

3D Rendering of the Quaternion Mandelbrot Set with Memory

Ricardo Fariello¹, Paul Bourke², and Gabriel V. S. Abreu¹

¹Universidade Estadual de Montes Claros, Montes Claros, MG, Brazil

²The University of Western Australia, Perth, Australia

February 24, 2024

Abstract

In this paper, we explore the quaternion equivalent of the Mandelbrot set equipped with memory and apply various visualization techniques to the resulting 4 dimensional geometry. Three memory functions have been considered, two that apply a weighted sum to only the previous two terms and one that performs a weighted sum of all previous terms of the series. The visualization includes one or two cutting planes for dimensional reduction to either 3 or 2 dimensions respectively, as well as employing an intersection with a half space to trim the 3D solids in order to reveal the interiors. Using various metrics, we quantify the effect of each memory function on the structure of the quaternion Mandelbrot set.

Keywords: Memory, Quaternion Mandelbrot Set, Visualization, Voxel.

1 Introduction

For the last 40 years, the Mandelbrot set has been employed as a means of illuminating chaos. The onset to the chaos takes place via the infinity of Myrberg–Feigenbaum points [25]. For the quadratic polynomial $f(z) = z^2 + c$, the complex Mandelbrot set, $M^{\mathbb{C}}$ -set, is defined as the set of all parameters $c \in \mathbb{C}$ for which the orbit of the origin under $f(z)$ remains bounded:

$$\begin{aligned} M^{\mathbb{C}}\text{-set} &= \{c \in \mathbb{C} \mid f^{[k]}(0) \not\rightarrow \infty \text{ as } k \rightarrow \infty\} \\ &= \{c \in \mathbb{C} \mid c, c^2 + c, (c^2 + c)^2 + c, \dots \not\rightarrow \infty \text{ as } k \rightarrow \infty\} \end{aligned} \quad (1)$$

In order to compute a trustworthy picture of the popular Mandelbrot set in the plane c , one would need to assess carefully how inefficiencies arise in the algorithm [28].

The standard notion of Mandelbrot set from Eq. (1) has a similar construct when transferred to a quaternionic context:

$$M^{\mathbb{H}}\text{-set} = \{q \in \mathbb{H} \mid f^{[k]}(0) \not\rightarrow \infty \text{ as } k \rightarrow \infty\} \quad (2)$$

but now $f^{[k]}(0)$ is the k th iterate of the quaternion polynomial $f(Q) = Q^2 + q$. Our working definition (Eq. (2)) follows, for example, from Refs. [8, 13, 14, 22, 26], which are based upon earlier proposed approaches by Pickover [27], Norton [23, 24] — who pioneered investigations into extending fractals to quaternions, and by Holbrook [16, 17].

Numerous alternatives to the Mandelbrot set [1, 18, 19, 29, 33] have been developed. They are aimed at the modification of the conventional iterative process to form new patterns through combinations of different parameters. Such patterns can be used for applications in physics. For example, [7] indicated that certain classical particle dynamics possess the complex logistic map as a stroboscopic mapping. Consequently, the Mandelbrot set of the complex logistic map has a solid physical meaning.

When quaternions are utilized, interesting variations in the Mandelbrot set emerge, for example, through the embedding of a memory effect [6] or of a noise source [31]. However, the potential of the discussion of the use of memory has been limited by the prevalence of only trivial planar cross-sections for the quaternion parameter q . Taking into account the 4D nature of the quaternion numbers, it is possible to go beyond such a 2D projection and to generate more representational 3D structures.

In discrete-time maps $z \mapsto f(z)$, memory effects can be implemented from two perspectives, delay and embedded. Both of these can be allocated to all (full) or some (partial) of the state variables. Regarding the $M^{\mathbb{C}}$ -set, the use of the embedded memory is the subject of two papers, one concerned with the partial scheme [4] and the other with the full scheme [3]. Another paper [5] is focused on the delay memory in connection with the full scheme. Regarding the $M^{\mathbb{H}}$ -set, there is only one paper [6] in which the author has examined both types of memory using the full scheme. These works have considered a number of postulated forms of memory, including short-term and long-term. They were introduced in [12] for general nonlinear discrete-time maps. Additionally, metrics (area and center of mass) were considered as a potential tool for assessing the impacts of memory on the $M^{\mathbb{C}}$ -set [3, 4, 5] and on the $M^{\mathbb{H}}$ -set [6].

In light of the above, in this paper we break new ground by addressing the issue of rendering and visualizing 3D projections of the $M^{\mathbb{H}}$ -set in the presence of memory via the embedded map $z \mapsto f(\bar{m})$ (being \bar{m} an average of past states). This paper is structured as follows. First, in Sec. 2, we review the constructs of memory that guide our work and use them to calculate metrics such as volume and center of mass, which will be needed in the sequel. In Sec. 3, we offer an outline of how the 3D solids for the $M^{\mathbb{H}}$ -set with memory can be assembled and visualized. Illustrative examples are presented in Sec. 4. Lastly, we give some concluding remarks and propose some future work in Sec. 5.

2 Preliminaries

We denote by \mathbb{H} Hamilton's quaternions, that is, the set $\{q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}, q_m \in \mathbb{R}\}$ with $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -\mathbf{1}$. We start by considering the quaternion series

$$Q_0, Q_1 = f(Q_0), Q_2 = f(Q_1), \dots, Q_k = f(Q_{k-1})$$

The next term of the series is

$$Q_{k+1} = f(\bar{m}(Q_1, Q_2, \dots, Q_k))$$

where the memory function $\bar{m}(Q_1, Q_2, \dots, Q_k)$ is a weighted sum of the previous k states. In this particular implementation, memory is called embedded memory.

We choose the following cases for $\bar{m}_k = \bar{m}(Q_1, Q_2, \dots, Q_k)$:

- Case 1: $\bar{m}_k = Q_k$
- Case 2: $\bar{m}_k = (1 - \sigma)Q_k + \sigma Q_{k-1}$
- Case 3: $\bar{m}_k = \frac{Q_k + \sigma Q_{k-1}}{1 + \sigma}$
- Case 4: $\bar{m}_k = \frac{Q_k + \sum_{i=1}^{k-1} \sigma^{k-i} Q_i}{1 + \sum_{i=1}^{k-1} \sigma^{k-i}}$

The number σ in the above is a real number between 0 and 1, inclusive.

