Calculating Julia fractal sets in any embedding dimension

Ricardo Fariello[1], Paul Bourke[2], and João P. Lopes[1]

[1]Universidade Estadual de Montes Claros, Montes Claros, Brazil
[2]The University of Western Australia, Perth, Australia

Friday 1st April, 2022

Abstract

In this paper we compute and display hyperdimensional Julia sets using a multiplication operator that can be applied to any embedding dimension. Special attention is given to 5D Julia sets, which are visualized in 3D through a voxel-based representation and volumetric ray casting rendering.

Keywords: Fractals, Hypercomplex Julia sets, Visualization, Voxel.

1 Introduction

The Julia set traces its origins back to the early twentieth century when G. Julia published his pioneering work on the iteration of polynomials and rational functions [1]. The Julia set is traditionally associated with the polynomial $P(z) = z^2 + c$. More generally the Julia set can be defined as the locus for which the orbits of $P_m(z)$ are bounded (P_m is the mth iterate of P). This locus is often a fractal.

Subsequent research was primarily focused on analytical aspects of the set. This changed in the 1980s when there was a renewed interest arising from revolutionary new visualisations of the Julia set starting with the works of C. A. Pickover [2], A. Norton [3, 4] and J. A. R. Holbrook [5, 6]. Up to this point the quaternion Julia set had been largely unexplored until J. C. Hart et al. introduced the use of ray tracing [7] as an effective means of visualizing these 4D fractals.

In the 1990s, the octonion Julia set was explored by C. J. Griffin and G. C. Joshi [8], and by S. L. Dixon et al. [9]. Afterwards, J. Cheng and J.-r. Tan [10], along with X.-Y. Wang and Y.-Y. Sun [11], generalized the quaternion Julia set, and X. Wang and T. Jin [12] extended these works to include all embedding dimensions $n \geq 1$ via doubling and truncation techniques. In the meantime S. Halayka [13] developed definitions of various nonstandard quaternion Julia sets. Other generalizations of the Julia set, that have inspired the work presented here, have been described by A. Katunin [14, 15].

Historically those studying the Julia set have focused to the basic number spaces: real, complex, quaternion, octonion, etc. For these number spaces an accurate construction of the Julia set is possible, but the deterioration of the properties of the algebras of such spaces could be viewed as a negative factor. For example, quaternions are not commutative under multiplication, while octonions are neither commutative or associative. The 16D algebra of sedenions is weakly associative [16], which makes it difficult, but not impossible, to formulate a procedure for building fractal sets. This is largely due to key properties of the multiplication operation being lost. Further, owing to the lack of a well-defined multiplication operator for intermediate dimensions, namely 3D, 5D, 6D, and 7D, fractal sets cannot be fully constructed following the current approaches. A deeper exploration of the standard Julia set and its hypercomplex higher dimensional analogues is given in the book of A. Katunin [17].

Here we will present examples in the 5D case but we could have addressed any other number of dimensions with $n \geq 1$. This includes n not necessarily being equal to a power of two (not necessarily 1, 2, 4, 8, etc). Due to the missing multiplication operator in the 5D number type, we will propose a new multiplication operator.

2 Traditional multiplication and truncation

For an octonion C we will write the 8 components as (C_0,C_1,C_2,C_3,C_4,C_5,C_6,C_7). Given two real octonions A and B, componentwise their product $C = AB$ can be expanded as follows:

