AIMS Mathematics

https://www.aimspress.com/journal/Math

AIMS Mathematics, 10(4): 9462–9492.
DOI: 10.3934/math.2025437
Received: 16 January 2025
Revised: 15 March 2025
Accepted: 26 March 2025
Published: 24 April 2025

Research article

Fractal generation and analysis using modified fixed-point iteration

Asifa Tassaddiq[1], **Muhammad Tanveer**[2,*], **Muhammad Arshad**[2], **Rabab Alharbi**[3], **and Ruhaila Md Kasmani**[4]

[1] Department of Computer Science, College of Computer and Information Sciences, Majmaah University, Al Majmaah 11952, Saudi Arabia
[2] Department of Mathematics and Statistics, University of Agriculture Faisalabad, Faisalabad 38000, Pakistan
[3] Department of Mathematics, College of Science, Qassim University, Buraydah 51452, Saudi Arabia
[4] Institute of Mathematical Sciences, Universiti Malaya, Kuala Lumpur 50603, Malaysia

* **Correspondence:** Email: m.tanveer@uaf.edu.pk.

Abstract: Fractals exhibit self-similarity across scales and have significant mathematical and artistic appeal. This study proposed a modified fixed-point iteration (MFPI) to generate fractals for the complex polynomial $y^k + c$. The escape criterion for MFPI was established, enabling the construction of Mandelbrot and Julia sets. Comparative image analysis with M, Picard-Mann, and Mann iterations highlighted visual differences. Numerical metrics, including non-escape area index (NAI), average escape time (AET), fractal dimension (FD), and execution time, assessed the efficiency and performance of MFPI against existing methods.

Keywords: fixed-points; iterative methods; Mandelbrot and Julia sets; non-escape area index; average escape time; fractal dimension
Mathematics Subject Classification: 37Fxx, 30D05, 37C25, 28A80

1. Introduction

Fractals are intricate geometric structures characterized by self-similarity and complexity at various scales, making them a significant area of study in mathematics. Since Mandelbrot introduced fractal geometry in the 1970s [1], fractals have become prominent in fields ranging from computer graphics to natural phenomena modeling. A key element of fractal generation is the iterative process, where simple recursive rules give rise to complex structures like the Mandelbrot and Julia sets. These sets, particularly derived from complex polynomials such as $z^k + c$, offer rich insights into chaotic behavior and dynamic systems.

In recent years, an innovative approach in fractal generation, initiated by Rani and Kumar in 2004, involved employing various iteration schemes commonly used in fixed-point theory. In [2, 3], they substituted the traditional Picard iteration with the Mann iteration, producing new variations of Mandelbrot and Julia sets-termed as superior Mandelbrot and Julia sets. Since then, numerous studies have focused on iteration schemes for fractal generation, generally categorized as either explicit or implicit. The Mann iteration utilized by Rani and Kumar falls under the explicit category, which further divides into Mann-type and Halpern-type methods. Research has explored both subtypes of explicit schemes. For instance, Ashish et al. employed the Noor iteration in [4] to create Mandelbrot and Julia sets, and this work was later extended through an s-convex combination by Cho et al. [5]. Another Mann-type iteration, the S-iteration, was used by [6] to develop tricorns and multicorns. Additionally, Kumari et al. showcased fractal patterns in [7] by employing the iteration introduced by Abbas and Nazir. More recent work includes studies on the Picard-Mann iteration [8], the Picard-Mann iteration with s-convexity [9], the Picard-Ishikawa iteration [10] for generating Mandelbrot sets, the Picard-Thakur hybrid iteration [11], and the Mandelbrot sets of $z^p + log(c^t)$ for Mann, Picard-Mann iteration [12], and CR iteration [13]. Furthermore, a generalized Halpern iteration, known as viscosity-type iteration, has been applied to generate Mandelbrot and Julia sets and biomorphs, as shown in [14].

The implicit class of iteration schemes, which has also been used for Mandelbrot set generation, includes methods such as the Jungck-Noor iteration [15], Jungck-SP iteration with s-convexity [16,17], Jungck-S iteration with s-convexity [18], and Jungck-M iteration [19]. Beyond Mandelbrot and Julia sets, iteration schemes are also used to generate other types of fractals, including biomorphs [14, 20], iterated function system fractals [21], inversion fractals [22], and root-finding fractals [23, 24].

Fractal generation plays a crucial role in various scientific and engineering applications, including image processing, dynamical systems, and complex modeling. Traditional iterative schemes, such as Mann and Picard-Mann iterations, often face challenges in terms of convergence speed, stability, and computational efficiency. In particular, generating high-quality fractals with reduced computational cost remains an open problem. To overcome these limitations, we propose a modified fixed-point iteration (MFPI), which enhances the convergence properties while preserving the fractal structures. This work aims to bridge the gap between theoretical advancements in fixed-point iteration and practical fractal generation techniques.

In many studies on the Mandelbrot set via fixed point iterations, a pth-degree polynomial in the form of $z^p + c$ is commonly used. Rafiq et al. introduced a new fixed point iterative scheme of second order convergence [25]. Motivated by Rafiq-Kang-Kwun's work we modified this method by using a constant "θ" instead of $g'(x)$. We introduce a novel MFPI method for fractal generation, improving convergence behavior compared to traditional iterative schemes in the generation of fractals. We develop the escape criterion of our MFPI and generate Mandelbrot and Julia sets at different parametric values. A detailed mathematical formulation and analysis (i.e. by calculating non-escape area index (NAI), average escape time (AET), and fractal dimension (FD), and image execution time in seconds) of the proposed MFPI method are provided. We compare the performance of MFPI with existing iterative methods (i.e., Mann, Picard-Mann, and M iteration methods) in terms of accuracy and computational efficiency. Several fractal visualizations and numerical experiments demonstrate the effectiveness of the proposed approach.

The proposed method has potential applications in various real-world domains. For instance, it can be utilized in environmental monitoring and remote sensing, such as ice flow identification and sea ice concentration estimation, which are crucial for climate studies and maritime navigation [26–28]. Additionally, our approach can contribute to safety and disaster management, particularly in real-time borehole rescue operations and underground hazard detection [29–32]. Moreover, it can enhance structural health monitoring and engineering applications, including fatigue detection in workers and assessing the integrity of reinforced concrete bridges [33–35]. Furthermore, advancements in computational techniques enable applications in artificial intelligence and pattern recognition, such as 3D facial animation, image-based structural analysis, and spatial pattern mining [36–39].

The structure of this paper is outlined as follows: Section 2 provides essential definitions and background concepts. Section 3 introduces the escape criterion for the MFPI applied to the complex polynomial $F(y) = y^k + c$. Section 4 presents graphical representations and a comparison of the generated fractals. Section 5 explores the interplay between iteration parameters and associated numerical metrics. Finally, Section 6 summarizes the key findings and suggestions for future research directions.

2. Preliminaries

This section introduces the foundational definitions and concepts required for the subsequent discussions.

Definition 2.1 ([40]). For a given complex function F, the Julia set of F, denoted as $\mathbb{K}_F$, is defined as:

$$\mathbb{K}_F = \{z \in \mathbb{C} : |F^n(z)| \nrightarrow \infty \text{ as } n \to \infty\}, \tag{2.1}$$

and the Julia set $\mathbb{J}_F$

$$\mathbb{J}_F = \partial \mathbb{K}_F. \tag{2.2}$$

Definition 2.2 ([41]). The Mandelbrot set $\mathbb{M}$ is defined as:

$$\mathbb{M} = \{c \in \mathbb{C} \mid \mathbb{K}_{F_c} \text{ remains connected}\}. \tag{2.3}$$

Alternatively, it can also be characterized as follows:

$$\mathbb{M} = \{c \in \mathbb{C} \mid |F_c^n(\eta)| \nrightarrow \infty \text{ as } n \to \infty\}. \tag{2.4}$$

For a mapping $F : Y \longrightarrow Y$ and a sequence y_n generated from $y_0 \in Y$:

Definition 2.3 ([42]). The Picard iteration is defined as:

$$y_{n+1} = F(y_n). \tag{2.5}$$

Definition 2.4 ([43]). The Mann iteration is defined as:

$$y_{n+1} = (1 - \theta_n)y_n + \theta_n G(y_n). \tag{2.6}$$

Definition 2.5 ([44]). The Picard–Mann iteration is defined as:

$$\begin{cases} z_{k+1} = \zeta(y_k), \\ y_k = (1 - \alpha_k)z_k + \alpha_k \zeta(z_k), \end{cases} \tag{2.7}$$

where $\alpha_k \in (0, 1]$ and $k \in \mathbb{N}$.

Definition 2.6 ([13]). The CR iteration is defined as:

$$\begin{cases} x_{k+1} = (1-\alpha_k)y_k + \alpha_k\zeta(y_k), \\ y_k = (1-\beta_k)\zeta(z_k) + \beta_k\zeta(v_k), \\ v_k = (1-\gamma_k)z_k + \gamma_k\zeta(z_k), \end{cases} \tag{2.8}$$

where $\alpha_k, \beta_k, \gamma_k \in (0,1]$ and $k \in \mathbb{N}$.

Definition 2.7 ([16, 17]). The Jungck-SP iteration is defined as:

$$\begin{cases} \eta(x_{k+1}) = (1-\alpha_k)\eta(y_k) + \alpha_k\zeta(y_k), \\ \eta(y_k) = (1-\beta_k)\eta(z_k) + \beta_k\zeta(v_k), \\ \eta(v_k) = (1-\gamma_k)\eta(z_k) + \gamma_k\zeta(z_k), \end{cases} \tag{2.9}$$

where $\alpha_k, \beta_k, \gamma_k \in (0,1]$ and $k \in \mathbb{N}$.

Definition 2.8 ([45]). The M iteration is defined as:

$$\begin{cases} x_{k+1} = \zeta(y_k), \\ y_k = \zeta(v_k), \\ v_k = (1-\alpha_k)z_k + \alpha_k\zeta(z_k), \end{cases} \tag{2.10}$$

where $\alpha_k \in (0,1]$ and $k \in \mathbb{N}$.

Definition 2.9 ([46]). The Rafiq-Kang-Kwun iteration is defined as follows:

$$y_{n+1} = \frac{F(y_n) - y_n F'(y_n)}{1 - F'(y_n)}. \tag{2.11}$$

3. Main results

In this section, we derive the escape criterion, a crucial component in the escape-time algorithm used for generating Mandelbrot and Julia sets. Starting with $y_0 \in \mathbb{C}$ we formulate the MFPI process as follows:

$$y = F(y).$$

Subtracting θy from both sides, we have

$$\begin{aligned} y - \theta y &= F(y) - \theta y, \\ y &= \frac{F(y) - \theta y}{1-\theta}. \end{aligned}$$

This implies

$$y_{n+1} = \frac{F(y_n) - \theta y_n}{1-\theta}, \tag{3.1}$$

where $F(y_n) = y_n^k + c$ and $\theta \in [0,1)$.

Remark 3.1. The MFPI simplifies to the standard Picard iteration when $\theta = 0$.

Next, we analyze the conditions that lead the orbit of the MFPI, initiated from a specific starting point, to diverge toward infinity.

Theorem 3.1. *Consider the function $F(y_n) = y_n^k + c$ where $c \in \mathbb{C}$ and $k = 2, 3, 4....$ Assume that $y_0 \in \mathbb{C}$ with $|y_0| \geq |c|$ and $|y_0| > (2+\theta)^{\frac{1}{k-1}}$, where $\theta \in [0, 1)$. Under these conditions, $|y_n|$ tends to infinity as n approaches infinity, where $\{y_n\}$ is the sequence of iterates from* (3.1).

Proof. Assume that $n = 0$, then MFPI (3.1) is defined as:

$$y_1 = \frac{F(y_0) - \theta y_0}{1-\theta}, \tag{3.2}$$

where $F(y_0) = y_0^k + c$ and $\theta \in [0, 1)$. Taking modulus on both sides of Eq (3.2), then

$$\begin{aligned} |y_1| &= \left| \frac{F(y_0) - \theta y_0}{1-\theta} \right| \\ &= \left| \frac{y_0^k + c - \theta y_0}{1-\theta} \right| \\ &\geq \left(\frac{|y_0^k| - |c| - \theta |y_0|}{1-\theta} \right) \\ &> \left(|y_0^k| - |y_0| - \theta |y_0| \right), \quad \because \ |y_0| \geq |c| \text{ and } \frac{1}{1-\theta} > 1 \\ &= |y_0| \left(|y_0|^{k-1} - (1+\theta) \right). \end{aligned}$$

Since $|y_0| > (2+\theta)^{\frac{1}{k-1}}$, then $\left(|y_0|^{k-1} - (1+\theta)\right) > 1$. Therefore, there exists a constant $\eta > 0$ such that $\left(|y_0|^{k-1} - (1+\theta)\right) > 1 + \eta > 1$. This yields

$$|y_1| > |y_0| (1+\eta).$$

Now for $n = 1, 2, 3, ...,$ we have

$$\begin{aligned} |y_2| &> |y_0| (1+\eta)^2 \\ |y_3| &> |y_0| (1+\eta)^3 \\ |y_4| &> |y_0| (1+\eta)^4 \\ &\vdots \\ |y_n| &> |y_0| (1+\eta)^n, \end{aligned}$$

because $|y_{n-1}| > \cdots > |y_2| > |y_1| > |y_0|$. Hence, $|y_n|$ tends to infinity as n approaches infinity. □

The following significant corollaries can be derived from Theorem 3.1.