Case 1 corresponds to the memoryless dynamics, while Cases 2 and 3 correspond to short-term memory in which the weighted sum of the previous two terms are applied. When $\sigma = 0$, Cases 2, 3 and 4 all revert to Case 1. In Case 2, if $\sigma = 1/2$, then the last two terms of the series are equally averaged, and similarly for Case 3 when $\sigma = 1$. In general, for some given $\sigma \leq 1/2$, Case 2 becomes Case 3 if we remap σ to $\sigma/(1 + \sigma)$. As such, Case 2 exhibits a wider range of dynamics than that of Case 3.

Finally, Case 4 corresponds to a long-term memory in which all past states are considered to determine the next state, and each contributes an inversely decreasing weight. For $\sigma = 1$, this memory is impartial, that is, it is just the average of all past terms in the series. One might note that all three of these types of memories are applicable for $k > 1$.

As stated in the introduction [see Eq. (2)], the $M^{\mathbb{H}}$ -set can be generated in quaternion space as

$$Q_{k+1} = f(Q_k) = Q_k^2 + q \quad (3)$$

with quaternions $Q = x + y\mathbf{i} + z\mathbf{j} + w\mathbf{k}$ and $q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$, and seed $Q_0 = 0 + 0\mathbf{i} + 0\mathbf{j} + 0\mathbf{k}$. In terms of real variables x, y, z and w , and constants a, b, c and d , Eq. (3) can be transformed into the form of four real equations:

$$\begin{aligned} x_{k+1} &= x_k^2 - y_k^2 - z_k^2 - w_k^2 + a \\ y_{k+1} &= 2x_k y_k + b \\ z_{k+1} &= 2x_k z_k + c \\ w_{k+1} &= 2x_k w_k + d \end{aligned}$$

We use these equations to allow the embedding of memory into the $M^{\mathbb{H}}$ -set. In particular, we are interested in the effects that exposure to memory has on the dynamics of the four variables. Such effects can be accounted for by means of the set of equations:

$$\begin{aligned} x_{k+1} &= \bar{x}_k^2 - \bar{y}_k^2 - \bar{z}_k^2 - \bar{w}_k^2 + a \\ y_{k+1} &= 2\bar{x}_k \bar{y}_k + b \\ z_{k+1} &= 2\bar{x}_k \bar{z}_k + c \\ w_{k+1} &= 2\bar{x}_k \bar{w}_k + d \end{aligned} \quad (4)$$

with the bar denoting averaging over past states of a single variable. Thus, for Case 2, for example, the barred variable \bar{x}_k in Eq. (4) is given by $\bar{x}_k = (1 - \sigma)x_k + \sigma x_{k-1}$, with analogous expressions for the other three variables (y , z and w). We refer to this approach as the full allocation method. More on this and additional approaches to the formation of models of discrete systems with memory can be found in [2].

The 4D volume of the interior of the set can be estimated by gridding the enclosing region into equal size tesseracts and evaluating the behaviour of the quaternion series at the center of each. The total count of tesseracts inside the set multiplied by the volume of a single tesseract gives an estimate of the total 4D volume. It's an estimate because, due to computational limitations, the practical size of the tesseracts are limited, with the true volume being the limit as the size of the sampled tesseract volumes tend to 0.

Figure 1 shows the 4D volume for the three Cases 2, 3 and 4 as σ is varied between 0 and 1. The enclosing range has been chosen to be -5 to 5 in all 4 dimensions, each centered at the origin. This was predetermined to enclose the set in all cases, except for $\sigma > 0.4$ in Case 4. The samples are taken on a regular grid every 0.05 units in all dimensions. Case 1 has no memory and no σ -dependence; the constant volume is estimated at 1.155. Cases 2 and 3 have clear maxima within the σ -range. The maximum for Case 2 occurs experimentally at $\sigma_* = 0.36$, whereas the maximum for Case 3 occurs experimentally at 0.565, which is in line with the expected theoretical value of $\sigma_*/(1 - \sigma_*)$. The volume for Case 4 continues to increase monotonically for larger σ than shown, as the set continues to grow.

Figure 2 shows the results of computing the center of mass of the x component of the set as a function of σ . The center of mass of the other three components (y , z and w) are zero. This is computed in a similar fashion to the volume, except now each 4D point on a regular grid in the set is summed and normalised by the number of sample points, that is, the center of mass of a uniform density material. As expected, there is no dependence on σ for Case 1. Cases 2 and 3 shift along the negative x axis to a maximum of 0.38 and 0.58, respectively, before shifting back towards the positive x axis. In Case 4, the center of mass of the x component continues to shift along the negative x axis. Knowing the volume, bounds and center of mass assists in determining the center and range of the set for sampling in the visualization discussed in the next section.

3 Recipe for the 3D solids

The visualization of the quaternion fractals involves investigating the behaviour of the series in the same way as the traditional Mandelbrot image is generated, except here positions within a bounded region in 4 dimensional space are explored rather than on a 2D plane. The challenge is often how to render the resulting volumetric data. One approach is to create isosurfaces [23, 24] which has the advantage of being able to be viewed and rendered in almost any 3D software. The problem with this approach is the high mesh density required to represent the fractal detail, and that one has to choose a single isosurface thus discarding the other escape time information. Another approach is to develop a ray-tracing solution specific to the problem at hand [15, 30]. Here, rendering techniques have been adopted from volumetric data visualization in the sciences, namely ray casting volume rendering widely used for 3D volumetric datasets in medicine [32] (e.g., MRI scans), materials science [11] (e.g.,

Micro-CT) and geology [21]. In this paper the Drishti software [20] was chosen because it utilizes the GPU for interactive performance and its shading model includes consideration of the local gradient, resulting in more informative and compelling visualizations [9].

Volume visualization typically requires that a scalar quantity is assigned to every 3D pixel (voxel), a regular discrete sampling of a rectangularly bounded region of 3D space. In order to apply this to the quaternion fractals we need to reduce the dimensionality by one. This is achieved by slicing the 4D volume in one dimension. In the examples here, the slice plane is achieved by setting d to 0.5. The volumetric data is therefore created by evaluating the quaternion series at each voxel location, and the scalar stored at each voxel is the number of terms before the series escapes to infinity (outside the set) or the maximum length the series can have in which cases the voxel is assigned to be within the set. In order to reduce the sampling effects that can arise from any discrete sampling, multiple samples are taken on a $3 \times 3 \times 3$ subsampling of each voxel and the results are averaged. This is known in the computer graphics industry as standard supersampling antialiasing. The volume rendering process involves mapping the scalar at each voxel to a color and opacity. Rays are then cast from a virtual camera position and the way those rays interact with the color and opacity of the voxels forms the rendered image. For example, in the figures here the voxels within the set are made totally opaque. The faster the voxels escape, the more transparent they are made.