$$
\begin{aligned}
C_0 &= A_0B_0 - A_1B_1 - A_2B_2 - A_3B_3 - A_4B_4 - A_5B_5 - A_6B_6 - A_7B_7,\\
C_1 &= A_0B_1 + A_1B_0 + A_2B_3 - A_3B_2 + A_4B_5 - A_5B_4 - A_6B_7 + A_7B_6,\\
C_2 &= A_0B_2 - A_1B_3 + A_2B_0 + A_3B_1 + A_4B_6 + A_5B_7 - A_6B_4 - A_7B_5,\\
C_3 &= A_0B_3 + A_1B_2 - A_2B_1 + A_3B_0 + A_4B_7 - A_5B_6 + A_6B_5 - A_7B_4,\\
C_4 &= A_0B_4 - A_1B_5 - A_2B_6 - A_3B_7 + A_4B_0 + A_5B_1 + A_6B_2 + A_7B_3,\\
C_5 &= A_0B_5 + A_1B_4 - A_2B_7 + A_3B_6 - A_4B_1 + A_5B_0 - A_6B_3 + A_7B_2,\\
C_6 &= A_0B_6 + A_1B_7 + A_2B_4 - A_3B_5 - A_4B_2 + A_5B_3 + A_6B_0 - A_7B_1,\\
C_7 &= A_0B_7 - A_1B_6 + A_2B_5 + A_3B_4 - A_4B_3 - A_5B_2 + A_6B_1 + A_7B_0.
\end{aligned}
\tag{1}
$$

We refer to this as the traditional multiplication for the 8D case.

Analyzing the above equation we note that the real multiplication is represented by the first term of the first line. Complex multiplication is represented by the first two terms of the first two lines. Likewise, the quaternion multiplication is represented by the first four terms of the first four lines.

For the 5D version of the traditional multiplication considered in [12], the last three components of the two operands are set to zero before each multiplication operation occurs, and the last three components of the product are set to zero after each multiplication occurs — in other words, they are truncated and the data along those dimensions are not considered.

3 Definition of a few useful functions in nD

Working in quaternion space we are able to write expressions for various functions in a compact way or in a componentwise fashion [18]. Here we propose to generalize some known quaternion-valued functions to n higher dimensions, by considering extra imaginary components. Specifically we consider generalizations of the exponential, natural logarithm, and power functions.

We treat each nD point A as consisting of a scalar part, namely the first component A_0, and an $(n-1)$D vector part with components $(A_1, \ldots, A_{n-1})$, and we define two measures for magnitude, one that does not include A_0 (ℓ_v), and one that includes A_0 (ℓ_s):

$$\ell_v = \sqrt{\textstyle\sum_{k=1}^{n-1} A_k^2}, \tag{2}$$

$$\ell_s = \sqrt{\textstyle\sum_{k=0}^{n-1} A_k^2}. \tag{3}$$

We now define the function $C = \exp(A)$ as

$$\begin{aligned} C_0 &= \exp(A_0)\cos(\ell_v), \\ C_k &= \frac{A_k}{\ell_v}\exp(A_0)\sin(\ell_v), \quad k = 1, \ldots, n-1. \end{aligned} \tag{4}$$

We note that $A_0, A_1, \ldots, A_{n-1}$ are real numbers, and that the individual instances of exp(), cos(), and sin() are the familiar real-valued functions.

Similarly the polar form of the function $C = \ln(A)$ may be written as

$$\begin{aligned} C_0 &= \ln(\ell_s), \\ C_k &= \frac{A_k}{\ell_v}\arccos\left(\frac{A_0}{\ell_s}\right), \quad k = 1, \ldots, n-1, \end{aligned} \tag{5}$$

where ln() denotes the natural logarithm. We note that technically the quaternion natural logarithm function is a multiple-valued function. Here we consider the principal value of the quaternion natural logarithm function so that $\ln(A)$ has a unique value.

The function $C = A^\beta$ is defined by its components as

$$\begin{aligned} C_0 &= \ell_s^\beta \cos(\theta), \\ C_k &= \frac{A_k}{\ell_v}\,\ell_s^\beta \sin(\theta), \quad k = 1, \ldots, n-1, \end{aligned} \tag{6}$$

where β is an integer exponent and θ represents an angle in radians:

$$\theta = \beta \arccos\left(\frac{A_0}{\ell_s}\right). \tag{7}$$

4 Definition of a multiplication operator in nD

Now that the power function has been defined, we can write an equation which expresses the multiplication of two operands in any dimension. For real, complex, and quaternion number types, it is known that

$$C = AB = BA = \exp(\ln(A) + \ln(B)). \tag{8}$$

As such, we propose to use this equation as a general-purpose multiplication operator.