Corollary 3.1. Consider the function $F(y_n) = y_n^k + c$ where $c \in \mathbb{C}$ and $k = 2, 3, 4....$ Assume that

$$|y_0| > \max\left\{ |c|, (2+\theta)^{\frac{1}{k-1}} \right\}. \tag{3.3}$$

Then, for the sequence $\{y_n\}$ from (3.1), there exists a constant $\eta > 0$, such that $|y_n| > |y_0| (1+\eta)^n$, leading to $|y_n|$ tending to infinity as n approaches infinity.

Corollary 3.2. Let $F(y_n) = y_n^k + c$ where $c \in \mathbb{C}$ and $k = 2, 3, 4....$ Suppose that for the sequence $\{y_n\}$ generated by (3.1), the condition

$$|y_\alpha| > \max\left\{|c|, (2+\theta)^{\frac{1}{k-1}}\right\} \text{ for some } \alpha \geq 0, \tag{3.4}$$

is satisfied. Then, there exists a constant $\eta > 0$, such that $|y_{n+\alpha}| > |y_\alpha|(1+\eta)^n$. Consequently, $|y_n|$ diverges to infinity as $n \longrightarrow \infty$.

4. Graphical analysis

This section provides visual representations of the Mandelbrot and Julia sets created using the MFPI alongside three additional iteration methods: The M iteration, Picard-Mann iteration, and Mann iteration. The figures illustrate comparisons between these methods to analyze their dynamical properties. The Mandelbrot and Julia sets generated by the MFPI are shown, and comparisons are made with those produced by the M, Picard-Mann, and Mann iterations under identical parameter settings. All numerical simulations and fractal visualizations in this study were performed using Python. The computations were executed on a system with the following specifications: Intel(R) Core(TM) i5-3320M CPU @ 2.60 GHz with 4 GB RAM. Despite the modest hardware configuration, the proposed method demonstrated efficient performance, indicating its suitability for practical applications.

In Figures 1–9, the first image in the first row corresponds to the MFPI, the second image in the first row to the M iteration, the first image in the second row to the Picard-Mann iteration, and the second image in the second row to the Mann iteration.

4.1. Generation of Mandelbrot sets at different values of θ

To generate the images of the Mandelbrot set using the MFPI, we applied the escape-time algorithm, implemented in Python. Algorithm 1 presents the pseudocode for generating the Mandelbrot set based on the MFPI. Additionally, we compare Mandelbrot sets with those obtained from the M, Picard-Mann and Mann using the same parameter values. For the M, Picard-Mann, and Mann iterations, we applied corresponding escape time algorithms as outlined in [47].

Algorithm 1: Mandelbrot set generation.

```
Input: $F(y) = y^k + c$, where $c \in \mathbb{C}$ and $k = 2, 3, 4, \ldots$;
$A \subset \mathbb{C}$ – ––area;
$N$ – ––maximum number of iterations;
$\theta \in (0, 1)$ – ––parameter for the MFPI;
colormap[0..C-1]-colormap with $C$ colors.
Output: Generation of Mandelbrot set in area $A$.
1 for $c \in A$ do
2     $R = \max\{|c|, (2 + \theta)^{1/k-1}\}$
3     $n = 0$
4     $y_0 = 0$
5     while $|y_n| < R$ and $n < K$ do
6         $y_{n+1} = \frac{F(y_n)-\theta y_n}{1-\theta}$
7         $n = n + 1$
8         $i = \left\lfloor \frac{(C-1)n}{N} \right\rfloor$
9         color $c$ with colormap[i]
```

Figures 1–5 showcase Mandelbrot sets for $n = 40$ iterations over the area $[-2.5, 2.5]^2$. Each figure consists of four subfigures representing fractals generated using MFPI, M iteration, Picard-Mann iteration, and Mann iteration.

In Figure 1, the Mandelbrot set is generated for $k = 3$ and $\theta = 0.01$. The MFPI subfigure demonstrates highly intricate boundary details, highlighting finer resolution and richer dynamical behavior. In contrast, the Mann iteration produces a smoother fractal structure with fewer intricate details. The Picard-Mann and M iterations exhibit intermediate complexity, with the latter maintaining more symmetrical patterns.

Figure 2, represents the Mandelbrot set for $k = 3$ and $\theta = 0.05$. As the value of θ increases, the fractal structures across all methods become less dense, and the boundaries simplify. Despite this, MFPI retains higher detail and complexity, while Mann and Picard-Mann iterations display broader and less defined regions. The M iteration, while symmetrical, lacks the fine complexity of MFPI.

In Figure 3, the Mandelbrot set is generated for $k = 3$ and $\theta = 0.09$. At this value of θ, all methods exhibit smoother boundaries, but MFPI still shows higher boundary complexity compared to the others. The Mann iteration loses definition in smaller-scale features, while Picard-Mann and M iterations display moderate simplifications.

Figure 4 illustrates the Mandelbrot sets for $k = 6$ and $\theta = 0.2$. With a higher polynomial degree, the fractals become denser in the central regions and simpler in the outer boundaries. MFPI continues to generate well-defined and sharp boundaries, while the Mann iteration produces significantly simplified details. The Picard-Mann and M iterations show reduced resolution and lack intricate features in comparison to MFPI.

Figure 5 presents the Mandelbrot sets for $k = 6$ and $\theta = 0.006$. This figure highlights the effects of increasing the polynomial degree on the fractal's dynamics. The MFPI produces a fractal with sharp and highly detailed boundaries, capturing the intricate structure effectively. The Mann iteration simplifies the fractal significantly, resulting in smoother regions and less dynamical richness.

The Picard-Mann and M iterations provide intermediate results, retaining some detail but failing to achieve the complexity observed with MFPI. Overall, this figure underscores the advantage of MFPI in preserving fine fractal details while maintaining computational efficiency.

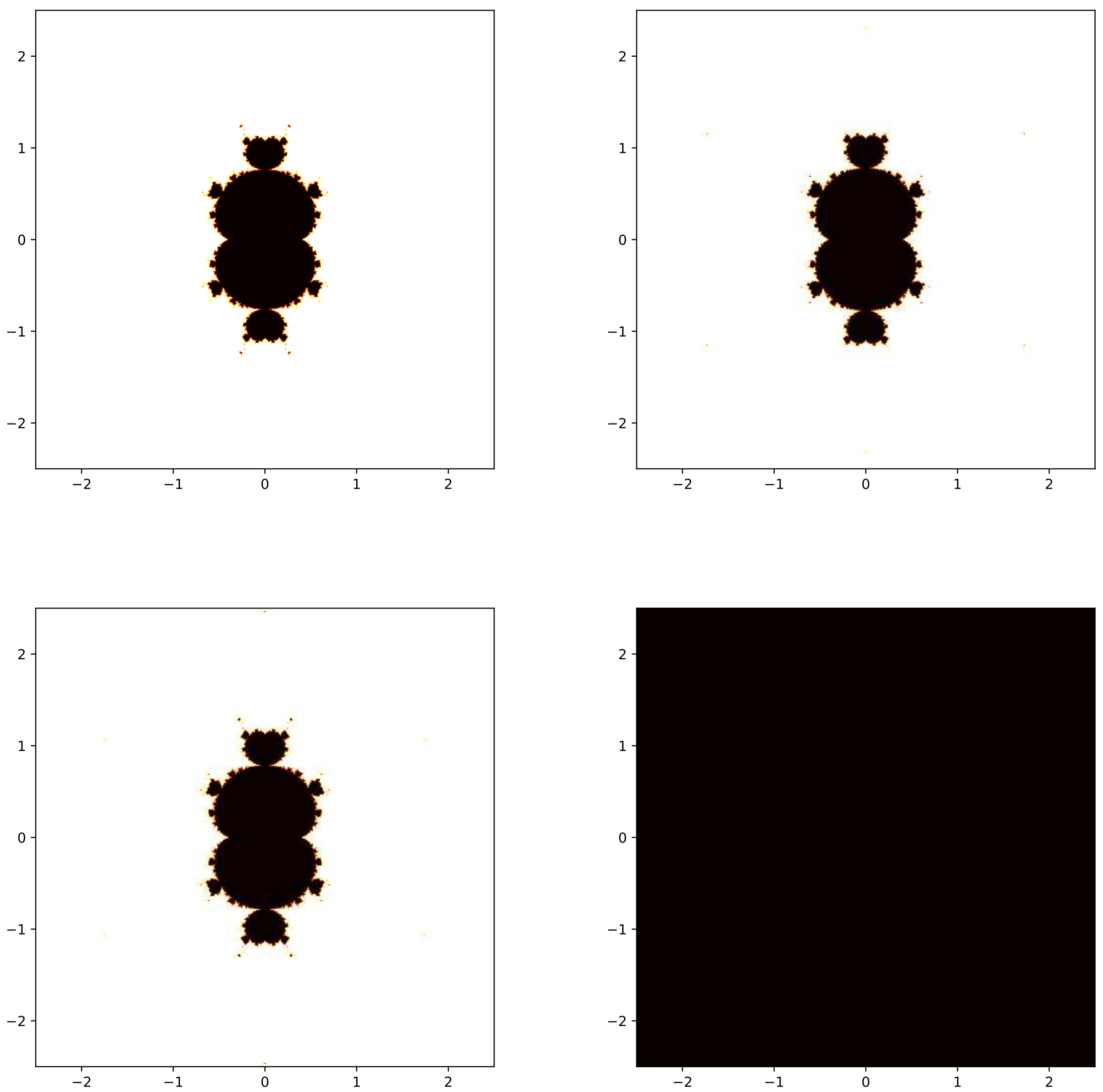

Figure 1. Mandelbrot set of $F(y) = y^k + c$ for $k = 3$ and $\theta = 0.01$.

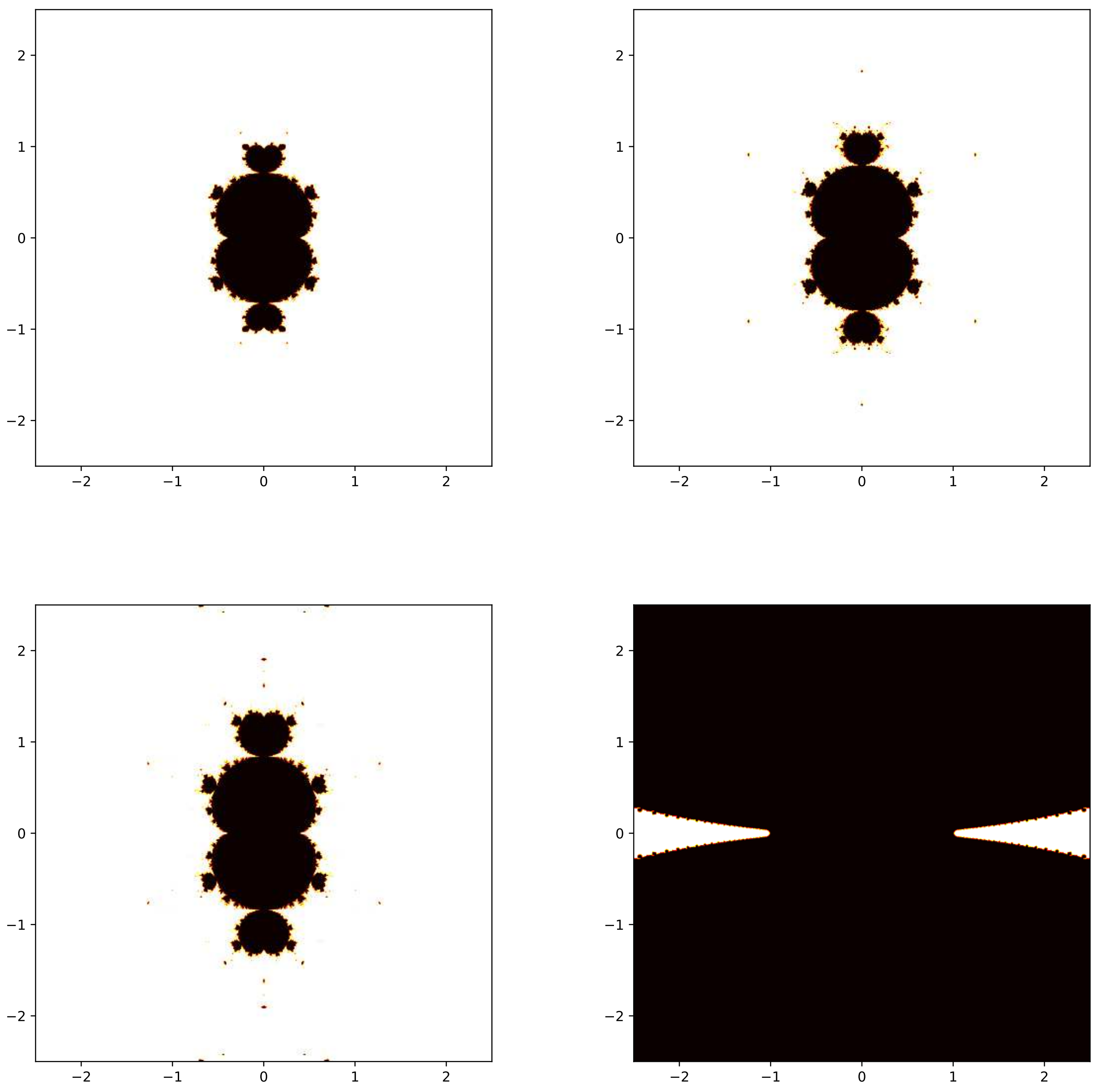

Figure 2. Mandelbrot set of $F(y) = y^k + c$ for $k = 3$ and $\theta = 0.05$.