In addition to the challenges of visualizing 4D solids, there are also computational challenges. This is particularly so for Case 4 where the sum occurs over the entire length of the series rather than just the previous two terms. The computation time for the various metrics performed in 4D increases as the 4th power of the per dimension sampling resolution. The generation of the 3D solids rises as the 3rd power of the product of the per dimension resolution and the degree of antialiasing. It should be noted, though, that the various calculations performed per sample are independent of all other samples and thus this is a trivially parallel process. All the data used in the volume visualizations as well as the various metrics were calculated using software written in C/C++. The implementation utilized threads where each thread was assigned a 2D plane. This provided a sufficient processing quantum and achieved a level of load balancing. As such, the performance improvement over a single thread was linear with the number of real cores available.

4 Graphical Examples

The 3D solids for Cases 2, 3 and 4 are shown, respectively, in Figs. 3, 4, and 5 for a selection of six values of σ (0, 0.2, 0.4, 0.6, 0.8 and 1). The center position and bounds for a and b are shown, the bounds for c are the same as for b , and d is a fixed slicing plane at 0.5. In all cases, the series is evaluated for at most 255 terms, chosen as such because the volume visualization software only supports single byte scalar values at each voxel. The volumes in all cases are generated at $512 \times 512 \times 512$ voxels, a tradeoff between computation time and the resolution that can be represented in the paper figures.

From Fig. 3, as expected, the set undergoes one round of notable changes when under short-term memory load, resulting in two distinct phases. The first phase, up to $\sigma = 0.36$, is characterized by a shift of the center of mass to the left accompanied by a dilation of the

volume; from then onwards, the set enters a second phase in which it is brought back to the memoryless configuration (Case 1). The sets shown in Fig. 3 for Case 2 with $\sigma = 0.2$ and $\sigma = 0.4$ are, as one might expect, similar to the sets shown in Fig. 4 for Case 3 with $\sigma = 0.2$ and $\sigma = 0.6$ respectively. Similar to Case 2, but in a broader σ -range, Case 3 sets also present a two-phase behavior consisting of changes in size and position. Besides the continuous increase in volume, the sets depicted in Fig. 5 for Case 4 remarkably show that structures similar to the Mandelbrot set are maintained, although in a range where the long-term memory load is not as high, up to $\sigma = 0.4$.

In addition to dimension reduction with a cutting plane, one can also trim the 3D solids by a half space. Figure 6 trims the solids at $\sigma = 0.4$ with the positive half space starting from the origin on the b axis. In Fig. 7 for $\sigma = 0.4$ the solids are trimmed with the positive half space starting 0.5 units from the origin on the b axis.

A further slice can be performed to reduce the dimensionality to 2. Figure 8 slices the b axis by a plane at the origin resulting in a 2 dimensional image of the (a, c) plane, also for $\sigma = 0.4$.

With the 4D visualization techniques presented, one can now consider other quaternion fractal sets. An example is to consider the generalized Mandelbrot set whose polynomial is

$$Q_{k+1} = Q_k^p + q, \quad p \in \mathbb{N}$$

Figure 9 shows the sliced volumes for $p = 3$ and $p = 4$ for the three memory cases with $\sigma = 0.33$. The slicing plane is at $d = 0.5$, and the trimming by a half space at $b = 0.25$.

5 Conclusions and Future Work

We have described three types of memory and applied them to the Mandelbrot fractal in quaternion space. Metrics such as volume and center of mass have been calculated in 4D rather than only lower dimensional cross-sections. These showed a transition in behaviour as a function of the memory load σ in Cases 2 and 3, but not in Case 4. Comparisons between the effects of the different types of memory have been made by generating 3D volumes using a single slice of the 4D fractal. These are visualized using volume rendering, including intersecting with a half space to reveal the interior.

The 4D volumes can be computationally expensive, especially for Case 4 which is k times more demanding than the other cases where k can be as large as the number of terms of the series used to determine whether a voxel is interior or exterior to the set. This consideration also reveals that the computation time is largely dependent on the volume of the interior of the set that is contained within the volumetric region under consideration. The trivial parallel nature of the generation is obvious from the equations (each voxel state is computed independently of the neighbors) and this has been demonstrated by employing threads and observing the computation time is linearly proportional to the number of cores. Future work can include the realtime computation most likely employing general purpose computing on the GPU such as Cuda or OpenCL.

With these tools and techniques in place, future research will look at exploring the application to other traditionally 2D escape-time fractals, like the Julia set, whose generalization to any dimension was recently given in [10].

References

- [1] Abbas, M., Iqbal, H., De la Sen, M.: Generation of Julia and Mandelbrot Sets via Fixed Points. *Symmetry* **12**(1) (2020). DOI 10.3390/sym12010086. URL <https://www.mdpi.com/2073-8994/12/1/86>
- [2] Alonso-Sanz, R.: *Discrete Systems with Memory*. World Scientific (2011). DOI 10.1142/8119. URL <https://www.worldscientific.com/doi/abs/10.1142/8119>
- [3] Alonso-Sanz, R.: A Glimpse of Complex Maps with Memory. *Complex Systems* **21**(04), 269–282 (2013). DOI 10.25088/ComplexSystems.21.4.269. URL https://www.complex-systems.com/abstracts/v21_i04_a02/
- [4] Alonso-Sanz, R.: The Mandelbrot Set with Partial Memory. *Complex Systems* **23**(03), 227–238 (2014). DOI 10.25088/ComplexSystems.23.3.227. URL https://www.complex-systems.com/abstracts/v23_i03_a02/
- [5] Alonso-Sanz, R.: On Complex Maps with Delay Memory. *Fractals* **23**(03), 1550027 (2015). DOI 10.1142/S0218348X15500279. URL <https://doi.org/10.1142/S0218348X15500279>
- [6] Alonso-Sanz, R.: On Quaternion Maps with Memory. *Complex Systems* **24**(03), 223–232 (2015). DOI 10.25088/ComplexSystems.24.3.223. URL https://www.complex-systems.com/abstracts/v24_i03_a02/
- [7] Beck, C.: Physical meaning for Mandelbrot and Julia sets. *Physica D: Nonlinear Phenomena* **125**(3), 171–182 (1999). DOI [https://doi.org/10.1016/S0167-2789\(98\)00243-7](https://doi.org/10.1016/S0167-2789(98)00243-7). URL <https://www.sciencedirect.com/science/article/pii/S0167278998002437>
- [8] Bedding, S., Briggs, K.: Iteration of quaternion functions. *The American Mathematical Monthly* **103**(8), 654–664 (1996). URL <http://www.jstor.org/stable/2974877>
- [9] Bourke, P.: Visualising volumetric fractals. *GSTF International Journal on Computing* **5**(2) (2017). DOI 10.5176/2251-3043_5.2.370
- [10] Fariello, R., Bourke, P., Lopes, J.P.: Calculating Julia Fractal Sets in Any Embedding Dimension. *Fractals* **31**(01), 2350018 (2023). DOI 10.1142/S0218348X23500184. URL <https://doi.org/10.1142/S0218348X23500184>
- [11] Fidan, S.: *The use of Micro-CT in Materials Science and Aerospace Engineering*, pp. 267–276. Springer International Publishing, Cham (2020). DOI 10.1007/978-3-030-16641-0_16. URL https://doi.org/10.1007/978-3-030-16641-0_16
- [12] Fulinski, A., Kleczkowski, A.S.: Nonlinear Maps with Memory. *Physica Scripta* **35**(2), 119 (1987). DOI 10.1088/0031-8949/35/2/004. URL <https://dx.doi.org/10.1088/0031-8949/35/2/004>
- [13] Gomatam, J., Doyle, J., Steves, B.: *Quaternionic Generalisation of the Mandelbrot Set*, pp. 557–562. Springer US, Boston, MA (1995). DOI 10.1007/978-1-4899-1085-1_54. URL https://doi.org/10.1007/978-1-4899-1085-1_54