For an arbitrary number of dimensions n, the multiplication is carried out by using the exp() and ln() functions along with the addition operator. We note that the generalized power function (6) is a special case of the multiplication operator (8).

The traditional multiplication operator (1) and the multiplication operator (8) are equivalent when $C = AB$ is an integer power function.

5 Results

In more conventional terms, the Julia set can be defined as the set of all positions within a bounded region of nD space that satisfy a particular mathematical criterion. The criterion usually applied is that, for each nD input position A, its magnitude (or modulus) remains less than some chosen threshold value while undergoing iteration. The simplest, yet nontrivial, iterative equation is provided by the quadratic polynomial

$$X \mapsto X^2 + D, \tag{9}$$

where D is a control parameter. (For clarity we omit the index of the iteration.) For all the examples presented here, we use a maximum iteration count of 255 and a threshold value of 4.

The space we are dealing with is five dimensional, and the Julia sets are visualized by considering planar cuts in two of the dimensions resulting in 3D solids. Two illustrations of views of the 3D solids from the 5D Julia sets for Eq. (9) are presented in Fig. 1 for two different values of D. Cross sections of the 3D solids in Fig. 1 are shown in Fig. 2. These cross sections in one of X_0, X_1 or X_2 are designed to reveal the interior structure of the solids. The 2D fractals revealed by these section cuts generally have a similar appearance, but are not directly predictable, to the traditional complex-valued 2D Julia sets. Examples of 2D cross sections are shown in Fig. 3. In addition to the planar cuts of $X_3 = 0$ and $X_4 = 0$ used in the other figures, these examples have a further planar cut of either $X_1 = 0$ or $X_2 = 0$.

A 3D solid and a cross sectional cut view of a 5D Julia set for $X \mapsto X^3 + D$ are shown in Fig. 4. The fact that this case of odd exponent appears to be previously undocumented is a curious fact, given that the traditional multiplication operator satisfies the property of closure. We can also consider the case of negative exponents in dimension 5, first introduced in [12] as a 2D cross section of a 5D generalized Julia set for $X \mapsto X^{-2} + D$ with $D = (0.3, 0.3, 0, 0.3, 0.44)$. Figure 5 shows a 3D solid view of this set and a 2D cross section. They exhibit not only the same pattern as that depicted in figure 5(c) of [12] but also a wealth of new detail and complexity.

The figures here were created by mapping 3 of the 5 dimensions onto a rectangularly bounded 3 dimensional volume. This mapping may be achieved by simply ignoring two dimensions (dimensional collapse) or by performing planar cuts in two of the dimensions. The scalar value at each voxel of this volume indicates the number of terms in the series before it exceeds the threshold value, that is, escapes to infinity. This volumetric dataset is rendered in real time using a ray marching algorithm running on the GPU. The particular algorithm used [19] was developed for microCT volumes. It not only allows each voxel value (escape time) to be mapped to colour and opacity, but provides shading effects derived from the local gradient. Voxels representing points that diverge almost immediately are made totally transparent. Voxels representing points that diverge less quickly are less transparent and shaded blue. Points that diverge more slowly are more opaque and golden in colour. Finally, points that do not exceed the threshold during the maximum allowed number of iterations, or converge to a single value, or are periodic are made opaque and shaded towards red. There are some artistic choices in assigning this system since some equations have very different voxel value distributions.

