonise/index.html/>; 1994.
[9] Osher SJ, Fedkiw RP. Level set methods and dynamic implicit surfaces; 2002. ISBN: 978-0387954820.
[10] Lewiner T, Lopes H, Wílson Vieira A, Tavares G. Efficient implementation of Marching Cubes' cases with topological guarantees. J Graph Tools 2003;8(2):1–15.
[11] Martin B. Quaternion fractal explorer for Linux. <http://www.theory.org/software/qfe/>; 1997.
[12] Gintz TW. QuaSZ (Quad Surface Zplot) for windows. <http://www.mysticfractal.com/QuaSZ.html/>; 2000.
[13] Dey TK, Levine JA. Delaunay meshing of isosurfaces. Visual Comput 2008;24(6).
[14] Gullberg J. Mathematics: from the birth of numbers; 1997. ISBN: 0-393-04002-X.
[15] Precision Agriculture Team, Department of Soil Science at North Carolina State University. Soil Geometry. <http://courses.soil.ncsu.edu/resources/physics/texture/soilgeo.swf/>; 2008.
[16] Packham DE. Handbook of adhesion, 2nd ed.; 2005. ISBN: 9780471808749.
[17] Sweetser D. Personal communication; 2007.
[18] Einstein A. On the electrodynamics of moving bodies. Ann Phys 1905;17.
[19] Schneider P, Eberly DH. Geometric tools for computer graphics; 2002. ISBN: 978-1558605947.
[20] Llamas I. Real-time voxelization of triangle meshes on the GPU. SIGGRAPH'07: ACM SIGGRAPH 2007 sketches; 2007.
[21] Mandelbrot BB. How long is the coast of Britain? Statistical self-similarity and fractional dimension. Science 1967;156(3775):636–8.
[22] Normant F, Tricot C. Method for evaluating the fractal dimension of curves using convex hulls. Phys Rev A 1991;43:6518–25.
[23] Gray A. Modern differential geometry of curves and surfaces with mathematica, 2nd ed.; 1997. ISBN: 978-0849371646.
[24] Cochran WO, Lewis RR, Hart JC. The normal of a fractal surface. Visual Comput 2001;17(4).
[25] Halayka S. Quaternion Julia set surface area and volume plotter, C++ code. <http://qjssurfareavolume.googlecode.com/files/ssa_pregpu3.zip/>; 2008.
[26] Stam J. Stable Fluids. SIGGRAPH 99 conference proceedings, Annual conference series; 1999. p. 121–8.
[27] Harris MJ. Fast fluid dynamics simulation on the GPU. GPU Gems: Programming techniques, tips, and tricks for real-time graphics; 2004. ISBN: 978-0321228321.
[28] Schmidt W. The numerical simulation of turbulence. Available from: arXiv:0712.0954v1 [astro-ph]; 2007.
[29] Misner CW, Thorne KS, Wheeler JA. Gravitation; 1973. ISBN: 0716703440.
[30] Vickers G. Personal communication; 2004.