- [14] Gomatam, J., Doyle, J., Steves, B., McFarlane, I.: Generalization of the Mandelbrot set: Quaternionic quadratic maps. *Chaos, Solitons & Fractals* **5**(6), 971–986 (1995). DOI [https://doi.org/10.1016/0960-0779\(94\)00163-K](https://doi.org/10.1016/0960-0779(94)00163-K). URL <https://www.sciencedirect.com/science/article/pii/096007799400163K>
- [15] Hart, J., Sandin, D., Kauffman, L.: Ray tracing deterministic 3-D fractals. *ACM SIGGRAPH Computer Graphics* **23**(3), 289–296 (1989). DOI 10.1145/74334.74363
- [16] Holbrook, J.A.R.: Quaternionic astroids and starfields. *Applied Mathematical Notes* **8**(2), 1–34 (1983)
- [17] Holbrook, J.A.R.: Quaternionic Fatou–Julia sets. *Ann. Sci. Math. Québec* **11**(1), 79–94 (1987)
- [18] Kumari, S., Gdawiec, K., Nandal, A., Kumar, N., Chugh, R.: An Application of Viscosity Approximation Type Iterative Method in the Generation of Mandelbrot and Julia Fractals. *Aequationes mathematicae* **97**(2), 257–278 (2023). DOI 10.1007/s00010-022-00908-z. URL <https://doi.org/10.1007/s00010-022-00908-z>
- [19] Kumari, S., Gdawiec, K., Nandal, A., Postolache, M., Chugh, R.: A novel approach to generate Mandelbrot sets, Julia sets and biomorphs via viscosity approximation method. *Chaos, Solitons & Fractals* **163**, 112540 (2022). DOI <https://doi.org/10.1016/j.chaos.2022.112540>. URL <https://www.sciencedirect.com/science/article/pii/S0960077922007366>
- [20] Limaye, A.: Drishti: a volume exploration and presentation tool. In: S.R. Stock (ed.) *Developments in X-Ray Tomography VIII*, vol. 8506, p. 85060X. International Society for Optics and Photonics, SPIE (2012). DOI 10.1117/12.935640. URL <https://doi.org/10.1117/12.935640>
- [21] Ma, C., Rokne, J.: *3D Seismic Volume Visualization*, pp. 241–262. Springer US, Boston, MA (2004). DOI 10.1007/1-4020-7775-0_13. URL https://doi.org/10.1007/1-4020-7775-0_13
- [22] Nakane, S.: Dynamics of a family of quadratic maps in the quaternion space. *International Journal of Bifurcation and Chaos* **15**(08), 2535–2543 (2005). DOI 10.1142/S0218127405013460. URL <https://doi.org/10.1142/S0218127405013460>
- [23] Norton, A.: Generation and display of geometric fractals in 3-d. In: *Proceedings of the 9th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '82*, p. 61–67. Association for Computing Machinery, New York, NY, USA (1982). DOI 10.1145/800064.801263. URL <https://doi.org/10.1145/800064.801263>
- [24] Norton, A.: Julia sets in the quaternions. *Computers & Graphics* **13**(2), 267–278 (1989). DOI [https://doi.org/10.1016/0097-8493\(89\)90071-X](https://doi.org/10.1016/0097-8493(89)90071-X). URL <https://www.sciencedirect.com/science/article/pii/009784938990071X>