Computing higher dimensional Julia sets as 3D volumes can be computationally expensive due to the N^3 relationship of the points to be considered. In the images here the volume resolution is 1000x1000x1000 voxels ($N = 1000$) so computationally this involves 1000 more samples than the equivalent 2D Julia set calculation. Furthermore, to avoid aliasing effects arising from sampling discreetly at the voxel positions, each voxel is antialiased by averaging a 3x3x3 set of subsamples within each voxel. As such the computation time rises as $27N^3$. Fortunately one notes that each position can be computed entirely independently from any other position. This trivial parallelism can be readily exploited by explicitly generating the volume with a multi-threaded code. The performance improves linearly with the number of real threads. Future opportunities exist for creating these Julia sets in real time by performing the calculation on the GPU using GLSL shaders, or APIs such as CUDA or OpenCL.

There are also symmetries that can be exploited for improved computation times. For example, for the quadratic case, Eq. (9), it can be shown that the series at position X is the same as at position $-X$. This reflection about all axis planes can be hinted at in Fig. 1. Such symmetries were not considered here in the interest of a simpler and more general implementation; for a discussion of some aspects related to symmetry of sets defined by Eq. (9) consult A. A. Bogush et al. [20].

6 Conclusion

We have provided a definition of various operators in n dimensions, specifically exponentiation, natural logarithm and powers. These in turn give a general definition for multiplication that can be immediately applied to any dimension. This is in contrast to the traditional multiplication operator which is unique to each dimension.

Armed with these definitions one can generate Julia sets in any embedding dimension. The examples provided illustrate Julia fractals of various polynomials in 5 dimensional quintonion space.

The definitions presented here can be used to construct hyperdimensional sets for a number of other fractals.

References

[1] G. Julia, Mémoire sur l'itération des fonctions rationnelles, Journal de Mathématiques Pures et Appliquées 1 (1918) 47–246.

[2] C. A. Pickover, Visualization of quaternion slices, Image Vision Comput. 6 (1988) 235–237.

[3] A. Norton, Generation and Display of Geometric Fractals in 3-D, SIGGRAPH Comput. Graph. 16 (1982) 61–67.

[4] A. Norton, Julia sets in the quaternions, Comput. & Graphics 13(2) (1989) 267–278.

[5] J. A. R. Holbrook, Quaternionic asteroids and starfields, Applied Mathematics Notes 8(2) (1983) 1–34.

[6] J. A. R. Holbrook, Quaternionic Fatou-Julia sets, Ann. Sci. Math. Québec 11 (1987) 79–94.

[7] J. C. Hart, D. J. Sandin, and L. H. Kauffman, Ray Tracing Deterministic 3-D Fractals, SIGGRAPH Comput. Graph. 23 (1989) 289–296.

[8] C. J. Griffin and G. C. Joshi, Octonionic Julia sets, Chaos, Solitons & Fractals 2(1) (1992) 11–24.

[9] S. L. Dixon, K. L. Steele, and R. P. Burton, Generation and graphical analysis of Mandelbrot and Julia sets in more than four dimensions, Comput. & Graphics 20(3) (1996) 451–456.

[10] J. Cheng and J.-r. Tan, Generalization of 3D Mandelbrot and Julia sets, J. Zhejiang Univ. Sci. A 8(1) (2007) 134–141.

[11] X.-Y. Wang and Y.-Y. Sun, The general quaternionic M-J sets on the mapping $z \leftarrow z^a + c$ ($a \in \mathbb{N}$), Computers and Mathematics with Applications 53(11) (2007) 1718–1732.

[12] X. Wang and T. Jin, Hyperdimensional generalized M-J sets in hypercomplex number space, Nonlinear Dyn. 73 (2013) 843–852.

[13] S. Halayka, Some visually interesting non-standard quaternion fractal sets, Chaos, Solitons & Fractals 41 (2009) 2842–2846.

[14] A. Katunin, On the symmetry of bioctonionic Julia sets, Journal of Applied Mathematics and Computational Mechanics 12(2) (2013) 23–28.

[15] A. Katunin, The Generalized Biquaternionic M-J Sets, Journal for Geometry and Graphics 22 (2018) 49–58.