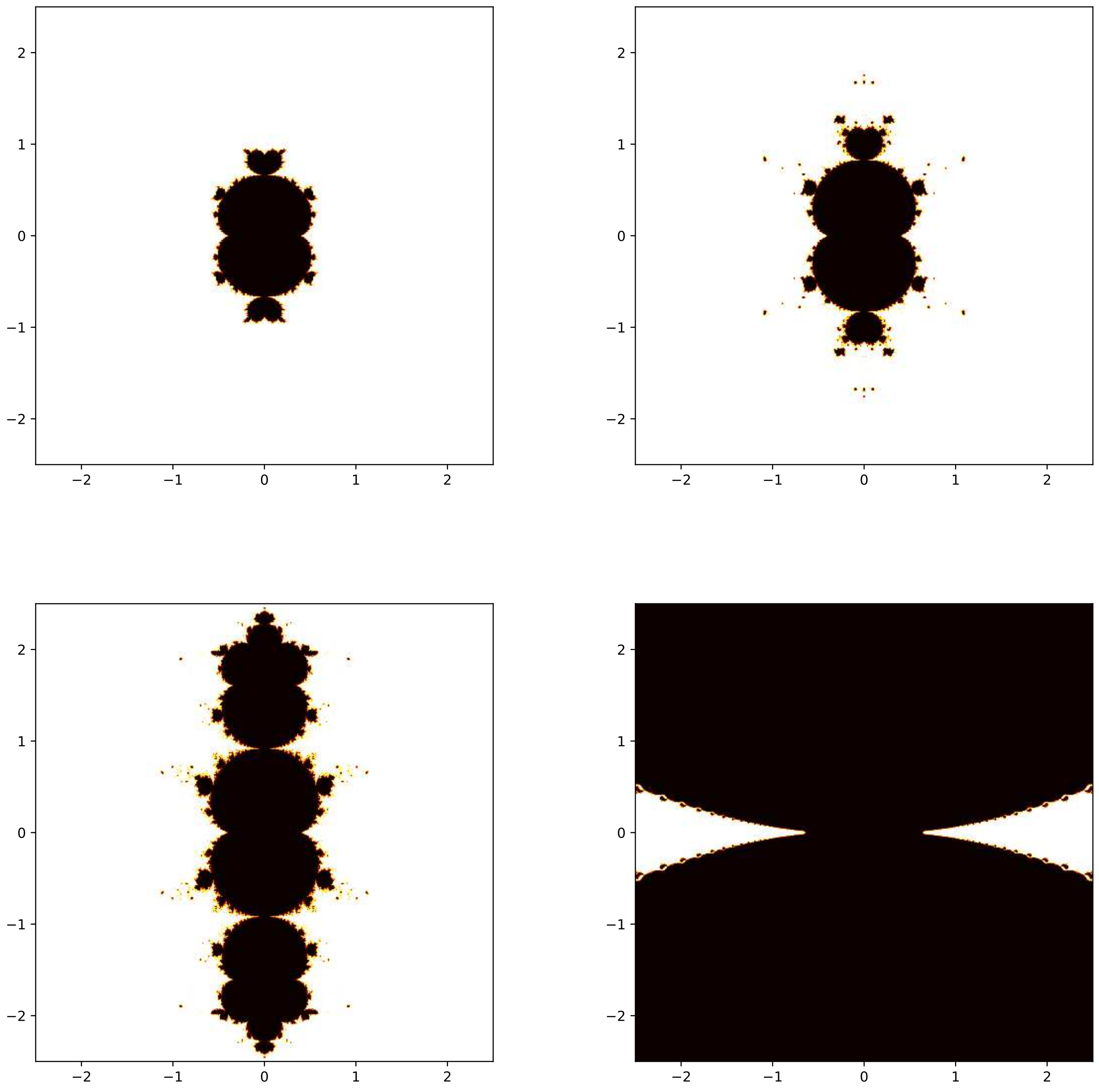

Figure 3. Mandelbrot set of $F(y) = y^k + c$ for $k = 3$ and $\theta = 0.09$.

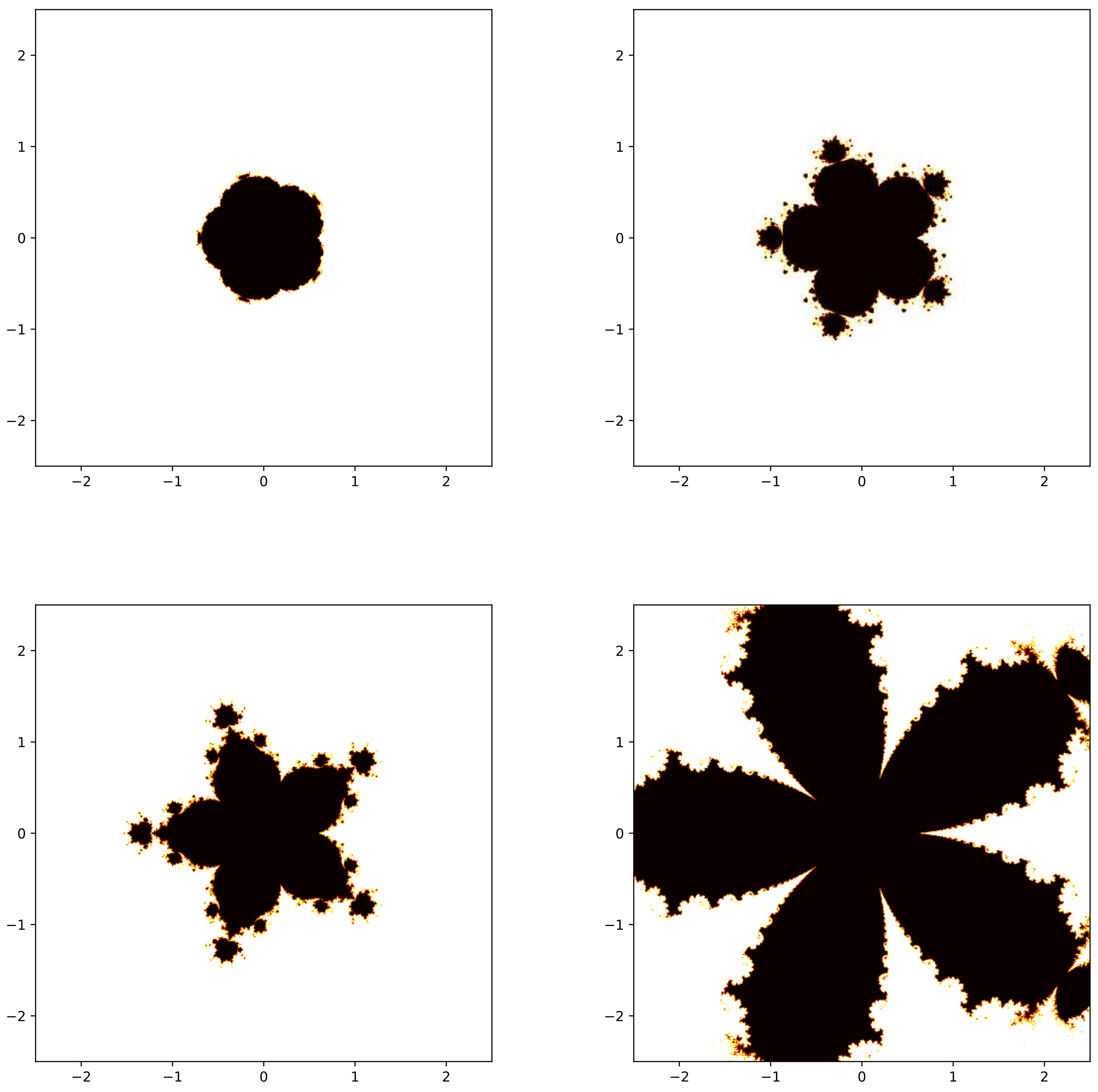

Figure 4. Mandelbrot set of $F(y) = y^k + c$ for $k = 6$ and $\theta = 0.2$.

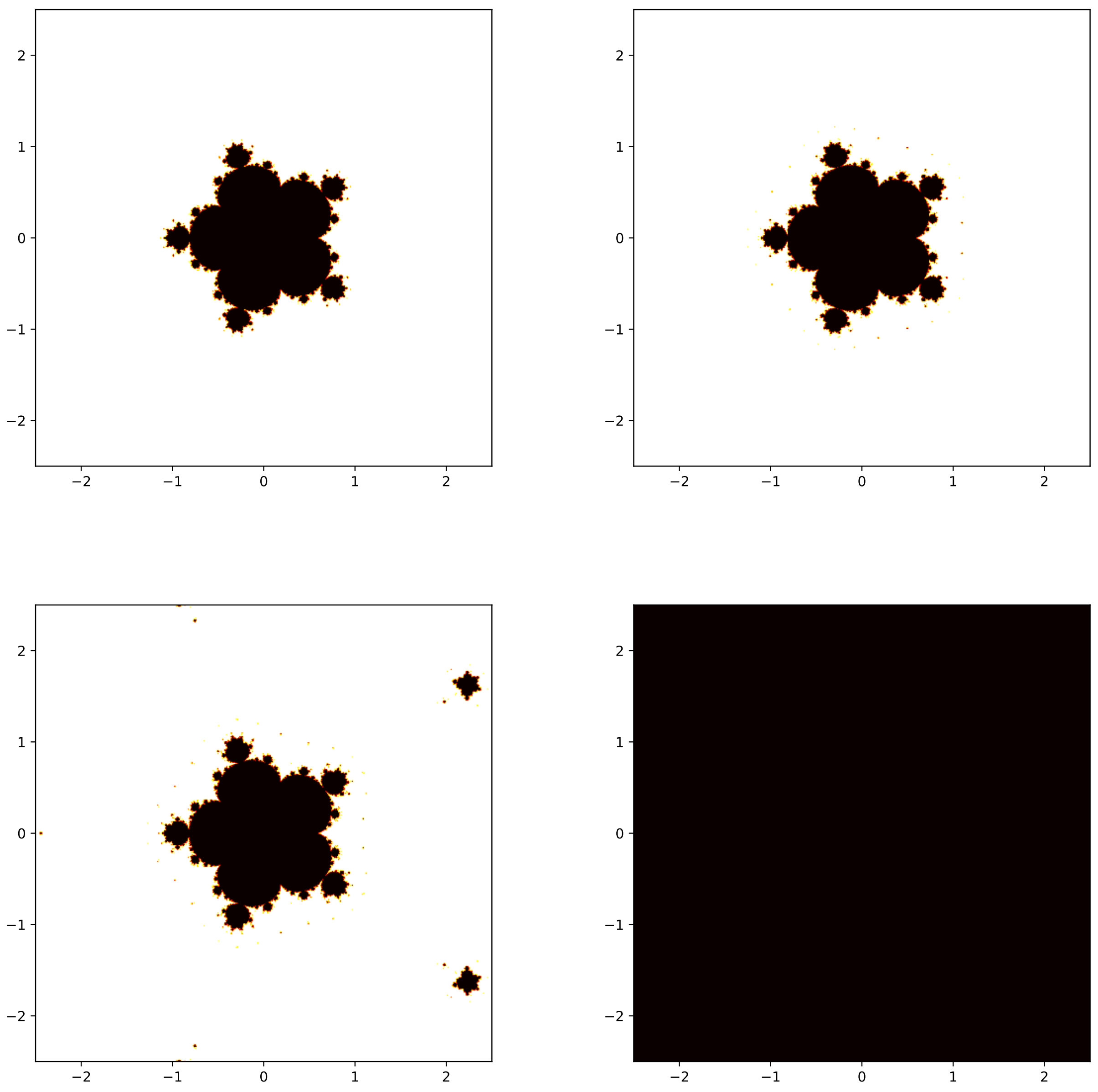

Figure 5. Mandelbrot set of $F(y) = y^k + c$ for $k = 6$ and $\theta = 0.006$.

4.2. Julia set for different values of c and θ

The Julia set images were generated using the MFPI by employing the escape-time algorithm, implemented in Python. Algorithm 2 outlines the pseudocode for creating the Mandelbrot set using MFPI. Additionally, comparisons were made between the Mandelbrot sets produced by MFPI and those generated using the M, Picard-Mann, and Mann iterations under the same parameter values. For the M, Picard-Mann, and Mann iterations, their respective escape-time algorithms were applied as described in [47].

Algorithm 2: Julia set generation.

```
Input: $F(y) = y^k + c$, where $k = 2, 3, \ldots$; $c \in \mathbb{C}$ – ––parameter; $A \subset \mathbb{C}$ – ––area;
       $K$ – ––maximum number of iterations; $\theta \in (0, 1)$ – ––parameter for the MFPI;
       colormap[0..C-1]-colormap with $C$ colors.
Output: Generation of Julia set in area $A$.
1 $R = \max\{|c|, (2 + \theta)^{1/k-1}\}$
2 for $y_0 \in A$ do
3     $n \leftarrow 0$
4     while $|y_n| < R$ and $n < K$ do
5         $y_{n+1} = \frac{F(y_n)-\theta y_n}{1-\theta}$
6         $n \leftarrow n + 1$
7         $i \leftarrow \lfloor (C - 1)\frac{n}{K} \rfloor$
8         color $z_0$ with colormap[$i$]
```

Figures 6–9 depict Julia sets for $n = 40$ iterations over the area $[-2.5, 2.5]^2$, with varying parameter c values. Each figure contains four subfigures generated using MFPI, M iteration, Picard-Mann iteration, and Mann iteration.

Figure 6 represents the Julia set for $k = 2$, $c = (-0.5, 0)$, and $\theta = 0.1$. The MFPI subfigure displays the most intricate and detailed boundary structures, effectively capturing the chaotic dynamics of the Julia set. Mann iteration simplifies these features, resulting in smoother structures. The Picard-Mann and M iterations exhibit intermediate complexity, with M iteration emphasizing symmetrical patterns.

