DOI: 10.22481/intermaths.v1i1.7704

Artigo | Article

INTERMATHS
ISSN 2675-8318
Vol. 1, N. 1, Jul - Dez 2020, p. 65 – 76

O conjunto de Mandelbrot usando Python

João Otávio Rodrigues Ferreira Frediani

Faculdade de Ciências - Unesp/Bauru

j.frediani@unesp.br

Tatiana Miguel Rodrigues de Souza

Faculdade de Ciências - Unesp/Bauru

tatiana.rodrigues@unesp.br

The Mandelbrot's set using Python

Abstract

Nature in general is constituted by forms in which irregularity and chaos predominate. Trying to simplify them using figures from classical geometry would be inappropriate. Fractal Geometry, in which makes it possible for objects with a fractional dimension to appear, offers a method for analyzing and describe natural objects and shapes, countering the limitations of Euclidean geometry. In this job we will present the Mandelbrot Fractal obtained through the Python language.

Key words: Fractal Geometry; Scientific Computing; Python.

Resumo

A Natureza em geral é constituída por formas nas quais predominam a irregularidade e o caos. Tentar simplificá-las usando figuras da geometria clássica seria inadequado. A Geometria Fractal, na qual se torna possível o surgimento de objetos com dimensão fracionária, oferece um método para analisar e descrever objetos e formas naturais, contrapondo-se às limitações da geometria euclidiana. Neste trabalho será apresentado o Conjunto de Mandelbrot obtido através da linguagem Python.

Palavras-chave: Geometria Fractal; Computação Científica; Python.

Submetido em: 30 de setembro de 2020 – **Aceito em**: 25 de novembro de 2020

1 INTRODUÇÃO

A geometria fractal oferece um método para analisar e descrever objetos e formas naturais, contrapondo-se com as limitações da geometria clássica. A geometria fractal permite a integração de diversos temas da matemática e de outras áreas, desde as ciências naturais às econômico-sociais e à tecnologia. Quando incluída no ensino, permite desenvolver o espírito experimental dos alunos de forma a entender a geometria de objetos não tradicionais e de estabelecer modelos matemáticos para auxiliar os estudos dos fenômenos naturais. Quem primeiramente usou a palavra “fractal” foi Benoit Mandelbrot, baseando-se no latim, do adjetivo fractus, cujo verbo frangere significa quebrar, criar fragmentos.

Esta geometria está intimamente ligada à ciência do Caos. As estruturas dessa geometria, fornecem uma certa ordem ao Caos. Por tal motivo, muitas vezes a geometria fractal é considerada a linguagem do caos.

Os fractais têm grande apelo estético e ambas as áreas se desenvolveram pelo aprimoramento das técnicas computacionais. Aplicações importantes também aparecem na ciência da computação, porque a geometria fractal permite comprimir as imagens; reproduzir, nos ambientes de realidade virtual, os padrões complexos e as formas irregulares presentes na natureza, usando algoritmos iterativos simples executados por computadores.

O objetivo deste trabalho é estudar analiticamente o conjunto de Julia e o conjunto de Mandelbrot e algumas propriedades topológicas usando [2]. Após esta etapa, foi usada a linguagem Python, a qual apresentou mais funções e maior robustez para trabalhar com muitos dados e assim fazer a representação geométrica do conjunto de Mandelbrot estudado.

2 PRELIMINARES

A Geometria Fractal, na qual se torna possível o surgimento de objetos com dimensão fracionária, oferece um método para analisar e descrever objetos e formas naturais, contrapondo-se às limitações da Geometria Euclidiana. Esses “objetos” foram denominados fractais por Benöit Mandelbrot, iniciador dos estudos da Geometria Fractal. As figuras de fractais são geradas a partir de processos iterativos e uma de suas principais características é a autossemelhança, isto é, cada uma de suas partes é semelhante à figura total, segundo [1].

Seja φ: A $\rightarrow$ A uma função definida em um conjunto A qualquer, nele mesmo. Um processo de iteração de uma dada função φ constitui-se em calcular $\varphi(x)$ e, em seguida, aplica-se novamente φ ao resultado para obter $\varphi(\varphi(x)) = \varphi^2(x)$ e assim sucessivamente. Repete-se o processo até $\varphi^k(x)$, isto é, a k-ésima, aplicação de φ em x. O

processo iterativo dessa função em um ponto inicial x_0 gera um sistema dinâmico.

A sequência dos resultados das sucessivas iterações de um ponto é chamado de órbita desse ponto.

Na função $f(x) = x^2$, a órbita de $x_0 = 2$ é dada por:

$$2, 4, 16, 256, 65536, \ldots \tag{1}$$

Pode-se ter órbitas finitas, com um número finito de elementos, ou órbitas infinitas com um número infinito de elementos.

Exemplo 2.1. *Seja $f : \mathbb{C} \longrightarrow \mathbb{C}$ tal que $f(z) = z^2 + c$, onde $z \in \mathbb{C}$ e $c = a + bi$ é uma constante com $a, b \in \mathbb{R}$ e $i = \sqrt{-1}$.*

Tomando $z_0 = 0$ como ponto inicial, observa-se o que ocorre com as órbitas para alguns valores de c nessa função:

(a) Para $c = -1$, obtemos: $0, -1, 0, -1, 0, -1, \ldots$

(b) Para $c = i$, obtemos: $0, i, i-1, -i, i-1, -i, \ldots$

Uma órbita é chamada periódica se existem inteiros m e n, com $m \geq n$, tais que: $f^{n+m}(x_0) = f^n(x_0)$.