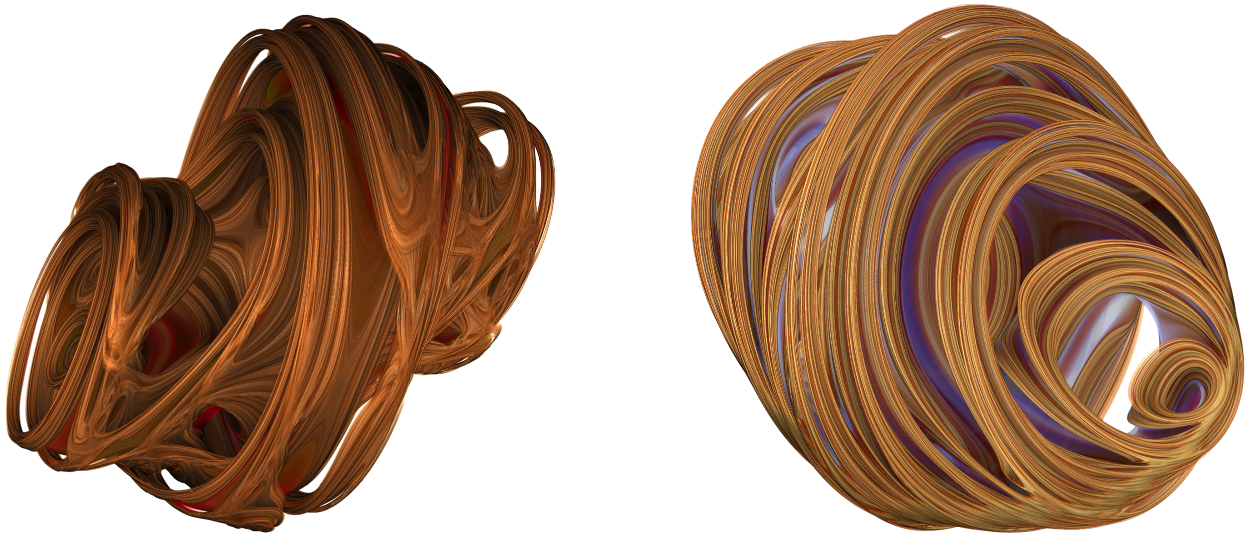
[16] K. Imaeda and M. Imaeda, Sedenions: algebra and analysis, Applied Mathematics and Computation 115 (2000) 77–88.

[17] A. Katunin, A Concise Introduction to Hypercomplex Fractals, Boca Raton: CRC Press; 2017.

[18] J. P. Morais, S. Georgiev, and W. Sprößig, Real Quaternionic Calculus Handbook, Basel: Birkhäuser; 2014.

[19] A. Limaye, Drishti: a volume exploration and presentation tool, Proc. SPIE 8506, Developments in X-Ray Tomography VIII, 85060X (17 October 2012).