- [25] Pastor, G., Romera, M., Álvarez, G., Arroyo, D., Montoya, F.: On periodic and chaotic regions in the Mandelbrot set. *Chaos, Solitons & Fractals* **32**(1), 15–25 (2007). DOI <https://doi.org/10.1016/j.chaos.2005.10.099>. URL <https://www.sciencedirect.com/science/article/pii/S0960077905010854>
- [26] Petek, P.: On the Quaternionic Julia Sets, pp. 53–58. Springer US, Boston, MA (1992). DOI [10.1007/978-1-4615-3464-8_6](https://doi.org/10.1007/978-1-4615-3464-8_6). URL https://doi.org/10.1007/978-1-4615-3464-8_6
- [27] Pickover, C.A.: Visualization of quaternion slices. *Image and Vision Computing* **6**(4), 235–237 (1988). DOI [https://doi.org/10.1016/0262-8856\(88\)90013-3](https://doi.org/10.1016/0262-8856(88)90013-3). URL <https://www.sciencedirect.com/science/article/pii/0262885688900133>
- [28] Rojas, R.: A tutorial on efficient computer graphic representations of the Mandelbrot set. *Computers & Graphics* **15**(1), 91–100 (1991). DOI [https://doi.org/10.1016/0097-8493\(91\)90035-G](https://doi.org/10.1016/0097-8493(91)90035-G). URL <https://www.sciencedirect.com/science/article/pii/009784939190035G>
- [29] Shahid, A.A., Nazeer, W., Gdawiec, K.: The Picard–Mann iteration with s-convexity in the generation of Mandelbrot and Julia sets. *Monatshefte für Mathematik* **195**(4), 565–584 (2021). DOI [10.1007/s00605-021-01591-z](https://doi.org/10.1007/s00605-021-01591-z). URL <https://doi.org/10.1007/s00605-021-01591-z>
- [30] da Silva, V., Novello, T., Lopes, H., Velho, L.: Real-Time Rendering of Complex Fractals, pp. 529–544. Apress, Berkeley, CA (2021). DOI [10.1007/978-1-4842-7185-8_33](https://doi.org/10.1007/978-1-4842-7185-8_33). URL https://doi.org/10.1007/978-1-4842-7185-8_33
- [31] Sun, Y., Li, P., Lu, Z.: Generalized quaternion M sets and Julia sets perturbed by dynamical noises. *Nonlinear Dynamics* **82**(1), 143–156 (2015). DOI [10.1007/s11071-015-2145-7](https://doi.org/10.1007/s11071-015-2145-7). URL <https://doi.org/10.1007/s11071-015-2145-7>
- [32] Zhang, Q., Eagleson, R., Peters, T.M.: Volume visualization: A technical overview with a focus on medical applications. *Journal of Digital Imaging* **24**, 640–664 (2011). URL <https://api.semanticscholar.org/CorpusID:2277438>
- [33] Zou, C., Shahid, A.A., Tassaddiq, A., Khan, A., Ahmad, M.: Mandelbrot Sets and Julia Sets in Picard–Mann Orbit. *IEEE Access* **8**, 64411–64421 (2020). DOI [10.1109/ACCESS.2020.2984689](https://doi.org/10.1109/ACCESS.2020.2984689)

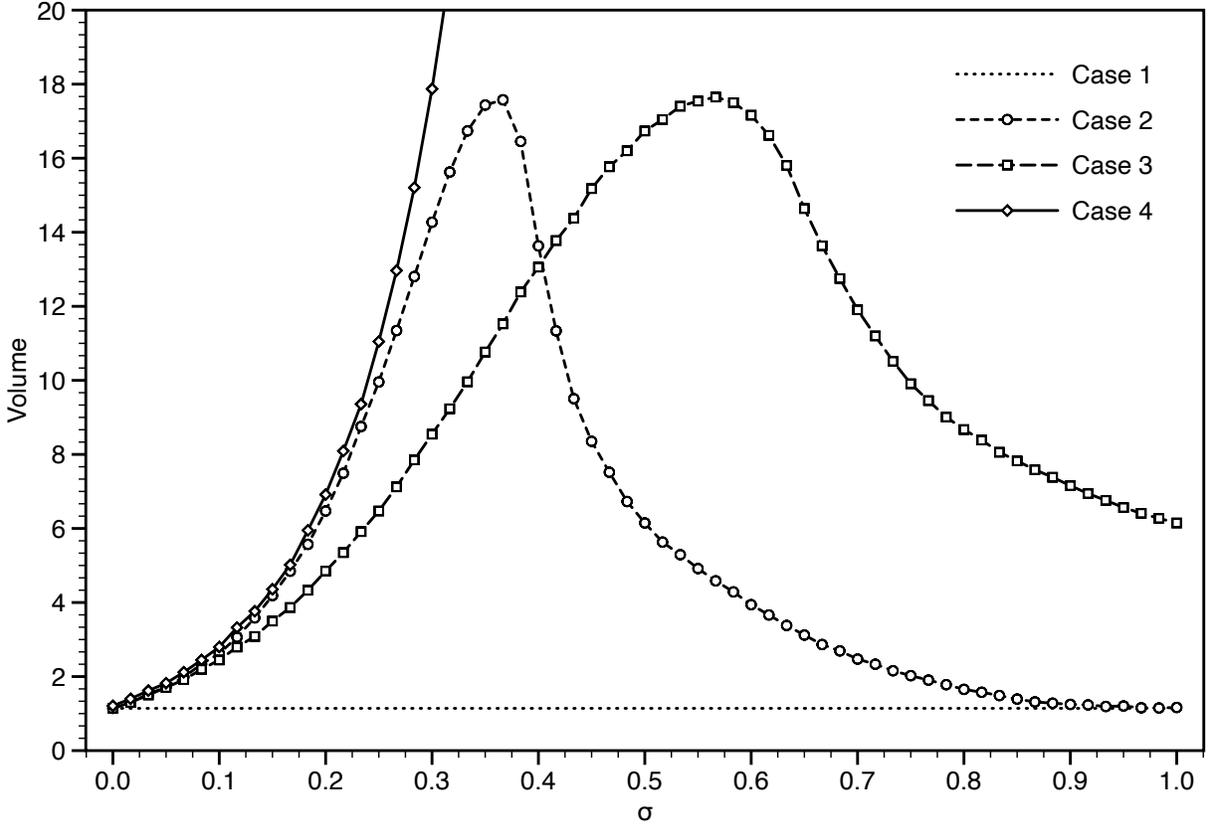


Figure 1: 4 dimensional volume as a function of σ for all four cases. The dotted line marks the volume of the $M^{\mathbb{H}}$ -set (Case 1). Both peaks for Case 2 and 3 coincide in value; that is, they are both equal to 17.7.