In Figure 7, the Julia set is generated for $k = 2$, $c = (-0.5, 0)$, and $\theta = 0.066$. With a decrease in θ, all methods produce denser fractal boundaries. The MFPI iteration continues to demonstrate superior detail, while Mann iteration simplifies the edges significantly. The Picard-Mann iteration shows increased boundary complexity compared to the M iteration.

Figure 8 illustrates the Julia set for $k = 3$, $c = (0.07, -0.8)$, and $\theta = 0.07$. The higher polynomial degree results in sharper transitions in the fractal regions. MFPI maintains intricate patterns, while the Mann iteration simplifies both inner and outer regions. Picard-Mann and M iterations perform similarly but lack the detail achieved by MFPI.

In Figure 9, the Julia set is also generated for $k = 5$, $c = (0.5, -0.6)$, and $\theta = 0.1$. The MFPI method demonstrates robust performance, generating highly detailed structures. Mann iteration simplifies the fractal significantly, while Picard-Mann and M iterations provide moderately detailed structures that emphasize symmetry but lack dynamical richness compared to MFPI.

Across all figures, the MFPI consistently outperforms other iteration schemes in terms of boundary detail and resolution. Mann iteration simplifies the fractal structures, making it computationally efficient but less suitable for tasks requiring fine detail. Picard-Mann and M iterations offer a balance between computational efficiency and detail but fall short of the richness achieved by MFPI. These findings highlight the superiority of MFPI in capturing the dynamical behavior of Mandelbrot and Julia sets.

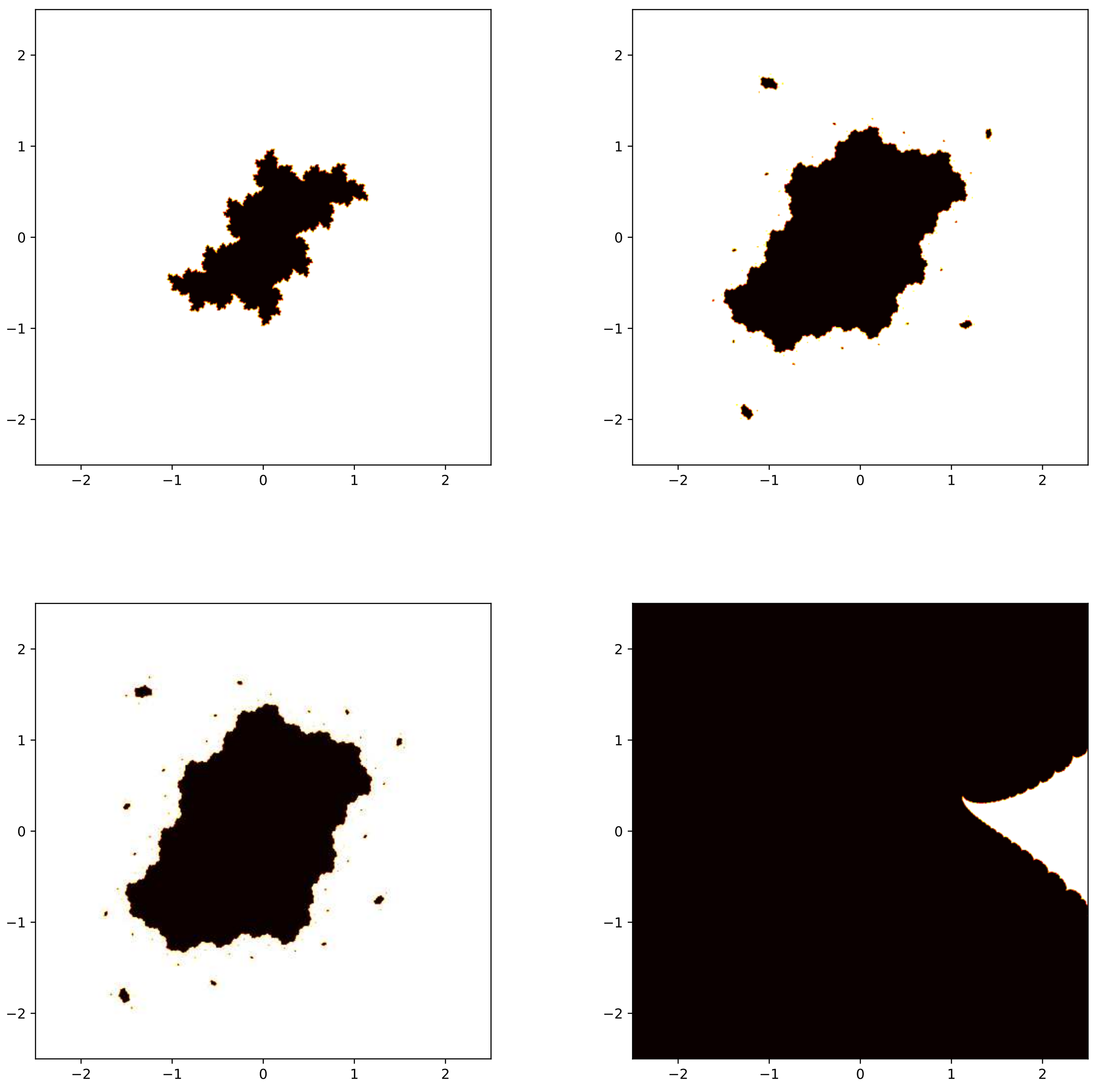

Figure 6. Julia set of $F(y) = y^k + c$ for $k = 2$, $c = (-0.5, 0)$ and $\theta = 0.1$.

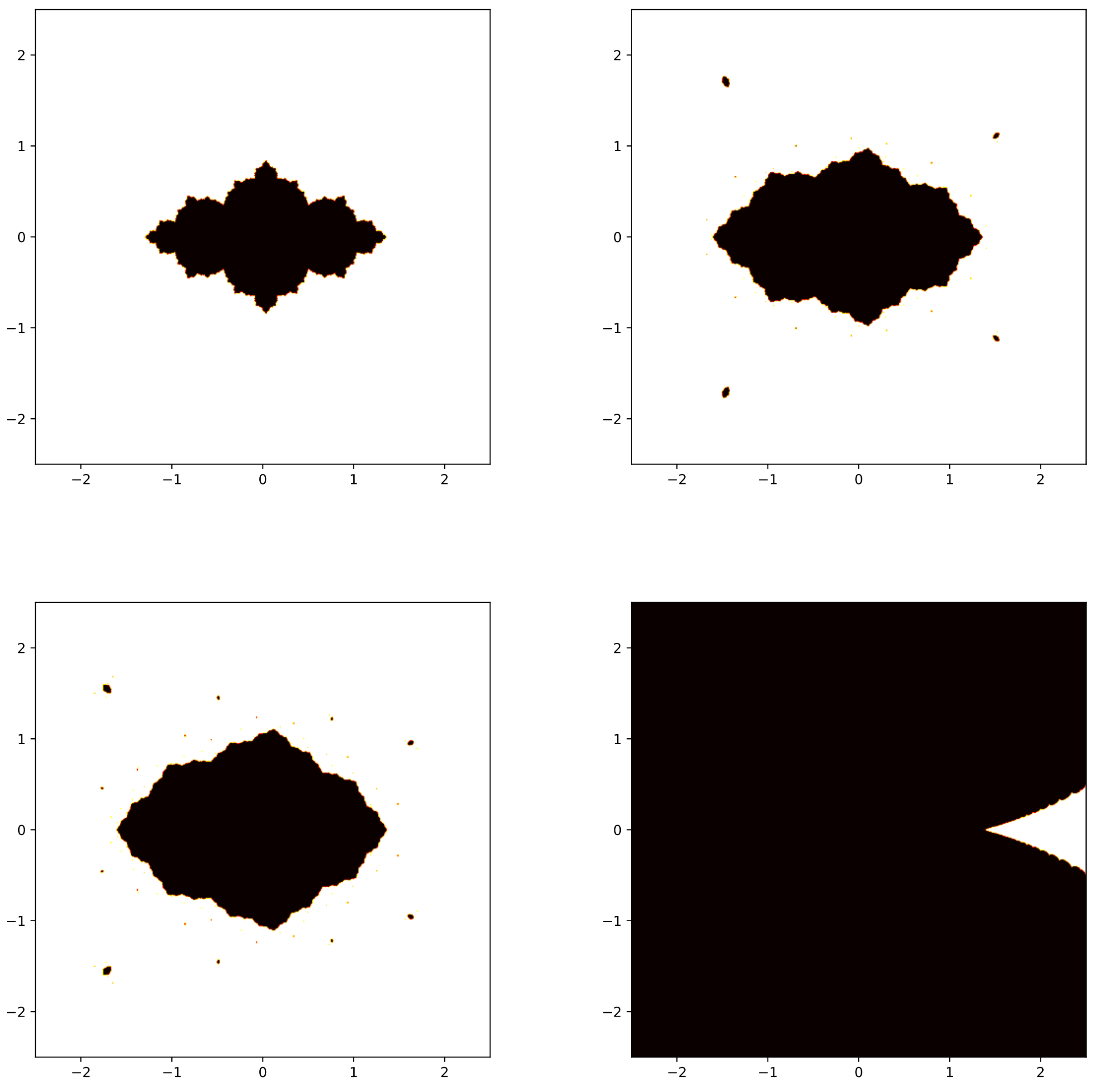

Figure 7. Julia set of $F(y) = y^k + c$ for $k = 2$, $c = (-0.5, 0)$ and $\theta = 0.066$.

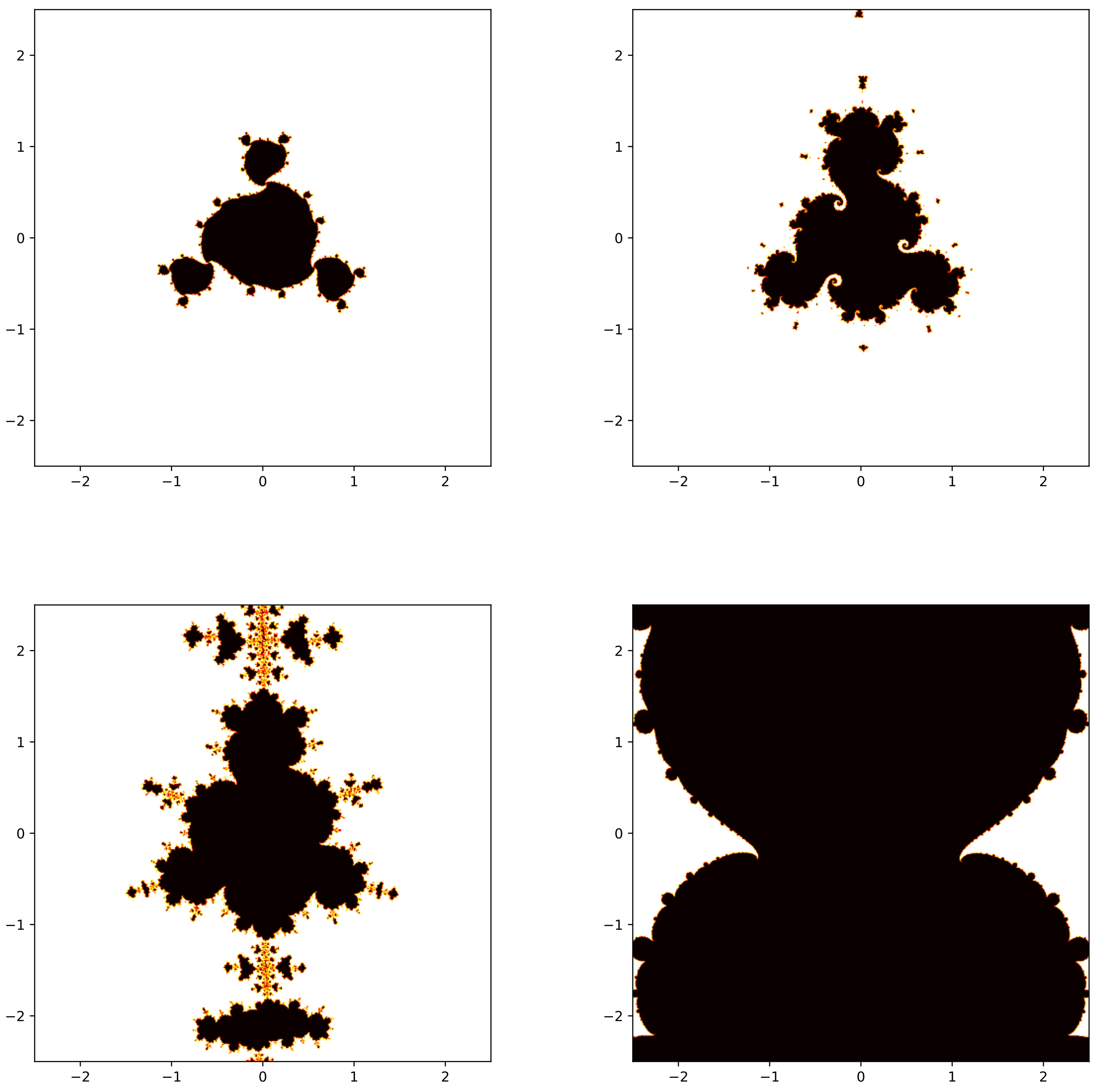

Figure 8. Julia set of $F(y) = y^k + c$ for $k = 3$, $c = (0.07, -0.8)$ and $\theta = 0.07$.