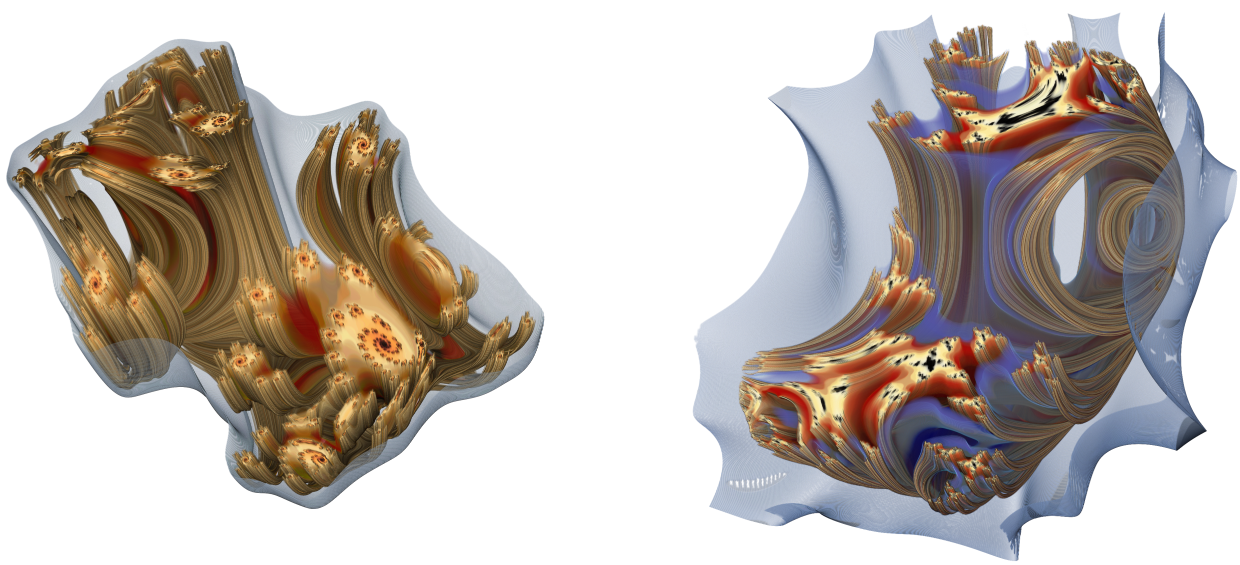
[20] A. A. Bogush, A. Z. Gazizov, Yu. A. Kurochkin, and V. T. Stosui, Symmetry properties of quaternionic and biquaternionic analogs of Julia sets, Ukr. J. Phys. 48(4) (2003) 295–299.