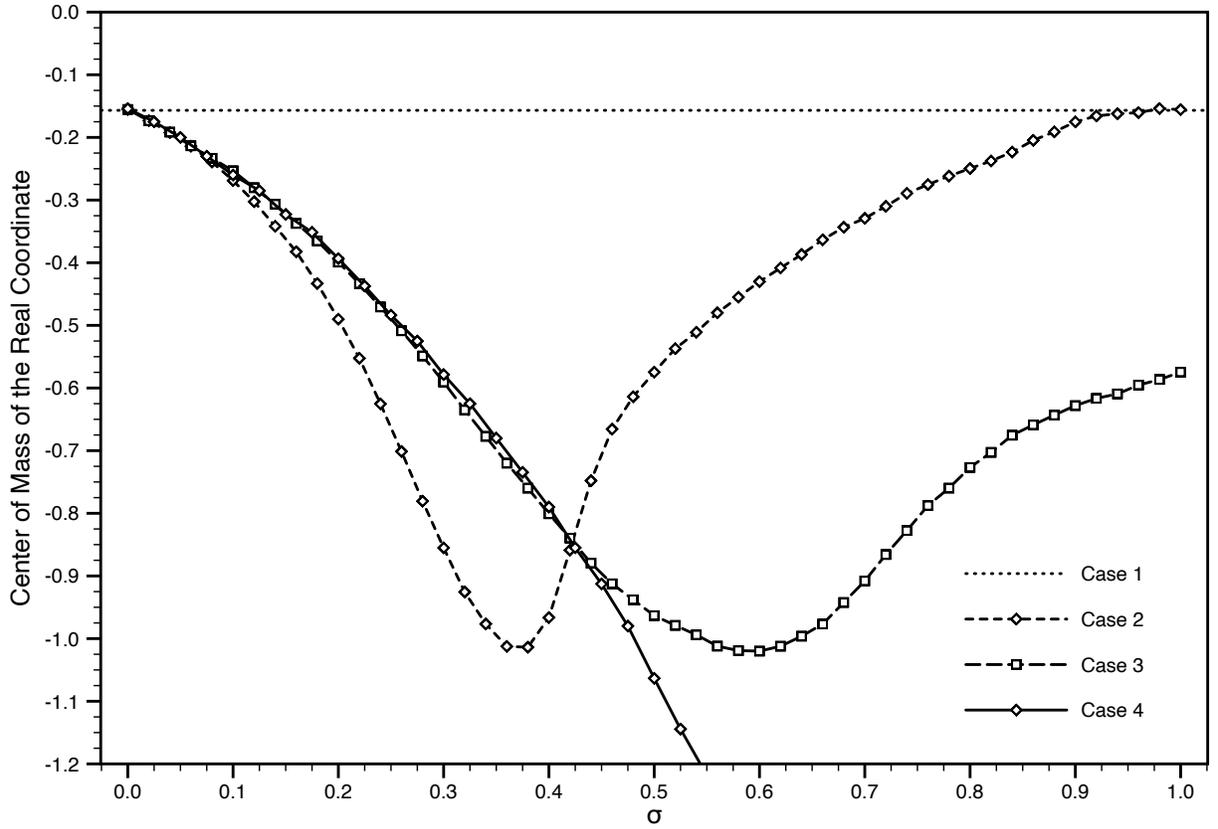


Figure 2: Center of mass of the real coordinate as a function of σ for all four cases. The minima of the valleys are precisely at the maxima of the peaks in the volume curves.

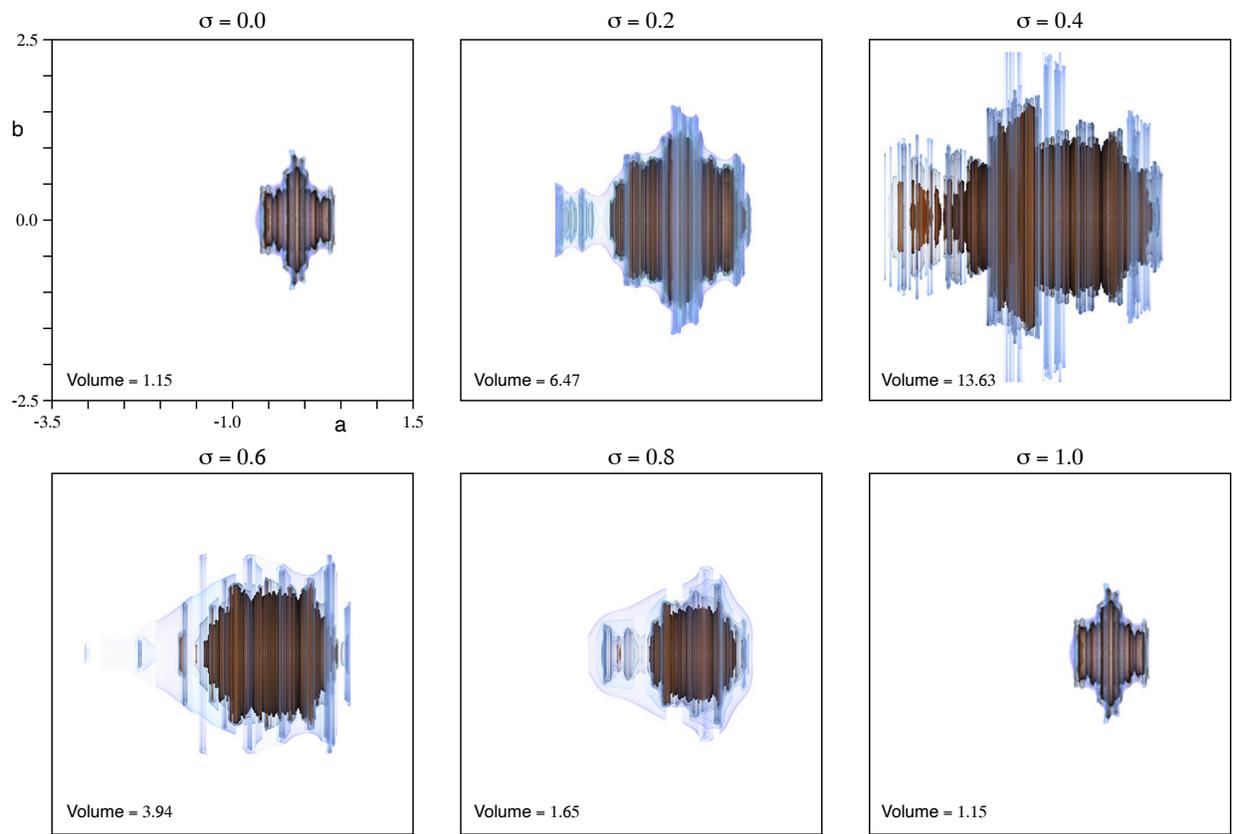


Figure 3: Volume visualization of the set for Case 2 at $d = 0.5$ at selected values of σ .