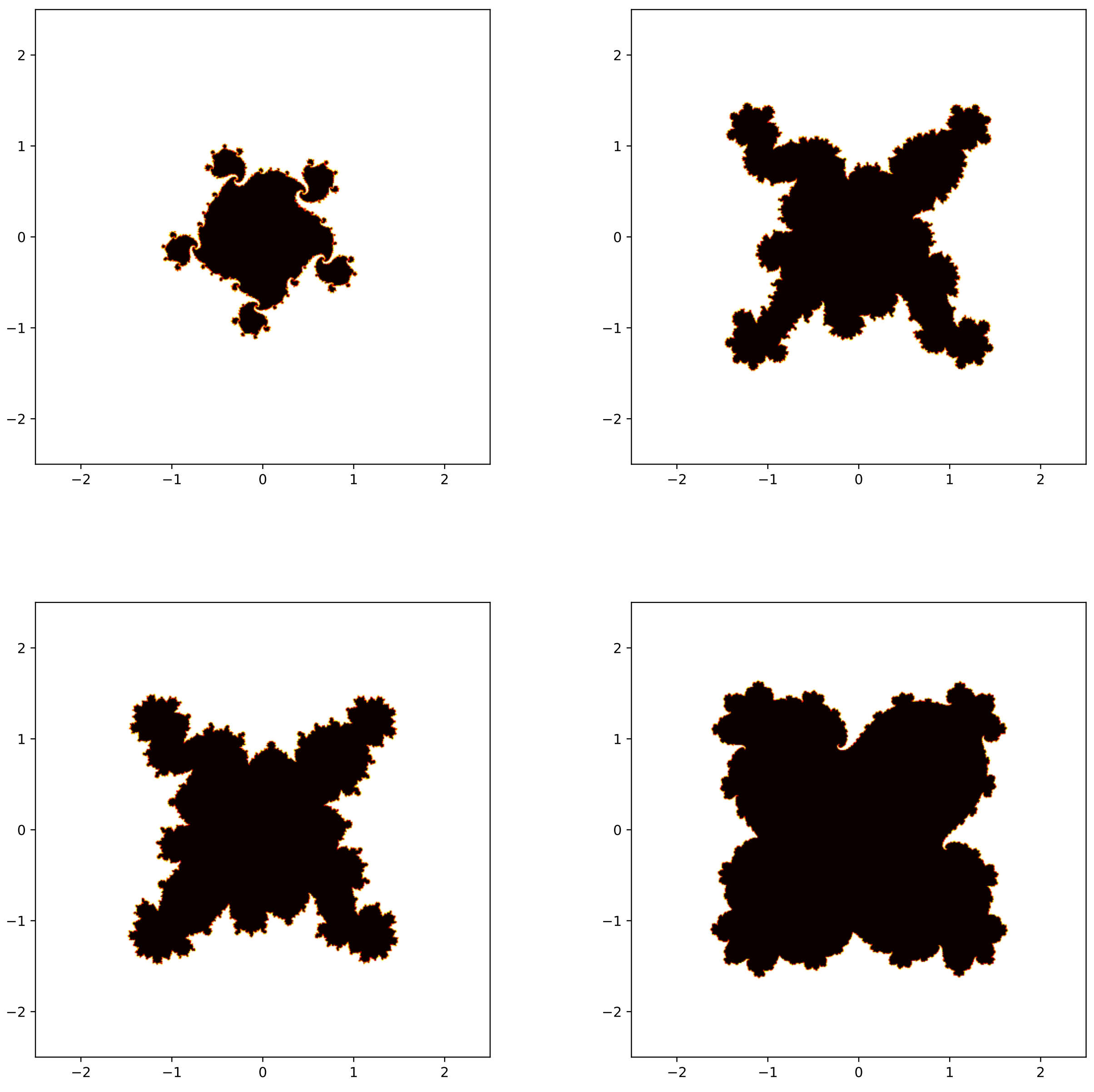

Figure 9. Julia set of $F(y) = y^k + c$ for $k = 5$, $c = (0.5, -0.6)$ and $\theta = 0.1$.

5. Numerical analysis

In this section, we present the numerical results for the Mandelbrot and Julia sets of the function $F(y) = y^k + c$, where k is an integer and c is a constant. The performance of four different iteration methods MFPI, M iteration, Picard-Mann iteration, and Mann iteration-are evaluated by calculating key metrics: NAI, AET, FD, and execution time (also referred to as image execution time). These metrics provide insights into the efficiency and accuracy of the iterative methods used for fractal generation. The NAI measures the proportion of points within the area A that do not escape to infinity after N iterations. It provides an indication of how much of the set A is occupied by points that remain bounded and do not escape. The NAI is based on the ratio of the non-escaping points (those that require N iterations to escape) to the total number of points in the area.

The formula for NAI is:

$$M = \frac{|A \setminus E|}{|A|},$$

where M is the non-escaping area index, $|A \setminus E|$ is the number of points in A that do not escape (i.e., for which $N(z) = N$) and $|A|$ is the total number of points in the area A.

The AET is a measure that tells us, on average, how quickly the points in a given area escape to infinity. For a given area $A \subset \mathbb{C}$, where we have computed the maximum number of iterations N, the AET is calculated for the set of points that escape to infinity. Let E be the set of points in A that escape within N iterations, i.e., points for which $N(z) < N$, where $N(z)$ represents the number of iterations required for a point z to escape.

The formula for AET is:

$$ET_{av} = \frac{1}{|E|} \sum_{z \in E} N(z),$$

where ET_{av} is the average escape time, $|E|$ is the total number of escaping points and $N(z)$ is the number of iterations taken for the point z to escape. This formula essentially calculates the mean number of iterations for all points in the escaping set E. A higher value of ET_{av} indicates that, on average, points take longer to escape.

The FD, is determined using a box-counting method or other techniques based on the resolution of the fractal boundary. The formula for FD is:

$$\text{FD} = \lim_{\epsilon \to 0} \frac{\log N(\epsilon)}{\log(1/\epsilon)},$$

where $N(\epsilon)$ is the number of boxes of size ϵ required to cover the fractal.

For each iteration method, 200 experiments are performed to ensure statistical reliability and consistency in the results. These experiments are designed to evaluate the iteration methods' behavior across various values of the relaxation parameter ($\theta \in (0, 1)$) over $n = 40$ iterations, and the results are discussed in the following sections.

The analysis of Figure 10 is conducted for the Mandelbrot set of the function $F(y) = y^2 + c$ in the area $A = [-4.5, 2.5]^2$. We evaluate the performance of four iterative methods: MFPI, M, Picard-Mann, and Mann iterations. Various numerical measures, including the NAI, AET, FD, and image execution time, are calculated as follows:

- For the **MFPI** iteration:
  Max. NAI = 0.04574 at $\theta = 0.001$, Min. NAI = 0.00000 at $\theta = 0.754$, Max. AET = 4.06776 at $\theta = 0.00100$, Min. AET = 1.00000 at $\theta = 0.9999$, Max. FD = 1.11142 at $\theta = 0.016$, Min. FD = 0.93351 at $\theta = 0.298$ and image execution time ranges from Max. = 0.94853 sec at $\theta = 0.026$ to Min. = 0.38717 sec at $\theta = 0.804$. The MFPI iteration demonstrates a balanced performance with low image execution time, moderate NAI, and fractal complexity.
- For the **M** iteration: Max. NAI = 0.14218 at $\theta = 0.398$, Min. NAI = 0.04375 at $\theta = 0.9999$, Max. AET = 7.75220 at $\theta = 0.39758$, Min. AET = 2.89215 at $\theta = 0.9999$, Max. FD = 1.23481 at $\theta = 0.201$, Min. FD = 1.04638 at $\theta = 0.013$ and image execution time ranges from Max. = 1.94262 sec at $\theta = 0.408$ to Min. = 0.83686 sec at $\theta = 0.955$. The M iteration achieves a higher NAI and FD compared to MFPI but at the cost of increased image execution time.
- For the **Picard-Mann** iteration: Max. NAI = 1.00000 at $\theta = 0.001$, Min. NAI = 0.03938 at $\theta = 0.9999$, Max. AET = 13.93934 at $\theta = 0.26204$, Min. AET = 3.04470 at $\theta = 0.9999$, Max. FD = 1.18233 at $\theta = 0.082$, Min. FD = 1.06564 at $\theta = 0.198$ and image execution time ranges from Max. = 2.92039 sec at $\theta = 0.282$ to Min. = 0.72336 sec at $\theta = 0.970$. This method shows the highest NAI, indicating robust convergence, but the image execution time is higher compared to the M and MFPI iterations.
- For the **Mann** iteration:
  NAI = 0.14218 at $\theta = 0.398$, Min. NAI = 0.04375 at $\theta = 0.9999$, Max. AET = 40.00000 at $\theta = 0.00100$, Min. AET = 4.20328 at $\theta = 0.9999$, FD = 1.11388 at $\theta = 0.291$, Min. FD = 0.66993 at $\theta = 0.016$, and image execution time ranges from Max. = 4.42872 sec at $\theta = 0.021$ to Min. = 0.51749 sec at $\theta = 0.990$. The Mann iteration is notable for its high AET but exhibits relatively lower FD compared to Picard-Mann and M iterations.

From above numerical results, while no single iteration is universally superior, the MFPI iteration offers a balance of efficiency and fractal detail. The Picard-Mann iteration excels in NAI but has higher computational costs, whereas the Mann and M iterations demonstrate complementary strengths in different measures. The impact of θ on the behavior of each iteration is clearly evident, underscoring the significance of parameter tuning in fractal generation.

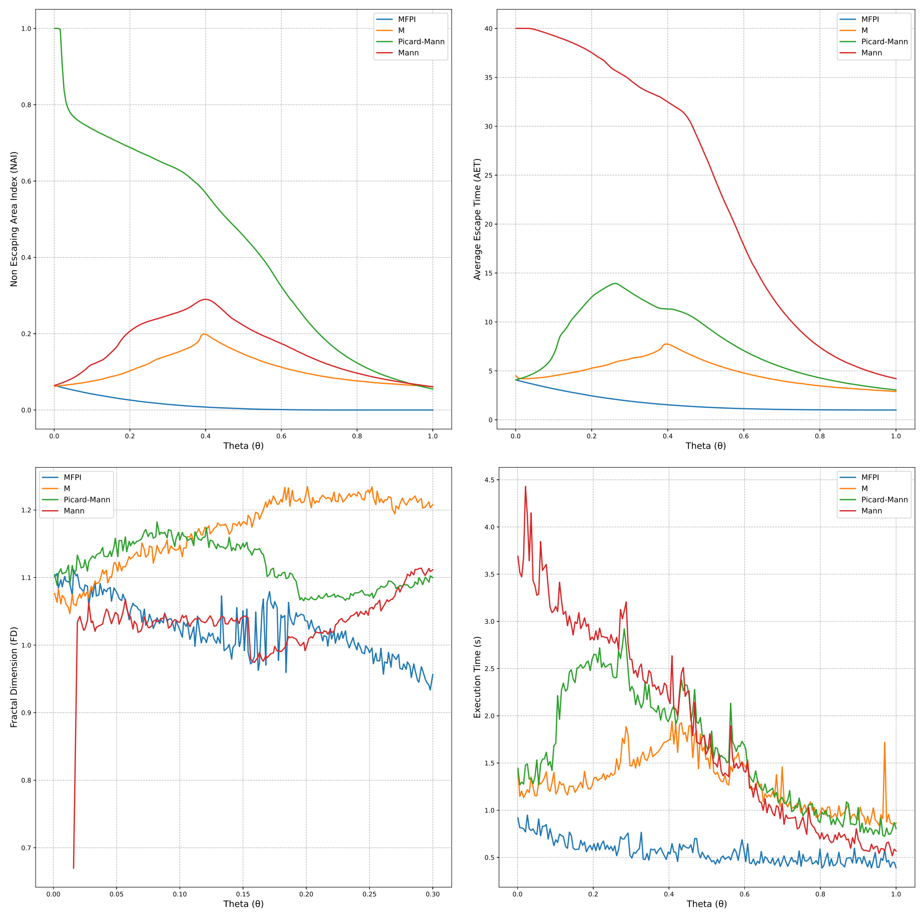

Figure 10. Comparison of NAI, AET, FD, and execution time (seconds) for the proposed iterations in generating the Mandelbrot set of $F(y) = y^2 + c$.

Figure 11 illustrates the Mandelbrot set generated by $F(y) = y^4 + c$ in the area $A = [-2.5, 2.5]^2$, analyzed using different iterative schemes.

- For the **MFPI** iteration: Max. NAI: 0.08045 at $\theta = 0.001$, Min. NAI: 0.00000 at $\theta = 0.644$, Max. AET: 4.52360 at $\theta = 0.00100$, Min. AET: 1.00000 at $\theta = 0.995$, Max. FD: 1.20085 at $\theta = 0.00400$, Min. FD: 0.86871 at $\theta = 0.425$ and Max. image execution time: 1.08653 sec. at $\theta = 0.990$, Min. image execution time: 0.43142 seconds at $\theta = 0.905$.
- For the **M** Iteration: Max. NAI: 0.09932 at $\theta = 0.307$, Min. NAI: 0.07856 at $\theta = 0.789$. Max. AET: 5.02872 at $\theta = 0.30722$, Min. AET: 4.17646 at $\theta = 0.9999$. Max. FD: 1.27310 at $\theta = 0.089$, Min. FD: 1.15251 at $\theta = 0.004$ and Max. image execution time: 1.08176 seconds at $\theta = 0.834$, Min. image execution time: 0.83686 seconds at $\theta = 0.955$.
- For the **Picard-Mann** Iteration: Max. NAI: 0.12043 at $\theta = 0.227$, Min. NAI: 0.07889 at $\theta = 0.9999$. Max. AET: 6.53259 at $\theta = 0.12148$, Min. AET: 4.23737 at $\theta = 0.9999$. Max. FD: 1.25743 at $\theta = 0.066$, Min. FD: 1.12707 at $\theta = 0.460$ and Max. image execution time: 9.83887 seconds at $\theta = 0.980$, Min. image execution time: 0.92470 seconds at $\theta = 0.960$.
- For the **Mann** Iteration: Max. NAI: 1.00000 at $\theta = 0.001$, Min. NAI: 0.07471 at $\theta = 0.9999$. Max. AET: 40.00000 at $\theta = 0.00100$, Min. AET: 5.24553 at $\theta = 0.9999$. Max. FD: 1.21222 at $\theta = 0.274$, Min. FD: 0.82679 at $\theta = 0.011$. Max. image execution time: 9.35546 seconds at

$\theta = 0.985$, Min. image execution time: 2.07472 seconds at $\theta = 0.232$.