(a) $X \mapsto X^2 + D$
$D = (-0.3, -0.2, 0.4, -0.2, 0.5)$

(b) $X \mapsto X^2 + D$
$D = (0.3, 0.5, 0.4, 0.2, 0.0)$

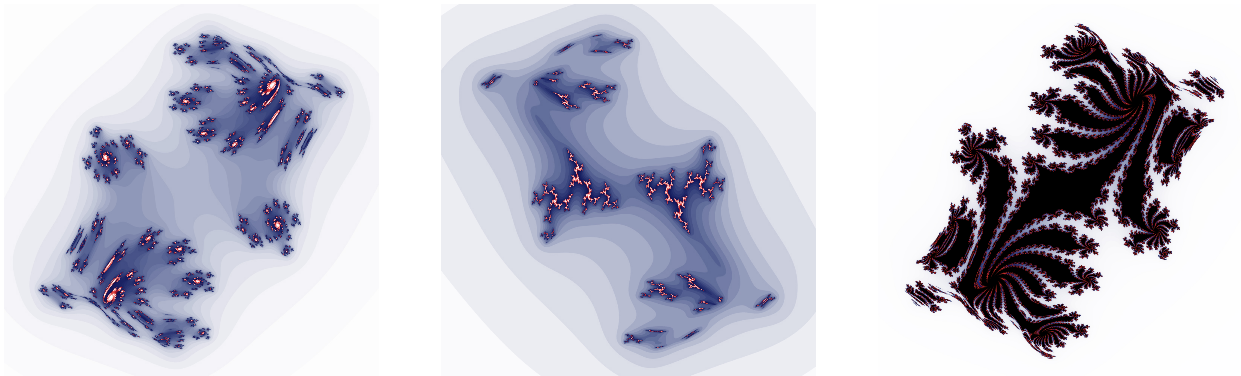
Figure 1



(a) $X \mapsto X^2 + D$
$D = (-0.3, -0.2, 0.4, -0.2, 0.5)$

(b) $X \mapsto X^2 + D$
$D = (0.3, 0.5, 0.4, 0.2, 0.0)$

Figure 2



(a) $X \mapsto X^2 + D$
$D = (-0.3, -0.2, 0.4, -0.2, 0.5)$

(b) $X \mapsto X^2 + D$
$D = (-0.3, -0.5, 0.4, -0.2, 0.5)$

(c) $X \mapsto X^2 + D$
$D = (-0.3, -0.2, 0.4, -0.2, 0.4)$

Figure 3

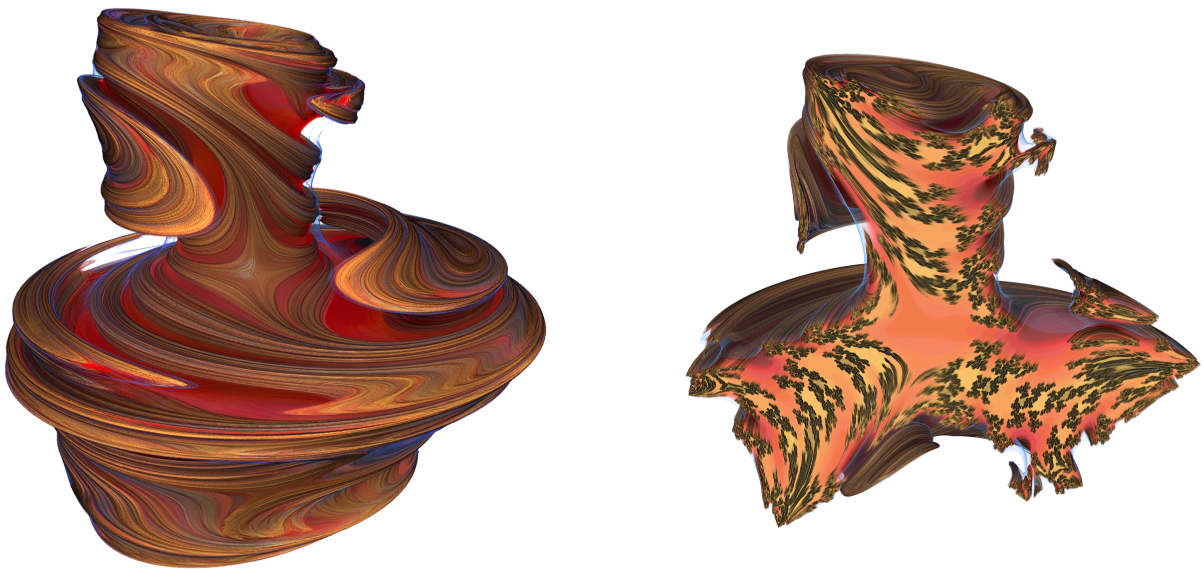


Figure 4: $X \mapsto X^3 + D$
$D = (0.4, 0.3, -0.3, 0.5, -0.2)$

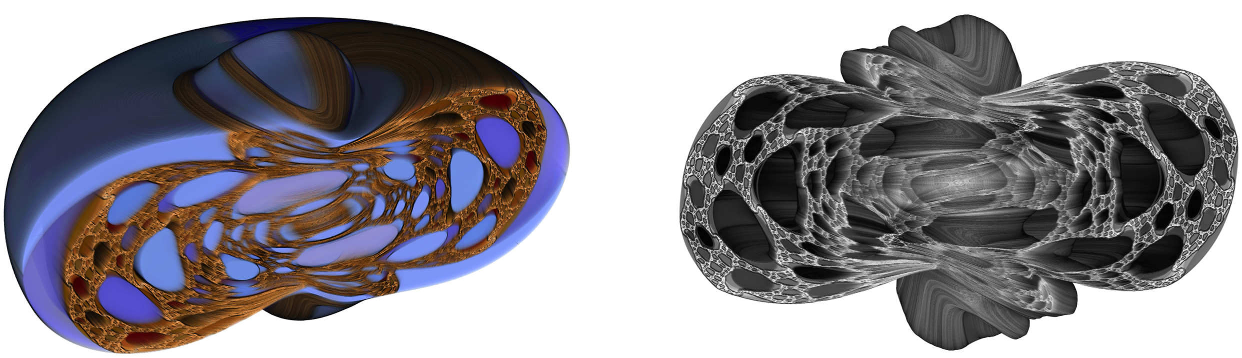


Figure 5: $X \mapsto X^{-2} + D$
$D = (0.3, 0.3, 0, 0.3, 0.44)$