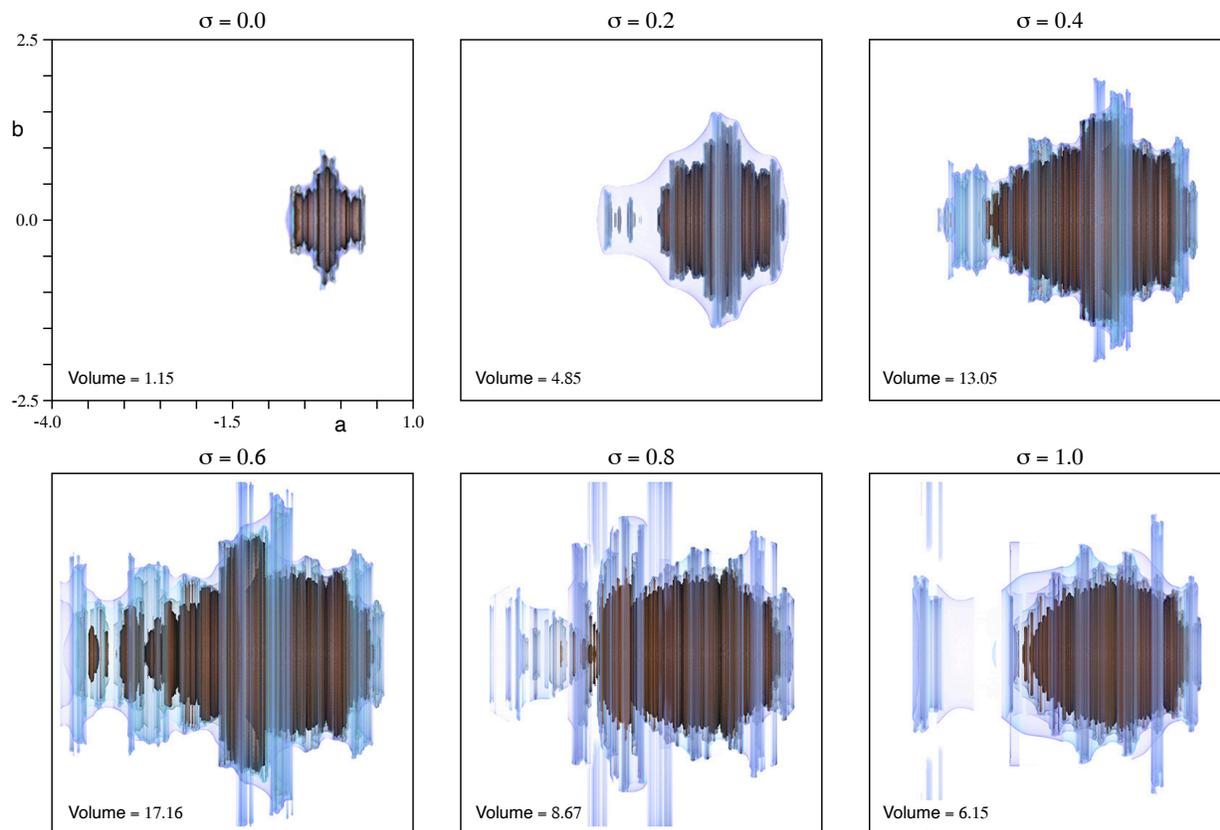


Figure 4: Volume visualization of the set for Case 3 at $d = 0.5$ at selected values of σ .

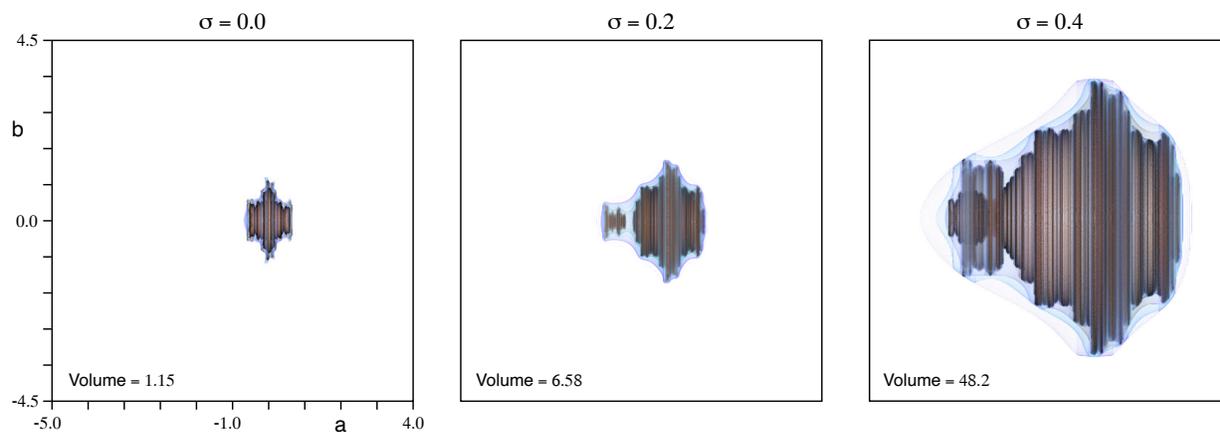


Figure 5: Volume visualization of the set for Case 4 at $d = 0.5$ at selected values of σ .

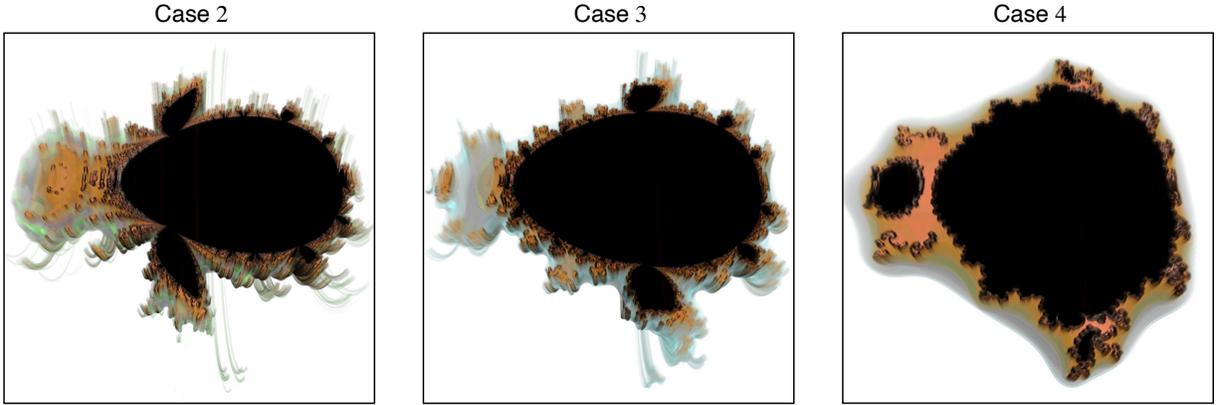


Figure 6: Trimming slice through a half space at the origin on the b axis. The slicing plane is at $d = 0.5$. The view bounds are scaled to fit the frame.