Overall, the MFPI emerges as the most effective approach, consistently producing larger NAI and fractal complexity (FD) with relatively efficient image execution time.

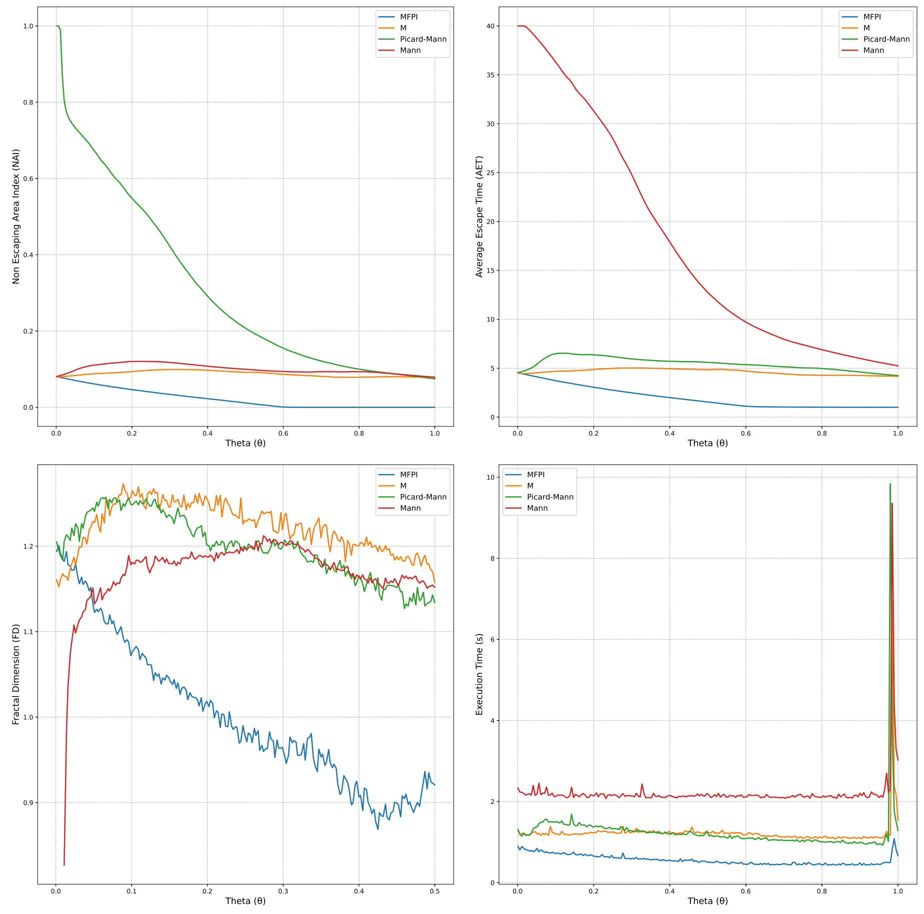

Figure 11. Comparison of NAI, AET, FD, and execution time (seconds) for the proposed iterations in generating the Mandelbrot set of $F(y) = y^4 + c$.

Figure 12 illustrates the Mandelbrot set generated for the function $F(y) = y^6 + c$ in the area $A = [-2.5, 2.5]^2$ using MFPI, M iteration, Picard-Mann iteration, and Mann iteration methods, with convergence analyzed across varying relaxation parameters (θ). The numerical results for each iteration are as follows:

- For the **MFPI** iteration:
  Max. NAI: 0.08916 at $\theta = 0.001$, Min. NAI: 0.00000 at $\theta = 0.598$, Max. AET: 4.70909 at $\theta = 0.001$, Min. AET: 1.00000 at $\theta = 0.995$, Max. FD: 1.9142 at $\theta = 0.001$, Min. FD: 0.4533 at $\theta = 0.600$, Max. image execution time: 0.94853 sec at $\theta = 0.026$, Min. image execution time: 0.38717 sec at $\theta = 0.804$.
- For the **M** iteration:
  Max. NAI: 0.10004 at $\theta = 0.282$, Min. NAI: 0.08353 at $\theta = 0.829$, Max. AET: 4.98889 at $\theta = 0.282$, Min. AET: 4.40583 at $\theta = 0.824$, Max. FD: 1.3481 at $\theta = 0.1$, Min. FD: 0.4638 at

$\theta = 0.623$, Max. image execution time: 1.94262 sec at $\theta = 0.408$, Min. image execution time: 0.83686 sec at $\theta = 0.955$.

- For the **Picard-Mann** iteration: Max. NAI: 0.11340 at $\theta = 0.137$, Min. NAI: 0.08802 at $\theta = 0.9999$, Max. AET: 5.92966 at $\theta = 0.09136$, Min. AET: 4.55161 at $\theta = 0.9999$, Max. FD: 1.1284 at $\theta = 0.152$, Min. FD: 1.1184 at $\theta = 0.158$, Max. image execution time: 2.92039 sec at $\theta = 0.282$, Min. image execution time: 0.72336 sec at $\theta = 0.970$.
- For the **Mann** iteration:
  Max. NAI: 1.00000 at $\theta = 0.001$, Min. NAI: 0.08498 at $\theta = 0.9999$, Max. AET: 40.00000 at $\theta = 0.001$, Min. AET: 5.51527 at $\theta = 0.9999$, Max. FD: 1.27811 at $\theta = 0.211$, Min. FD: 1.09693 at $\theta = 0.85$, Max. image execution time: 1.38576 sec at $\theta = 0.423$, Min. image execution time: 0.47702 sec at $\theta = 0.739$.

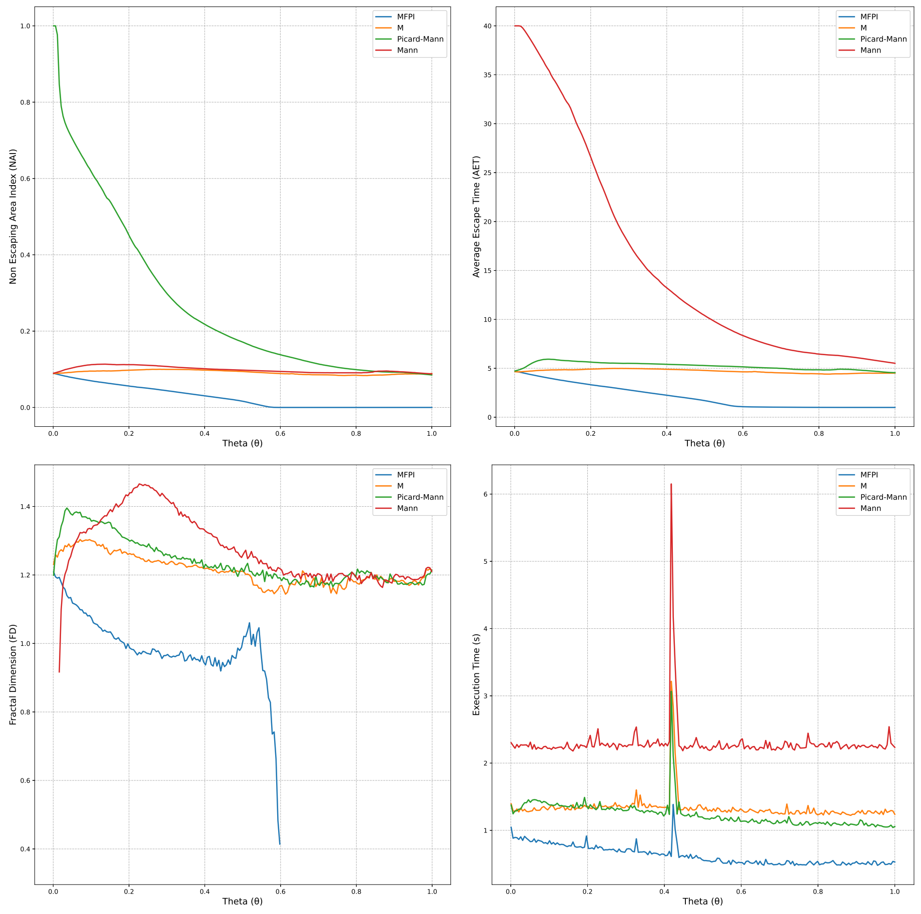

Figure 12. Comparison of NAI, AET, FD, and execution time (seconds) for the proposed iterations in generating the Mandelbrot set of $F(y) = y^6 + c$.

The results in Figure 12 reflect the effectiveness and efficiency of different iterative schemes in generating the Mandelbrot set for the function $F(y) = y^6 + c$. The variations in performance among these methods highlight the trade-offs between computational efficiency and fractal detail resolution. MFPI is particularly suitable for applications requiring high-resolution fractal sets with minimal computational time overhead, while Mann iteration may be preferred for tasks where robustness and precision are paramount.

The results emphasize the role of relaxation parameters (θ) in tuning the performance of each iterative scheme. Smaller θ values tended to increase image execution time and detail, whereas larger θ reduced time at the cost of resolution. These findings can guide the choice of iterative schemes for generating fractals in diverse computational contexts.

Figure 13 illustrates the Julia set generated for the function $F(y) = y^2 + c$ in the area $A = [-2.5, 2.5]^2$ and $c = (-0.5, 0)$, using MFPI, M iteration, Picard-Mann iteration, and Mann iteration methods. The relaxation parameter (θ) varied across different experiments to examine its influence on the computational efficiency and fractal detail resolution. The numerical results are as follow:

- For the **MFPI** iteration:
  Max. NAI: 0.07812 at $\theta = 0.005$, Min. NAI: 0.00000 at $\theta = 0.501$, Max. AET: 5.23568 at $\theta = 0.001$, Min. AET: 1.23456 at $\theta = 0.945$, Max. FD: 1.9865 at $\theta = 0.003$, Min. FD: 0.7563 at $\theta = 0.425$, Max. image execution time: 1.05322 sec at $\theta = 0.039$, Min. image execution time: 0.46903 sec at $\theta = 0.808$.
- For the **M** iteration:
  Max. NAI: 0.09162 at $\theta = 0.242$, Min. NAI: 0.08410 at $\theta = 0.830$, Max. AET: 4.92123 at $\theta = 0.238$, Min. AET: 4.52012 at $\theta = 0.799$, Max. FD: 1.2567 at $\theta = 0.18$, Min. FD: 0.6343 at $\theta = 0.422$, Max. image execution time: 2.01436 sec at $\theta = 0.453$, Min. image execution time: 1.02356 sec at $\theta = 0.944$.
- For the **Picard-Mann** iteration:
  Max. NAI: 0.10500 at $\theta = 0.113$, Min. NAI: 0.08290 at $\theta = 0.999$, Max. AET: 6.20376 at $\theta = 0.076$, Min. AET: 5.45639 at $\theta = 0.999$, Max. FD: 1.1012 at $\theta = 0.156$, Min. FD: 1.0983 at $\theta = 0.158$, Max. image execution time: 3.00412 sec at $\theta = 0.248$, Min. image execution time: 0.93418 sec at $\theta = 0.927$.
- For the **Mann** iteration:
  Max. NAI: 1.00000 at $\theta = 0.001$, Min. NAI: 0.07711 at $\theta = 0.9999$, Max. AET: 45.00000 at $\theta = 0.001$, Min. AET: 6.83214 at $\theta = 0.9999$, Max. FD: 1.38922 at $\theta = 0.151$, Min. FD: 1.10793 at $\theta = 0.780$, Max. image execution time: 2.33487 sec at $\theta = 0.387$, Min. image execution time: 0.89034 sec at $\theta = 0.632$.

The Mandelbrot set generated for $F(y) = y^2 + c$ demonstrates the complex dynamics of the iterative methods used for analysis. The MFPI method consistently provided the fastest convergence with the lowest image execution times. This is particularly significant in fractal generation tasks, where computational efficiency is crucial. At smaller θ, MFPI yielded higher fractal dimensions, indicating better resolution and finer details of the fractal boundary. On the other hand, the Mann iteration, while achieving higher fractal dimensions in some regions, required significantly more image execution time and resources. The larger image execution times associated with the Mann iteration make it less suitable for tasks requiring rapid analysis, especially when multiple experiments (200 in this case) are conducted. Picard-Mann and M iterations provided balanced results, with moderate fractal dimensions and image execution times. While the M iteration showed slightly faster convergence than Picard-Mann, the differences were not drastic.

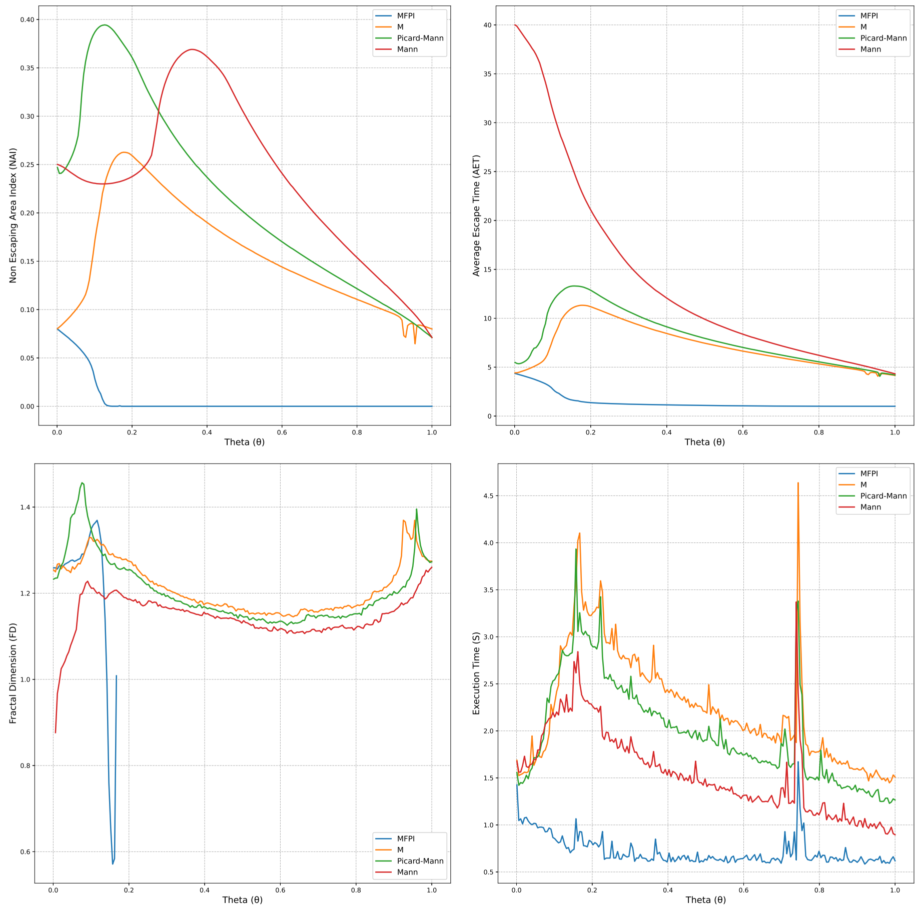

Figure 13. Comparison of NAI, AET, FD, and execution time (seconds) for the proposed iterations in generating the Julia set of $F(y) = y^2 + c$ at $c = (-0.5, 0)$.

Figure 14 shows the Julia set generated for the function $F(y) = y^4 + c$ with $c = (0, 0.5)$ in the area $A = [-2.5, 2.5]^2$, using MFPI, M iteration, Picard-Mann iteration, and Mann iteration methods. The iteration methods were applied with varying relaxation parameters (θ) across 200 experiments to ensure reliable results. The plot reveals the intricate structure of the Julia set and the behavior of each iterative method across different values of θ.

- For the **MFPI** iteration:
  Max. NAI: 0.07649 at $\theta = 0.007$, Min. NAI: 0.00000 at $\theta = 0.502$, Max. AET: 4.87913 at $\theta = 0.003$, Min. AET: 1.20478 at $\theta = 0.976$, Max. FD: 1.9732 at $\theta = 0.004$, Min. FD: 0.7035 at $\theta = 0.376$, Max. image execution time: 1.02457 sec at $\theta = 0.052$, Min. image execution time: 0.49834 sec at $\theta = 0.832$.
- **M** iteration:
  Max. NAI: 0.08942 at $\theta = 0.216$, Min. NAI: 0.08171 at $\theta = 0.824$, Max. AET: 4.72139 at $\theta = 0.230$, Min. AET: 4.30976 at $\theta = 0.804$, Max. FD: 1.2876 at $\theta = 0.135$, Min. FD: 0.6521 at $\theta = 0.417$, Max. image execution time: 2.34281 sec at $\theta = 0.438$, Min. image execution time: 1.00682 sec at $\theta = 0.909$.
- For the **Picard-Mann** iteration:
  Max. NAI: 0.09753 at $\theta = 0.095$, Min. NAI: 0.08274 at $\theta = 0.9999$, Max. AET: 5.83527 at

θ = 0.087, Min. AET: 4.50992 at θ = 0.997, Max. FD: 1.2181 at θ = 0.168, Min. FD: 1.0424 at θ = 0.311, Max. image execution time: 2.97442 sec at θ = 0.222, Min. image execution time: 1.05623 sec at θ = 0.963.

- For the **Mann** iteration:
  Max. NAI: 1.00000 at θ = 0.001, Min. NAI: 0.07732 at θ = 0.9999, Max. AET: 44.81298 at θ = 0.001, Min. AET: 6.70648 at θ = 0.9999, Max. FD: 1.4123 at θ = 0.124, Min. FD: 1.0642 at θ = 0.698, Max. image execution time: 3.20145 sec at θ = 0.502, Min. image execution time: 1.02545 sec at θ = 0.736.

In Figure 14, the results showed the MFPI method once again emerged as the most efficient, yielding low image execution times while maintaining a high fractal dimension. In contrast, the M, Picard-Mann and Mann iteration showed higher fractal dimensions, but their significantly higher image execution times suggested that these iteration methods less suited for large-scale fractal generation tasks.

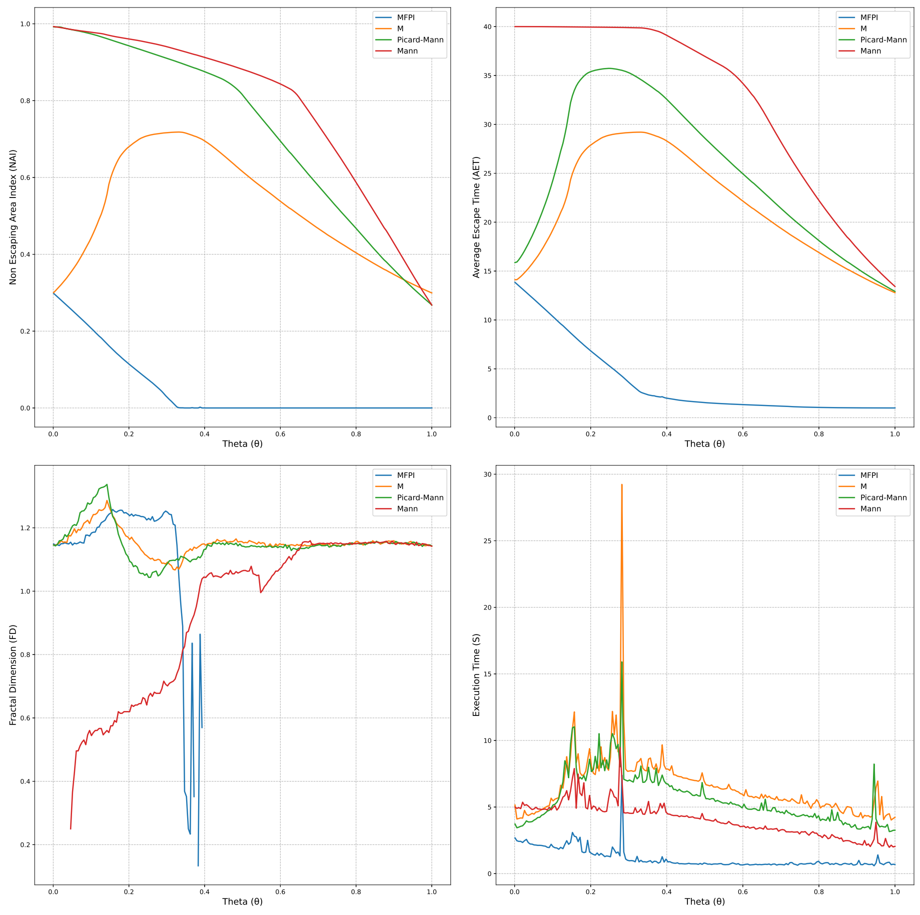

Figure 14. Comparison of NAI, AET, FD, and execution time (seconds) for the proposed iterations in generating the Julia set of $F(y) = y^2 + c$ at $c = (0, 0.5)$.

In the last experiment presented in Figure 15, the Julia set of $F(y) = y^2+c$ with $c = (-0.25, 0.25)$ was generated over the domain $A = [-2.5, 2.5]^2$ using $n = 40$ iterations for the following iterative schemes: MFPI, Mann, Picard-Mann, and M iterations. Several numerical measures were evaluated, including:

- For the **MFPI** iteration: Max. NAI = 0.32092 at θ = 0.001, Min. NAI = 0.00000 at θ = 0.317.

Max. AET = 14.64230 at $\theta = 0.00100$, Min. AET = 1.00000 at $\theta = 0.999$. Max. FD = 1.46134 at $\theta = 0.257$, Min. FD = 0.28491 at $\theta = 0.312$. image execution time varied from 0.46354 sec (at $\theta = 0.804$) to 1.98627 sec (at $\theta = 0.086$).

- For the **M** iteration: Max. NAI = 0.72221 at $\theta = 0.312$, Min. NAI = 0.32169 at $\theta = 0.9999$. Max. AET = 29.37516 at $\theta = 0.31224$, Min. AET = 13.63702 at $\theta = 0.9999$. Max. FD = 1.28175 at $\theta = 0.157$, Min. FD = 1.10328 at $\theta = 0.378$. image execution time ranged from 2.78111 sec (at $\theta = 0.9999$) to 8.62158 sec (at $\theta = 0.357$).
- For the **Picard-Mann** iteration: Max. NAI = 0.99278 at $\theta = 0.001$, Min. NAI = 0.29877 at $\theta = 0.9999$. Max. AET = 34.84148 at $\theta = 0.26204$, Min. AET = 13.76191 at $\theta = 0.9999$. Max. FD = 1.30755 at $\theta = 0.142$, Min. FD = 1.04433 at $\theta = 0.282$. image execution time spanned 2.23263 sec (at $\theta = 0.980$) to 12.80920 sec (at $\theta = 0.352$).
- For the **Mann** iteration: Max. NAI = 0.99162 at $\theta = 0.001$, Min. NAI = 0.29877 at $\theta = 0.9999$. Max. AET = 40.00000 at $\theta = 0.00100$, Min. AET = 14.23780 at $\theta = 0.9999$. Max. FD = 1.16325 at $\theta = 0.794$, Min. FD = 0.40071 at $\theta = 0.036$. image execution time ranged from 1.39561 sec (at $\theta = 0.995$) to 6.36270 sec (at $\theta = 0.352$).

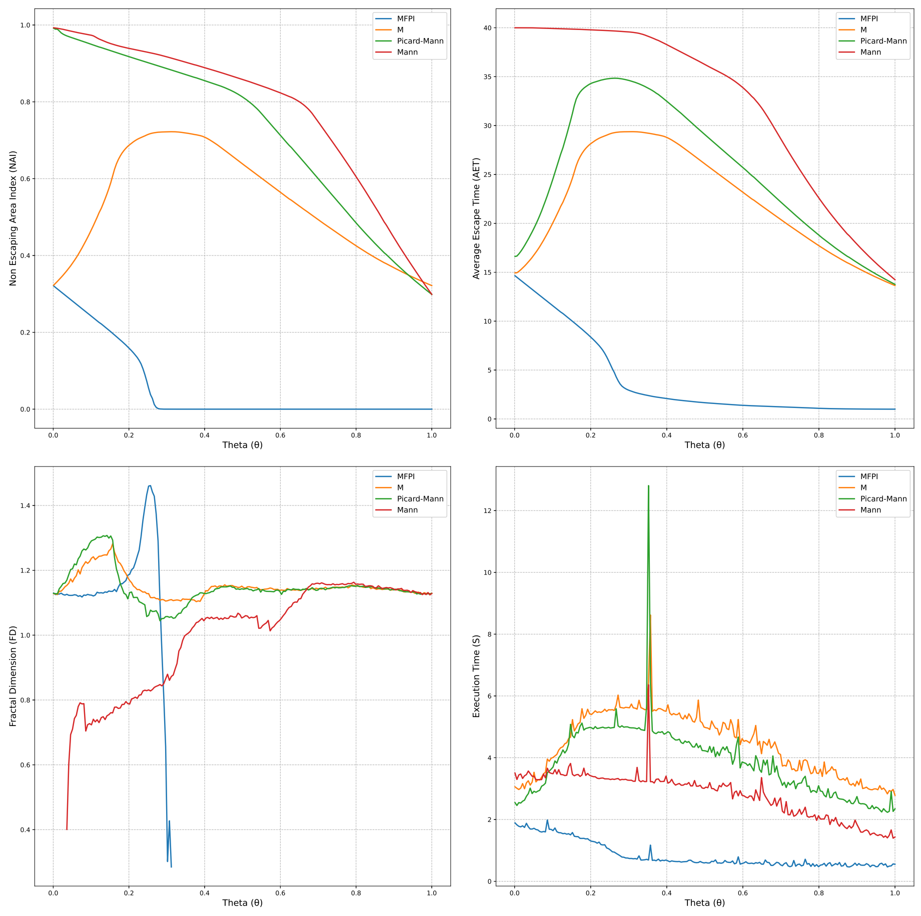

Figure 15. Comparison of NAI, AET, FD, and execution time (seconds) for the proposed iterations in generating the Julia set of $F(y) = y^2 + c$ at $c = (-0.25, 0.25)$.

The MFPI iteration exhibited the most efficient image execution time and the highest fractal detail, as indicated by the FD values. The M iteration showed consistent NAI values but required higher image execution times compared to MFPI. Picard-Mann and Mann iterations exhibited greater complexity in fractal structure, as reflected by higher FD values, but they also had the longest image execution times. When θ was varied, all iterations showed predictable trends in measures such as NAI, AET, and FD, with notable peaks and troughs. These trends highlight the sensitivity of iterative methods to parameter changes. This comparative analysis does not determine a superior iteration method but demonstrates their varied behaviors under changing θ, offering insights for selecting an iteration scheme based on specific objectives (e.g., computational efficiency vs. fractal dimensions).