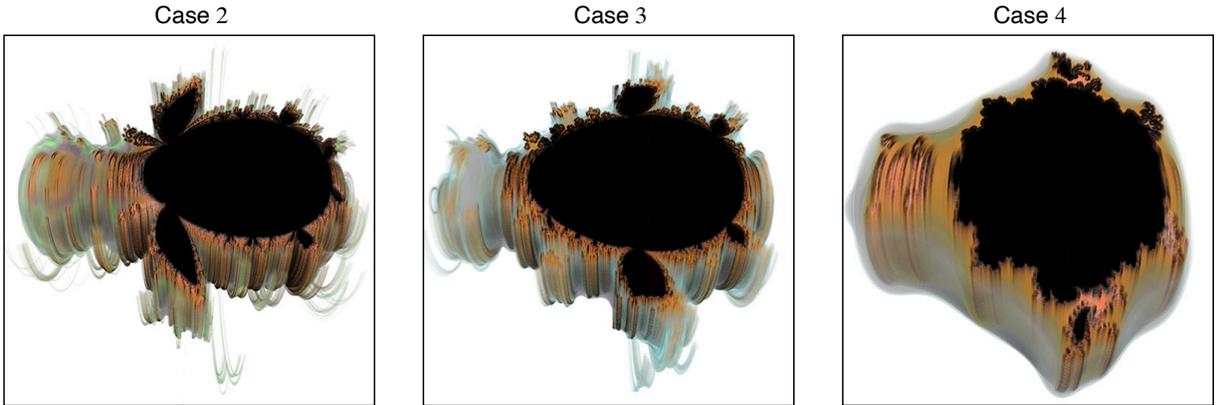


Figure 7: Trimming slice through a half space 0.5 units from the origin on the b axis. The slicing plane is at $d = 0.5$. The view bounds are scaled to fit the frame.

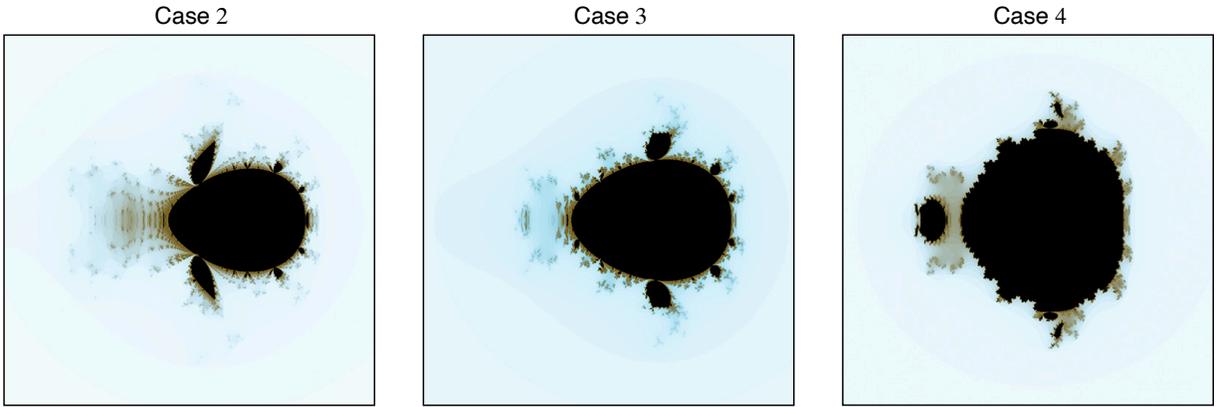


Figure 8: Slicing plane at $d = 0.5$ followed by a slicing plane at $b = 0$. The view bounds are scaled to fit the frame.

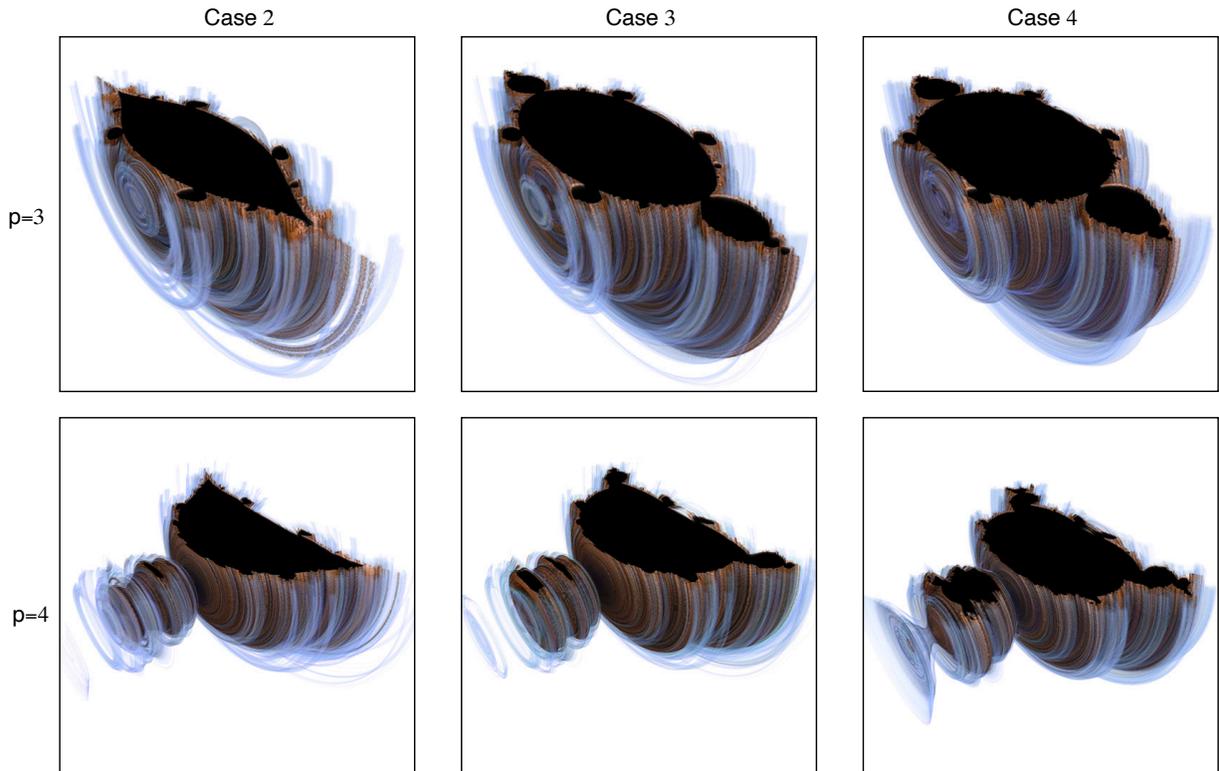


Figure 9: Half space intersections for powers of 3 and 4 for each case at $\sigma = 0.33$. The slicing plane is at $d = 0.5$. The view bounds are scaled to fit the frame.