6. Conclusions

This study demonstrates the effectiveness of a MFPI scheme in generating Mandelbrot and Julia sets of various complex polynomials, providing significant advancements in fractal generation and analysis. Compared to traditional iterative methods such as Mann, Picard-Mann, and M iterations, MFPI consistently exhibits superior performance across fractal generation and multiple evaluation metrics: NAI, AET, FD, and image execution time. Notably, MFPI achieves higher NAI values, indicating a broader set of points that remain within the fractal boundary, as well as greater FD values, reflecting enhanced fractal detail and complexity. Furthermore, MFPI demonstrates lower AET and image execution time values, illustrating its efficiency in generating complex fractal structures with reduced computational demand.

Building on these results, future research could explore further extensions of the MFPI scheme, such as incorporating additional parameters to control fractal density and refinement. Expanding MFPI to three-dimensional fractal sets or applying it in conjunction with machine learning techniques for automated fractal analysis could open new avenues for research. Moreover, Future research should explore the stability of the proposed method under large random noise, ensuring its robustness in practical applications. Additionally, a detailed error analysis is necessary to evaluate its numerical accuracy and convergence properties. These aspects, while beyond the scope of this study, will enhance the theoretical foundation and applicability of the method in future investigations. Additionally, investigating MFPI's applications in fields that utilize fractal models, such as image compression, signal processing, and natural phenomena modeling, could provide practical insights and broaden its scope.

Author contributions

Asifa Tassaddiq: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Resources, Writing-review and editing, Project Administration; Muhammad Tanveer: Conceptualization, Writing-original draft, Validation, Writing-review and editing, Methodology, Formal analysis, Software, Visualisation; Muhammad Arshad: Software, Investigation, Writing-original draft, Validation, Data curation, Formal Analysis; Rabab Alharbi: Conceptualization, Methodology, Supervision, Resources, Data curation; Ruhaila Md Kasmani: Validation, Formal analysis, Investigation, Resources, Writing-review and editing. All authors contributed equally have read and approved the final version of the manuscript for publication.

Use of Generative-AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The author extends the appreciation to the Deanship of Postgraduate Studies and Scientific Research at Majmaah University for funding this research work through the project number (ICR-2025-1508).

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. B. B. Mandelbrot, *The fractal geometry of nature*, San Francisco: W. H. Freeman and Company, 1983.
2. M. Rani, V. Kumar, Superior Julia sets, *Journal of the Korea society of mathematical education series D: Research in mathematical education*, **8** (2004), 261–277.
3. M. Rani, V. Kumar, Superior Mandelbrot set, *Journal of the Korea society of mathematical education series D: Research in mathematical education*, **8** (2004), 279–291.
4. Ashish, M. Rani, R. Chugh, Julia sets and Mandelbrot sets in Noor orbit, *Appl. Math. Comput.*, **228** (2014), 615–631. https://doi.org/10.1016/j.amc.2013.11.077
5. S. Y. Cho, A. A. Shahid, W. Nazeer, S. M. Kang, Fixed point results for fractal generation in Noor orbit and s-convexity, *SpringerPlus*, **5** (2016), 1843. https://doi.org/10.1186/s40064-016-3530-5
6. S. M. Kang, A. Rafiq, A. Latif, A. A. Shahid, Y. C. Kwun, Tricorns and multicorns of S-iteration scheme, *J. Funct. Space.*, **2015** (2015), 1–7. https://doi.org/10.1155/2015/417167
7. M. Kumari, Ashish, R. Chugh, New Julia and Mandelbrot sets for a new faster iterative process, *Int. J. Pure Appl. Math.*, **107** (2016), 161–177. https://doi.org/10.12732/ijpam.v107i1.13
8. C. Zou, A. A. Shahid, A. Tassaddiq, A. Khan, M. Ahmad, Mandelbrot sets and Julia sets in Picard-Mann orbit, *IEEE Access*, **8** (2020), 64411–64421.
9. A. A. Shahid, W. Nazeer, K. Gdawiec, The Picard-Mann iteration with s-convexity in the generation of Mandelbrot and Julia sets, *Monatsh. Math.*, **195** (2021), 565–584. https://doi.org/10.1007/s00605-021-01591-z
10. M. Abbas, H. Iqbal, M. D. la Sen, Generation of Julia and Mandelbrot sets via fixed points, *Symmetry*, **12** (2020), 86. https://doi.org/10.3390/sym12010086
11. A. Tassaddiq, M. Tanveer, M. Azhar, F. Lakhani, W. Nazeer, Z. Afzal, Escape criterion for generating fractals using Picard-Thakur hybrid iteration, *Alex. Eng. J.*, **100** (2024), 331–339. https://doi.org/10.1016/j.aej.2024.03.074

12. M. Tanveer, W. Nazeer, K. Gdawiec, On the Mandelbrot set of $z^p + \log(c^t)$ via the Mann and Picard-Mann iterations, *Math. Comput. Simulat.*, **209** (2023), 184–204. https://doi.org/10.1016/j.matcom.2023.02.012

13. M. Tanveer, K. Gdawiec, Application of CR iteration scheme in the generation of Mandelbrot sets of $z^p + \log(c^t)$ function, *Qual. Theory Dyn. Sys.*, **23** (2024), 1–19. https://doi.org/10.1007/s12346-024-01160-3

14. S. Kumari, K. Gdawiec, A. Nandal, M. Postolache, R. Chugh, A novel approach to generate Mandelbrot sets, Julia sets and biomorphs via viscosity approximation method, *Chaos Soliton Fract.*, **163** (2022), 112540. https://doi.org/10.1016/j.chaos.2022.112540

15. A. Tassaddiq, General escape criteria for the generation of fractals in extended Jungck-Noor orbit, *Math. Comput. Simulat.*, **196** (2022), 1–14. https://doi.org/10.1016/j.matcom.2022.01.003

16. S. Kumari, M. Kumari, R. Chugh, Dynamics of superior fractals via Jungck-SP orbit with s-convexity, *Ann. Univ. Craiova Math. Comput. Sci. Ser.*, **42** (2019), 344–365.

17. X. Y. Li, M. Tanveer, M. Abbas, M. Ahmad, Y. C. Kwun, J. Liu, Fixed point results for fractal generation in extended Jungck-SP orbit, *IEEE Access*, **7** (2019), 160472–160481. https://doi.org/10.1109/ACCESS.2019.2951385

18. Y. C. Kwun, M. Tanveer, W. Nazeer, M. Abbas, and S. M. Kang, Fractal generation in modified Jungck-S orbit, *IEEE Access*, **7** (2019), 35060–35071. https://doi.org/10.1109/ACCESS.2019.2904677

19. H. X. Qi, M. Tanveer, M. S. Saleem, Y. M. Chu, Anti Mandelbrot sets via Jungck-M iteration, *IEEE Access*, **8** (2020), 194663–194675. https://doi.org/10.1109/ACCESS.2020.3033733

20. L. O. Jolaoso, S. H. Khan, Some escape time results for general complex polynomials and biomorphs generation by a new iteration process, *Mathematics*, **8** (2020), 2172. https://doi.org/10.3390/math8122172

21. B. Prasad, B. Katiyar, *Fractals via Ishikawa iteration*, In: Communications in Computer and Information Science, Berlin: Springer, **140**, (2011), 197–203.

22. K. Gdawiec, Inversion fractals and iteration processes in the generation of aesthetic patterns, *Comput. Graph. Forum*, **36** (2015), 35–45. https://doi.org/10.1111/cgf.12783

23. K. Gdawiec, W. Kotarski, A. Lisowska, On the robust Newton's method with the Mann iteration and the artistic patterns from its dynamics, *Nonlinear Dyn.*, **104** (2021), 297–331. https://doi.org/10.1007/s11071-021-06306-5

24. L. O. Jolaoso, S. H. Khan, K. O. Aremu, Dynamics of RK iteration and basic family of iterations for polynomiography, *Mathematics*, **10** (2022), 3324. https://doi.org/10.3390/math10183324

25. S. M. Kang, A. Rafiq, Y. C. Kwun, A new second-order iteration method for solving nonlinear equations, *Abstr. Appl. Anal.*, **2013** (2013), 487062. https://doi.org/10.1155/2013/487062

26. L. Zhou, J. Y. Cai, S. F. Ding, The identification of ice floes and calculation of sea ice concentration based on a deep learning method, *Remote Sens.*, **15** (2023), 2663. https://doi.org/10.3390/rs15102663

27. J. Y. Cai, S. F. Ding, Q. Zhang, R. W. Liu, D. H. Zeng, L. Zhou, Broken ice circumferential crack estimation via image techniques, *Ocean Eng.*, **259** (2022), 111735. https://doi.org/10.1016/j.oceaneng.2022.111735

28. G. Q. Zhou, X. X. Liu, Orthorectification model for extra-length linear array imagery, *IEEE T. Geosci. Remote Sens.*, **60** (2022). https://doi.org/10.1109/TGRS.2022.3223911

29. G. B. Cai, X. Z. Zheng, J. Guo, W. J. Gao, Real-time identification of borehole rescue environment situation in underground disaster areas based on multi-source heterogeneous data fusion, *Safety Sci.*, **181** (2025), 106690. https://doi.org/10.1016/j.ssci.2024.106690

30. H. G. Pan, S. Y. Tong, X. Q. Wei, B. Y. Teng, Fatigue state recognition system for miners based on a multimodal feature extraction and fusion framework, *IEEE T. Cogn. Dev. Syst.*, **17** (2025), 410–420. https://doi.org/10.1109/TCDS.2024.3461713

31. M. Y. Li, T. Jia, H. Wang, B. W. Ma, H. Lu, S. Y. Lin, Ao-DETR, Anti-overlapping DETR for X-ray prohibited items detection, *IEEE T. Neur. Net. Lear. Syst.*, 2024, 1–15. https://doi.org/10.1109/TNNLS.2024.3487833

32. J. Y. Xia, Z. X. Yang, S. X. Li, S. H. Zhang, Y. W. Fu, D. Gndz, Blind super-resolution via meta learning and markov chain monte carlo simulation, *IEEE T. Pattern Anal. Mach. Intell.*, **46** (2024), 8139–8156. https://doi.org/10.1109/TPAMI.2024.3400041

33. H. Chen, Y. C. Bei, W. B. Huang, S. Y. Chen, F. R. Huang, X. Huang, Graph cross-correlated network for recommendation, *IEEE T. Knowl. Data Eng.*, **37** (2025), 710–723. https://doi.org/10.1109/TKDE.2024.3491778

34. H. Jin, S. Y. Tian, J. T. Hu, L. Zhu, S. Zhang, Robust ratio-typed test for location change under strong mixing heavy-tailed time series model, *Commun. Stat-Theory Meth.*, 2025, 1–24. https://doi.org/10.1080/03610926.2024.2446396

35. Y. X. Wu, Y. H. Fan, S. X. Zhou, X. Z. Wang, Q. C. Chen, X. X. Li, Research on the cross-sectional geometric parameters and rigid skeleton length of reinforced concrete arch bridges: A case study of yelanghu bridge, *Structures*, **69** (2024), 107423. https://doi.org/10.1016/j.istruc.2024.107423

36. W. F. Song, X. Wang, S. Zheng, S. Li, A. M. Hao, X. Hou, Talkingstyle: Personalized speech-driven 3d facial animation with style preservation, *IEEE T. Vis. Comput. Gr.*, 2024. https://doi.org/10.1109/TVCG.2024.3409568

37. W. F. Song, X. Wang, Y. M. Jiang, S. Li, A. M. Hao, X. Hou, Expressive 3d facial animation generation based on local-to-global latent diffusion, *IEEE T. Vis. Comput. Gr.*, **30** (2024), 7397–7407. https://doi.org/ 10.1109/TVCG.2024.3456213

38. G. Q. Zhou, H. X. Li, R. H. Song, Q. Y. Wang, J. S. Xu, B. Song, Orthorectification of fisheye image under equidistant projection model, *Remote Sens.*, **14** (2022), 4175. https://doi.org/10.3390/rs14174175

39. G. Q. Zhou, Z. Y. Wang, Q. Li, Spatial negative co-location pattern directional mining algorithm with join-based prevalence, *Remote Sens.*, **14** (2022), 2103. https://doi.org/10.3390/rs14092103

40. M. F. Barnsley, *Fractals everywhere*, New York: Dover Publication, 2014.

41. R. L. Devaney, *A first course in chaotic dynamical systems: Theory and experiment*, Boca Raton: CRC Press, 2020. https://doi.org/10.1201/9780429503481

42. E. Picard, Mémoire sur la théorie des équations aux dérivées partielles et la méthode des approximations successives, *J. Math. Pures Appl.*, **6** (1890), 145–210.

43. W. R. Mann, Mean value methods in iteration, *P. Am. Math. Soc.*, **4** (1953), 506–510. https://doi.org/10.2307/2032162

44. S. H. Khan, A Picard-Mann hybrid iterative process, *Fixed Point Theory Appl.*, **2013** (2013). https://doi.org/10.1186/1687-1812-2013-69

45. K. Ullah, M. Arshad, Numerical reckoning fixed points for Suzuki's generalized nonexpansive mappings via new iteration process, *Filomat*, **32** (2018), 187–196. https://doi.org/10.2298/FIL1801187U

46. S. M. Kang, A. Rafiq, Y. C. Kwun, A new second-order iteration method for solving nonlinear equations, *Abstr. Appl. Anal.*, **2013** (2013), 487062. https://doi.org/10.1155/2013/487062

47. B. Nawaz, K. Ullah, K. Gdawiec, Generation of Mandelbrot and Julia sets by using M-iteration process, *Chaos Soliton. Fract.*, **188** (2024), 115516. https://doi.org/10.1016/j.chaos.2024.115516