Agora, para $c = 0$, para essa função $f(z) = z^2 + c$, definida em $\mathbb{C}$, também é válido que:

- Se $z_0 = 1$, então é gerada uma órbita invariante.
- Se $|z_0| < 1$, então é gerada uma órbita regressiva.
- Se $|z_0| > 1$, então é gerada uma órbita progressiva.

De fato, sabendo que um número complexo pode ser representado na forma trigonométrica por $z = \rho(cos(\theta) + isen(\theta))$, onde $\rho = |z_0|$, tem-se:

$z = \rho(cos(\theta)+isen(\theta)) \Leftrightarrow z^n = [\rho(cos(\theta) + isen(\theta)]^n \Leftrightarrow z^n = \rho^n.(cos(\theta)+isen(\theta))^n \Leftrightarrow |z^n| = |\rho^n.(cos(\theta)+isen(\theta))^n| \Leftrightarrow |z^n| = |\rho^n|.|(cos(\theta)+isen(\theta))^n| \Leftrightarrow |z^n| = |\rho^n|.1 \Leftrightarrow |z^n| = |\rho^n|$

Assim, para a função $|f(z_{n-1})| = |z_0|^{2^n}$, temos:

- Se $z_0 = 1 \Rightarrow \lim_{n\to+\infty} |z_0|^{2^n} = 1$, órbita invariante.

- Se $z_0 < 1 \Rightarrow \lim_{n\to+\infty} |z_0|^{2^n} = 0$, órbita regressiva.
- Se $z_0 > 1 \Rightarrow \lim_{n\to+\infty} |z_0|^{2^n} = \infty$, órbita progressiva.

Quando $|f^n(z_0)|$ cresce infinitamente, ou seja, quando tem-se uma órbita progressiva, o infinito é chamado atrator. Todos os pontos do plano complexo que possuem o infinito como atrator constituem a bacia de atração do infinito, representada por B_∞.

Quando tem-se uma órbita regressiva, a origem é o atrator. Todos os pontos de $\mathbb{C}$ que possuem a origem como atrator constituem a bacia de atração B_0.

Dessa forma o plano complexo fica dividido em dois conjuntos: o conjunto dos pontos com órbitas infinitamente crescentes e outro com órbitas limitadas. Também restam os pontos com órbitas invariantes $(f(z) = z)$ ou periódicas $(f^n(z) = z)$.

Para determinar os pontos fixos deve-se resolver a equação $f(z) = z$. Agora, para determinar os pontos de período $2, 3$ e assim por diante tem-se que resolver, respectivamente, as equações $f^2(z) = z$, $f^3(z) = z$ até $f^n(z) = z$. Porém, nem sempre é fácil encontrar a solução dessas, sendo necessário utilizar métodos numéricos e computacionais.

Esses pontos fixos ou periódicos constituem um conjunto dinamicamente invariante, o qual é importante para a geração de fractais. Já os pontos do plano complexo que não tem atrator são os pontos de fronteira das bacias B_0 e B_∞. Estes são importantes para formar o conjunto de Julia e de Mandelbrot.

2.1 Sobre a linguagem *Python*

A linguagem *Python* foi criada em 1991 inspirada na linguagem “ABC”. Desde sua criação seu uso vem crescendo exponencialmente sendo hoje uma das mais populares para diferentes funcionalidades como análise de dados, inteligência artificial e redes neurais, e, a utilizada neste trabalho, visualização de dados.

É uma linguagem de alto nível interpretada, ou seja, utiliza um interpretador para processá-la linha por linha, então não há tempo de compilação antes da execução, diferente de linguagens como C++ e *Java*. *Python* também suporta orientação a objeto, permitindo a criação de classes de objeto, garantindo herança de métodos e atributos.

A linguagem também chama atenção de indivíduos não acostumados a programação, a sua obrigatoriedade, a indentação, torna o código de fácil leitura, exigindo apenas um pequeno conhecimento sobre os conceitos de programação. A sua filosofia de desenvolvimento comunitário sem fins lucrativos garante que qualquer um possa não só usá-la, como também incrementá-la. Dessa forma, com as possibilidades de se fazer cálculos vetorizados e com auxílio da propagação automática de dados por diversas dimensões *Python* possui vantagem sobre ferramentas especializadas em cálculo

científico como *Octave* e *Matlab*.

Além disso, a linguagem também concentra diversas bibliotecas e *frameworks* extremamente ricos que incluem diversos métodos que podem ser aplicados de maneira simples e flexível. Neste trabalho foram utilizadas as bibliotecas *Numpy*, biblioteca auxiliar para cálculos extensos que inclui vetores multidimensionais, operações estatísticas clássicas, rotinas de operação com matrizes e diversos outros objetos e métodos, e a biblioteca *Matplotlib*, que gera gráficos de alta qualidade com pleno controle de manipulação do programador.

3 RESULTADOS PRINCIPAIS

Os resultados abaixo serão usados para a construção usando Python do conjunto de Mandelbrot ([2], [3] e [4]).

Teorema 3.1 (do ponto fixo atrator)**.** *Sejam $A \subset \mathbb{C}$ e w um ponto fixo atrator da função $f : A \longrightarrow A$. Então existe uma bola aberta de centro em w e raio δ na qual a seguinte condição é satisfeita: se $z \in B_\delta(w) \cap A$, então $f^k(z) \in B_\delta(w) \cap A$ e $f^k(z) \longrightarrow w$ quando $k \longrightarrow \infty$.*

Demonstração: De fato, como $|f^{'}(w)| < 1$, tomando $r = \dfrac{1 + |f^{'}(w)|}{2}$. Desta forma $|f^{'}(w)| < r < 1$. Por definição, $f^{'}(w) = \lim\limits_{z \longrightarrow w} \dfrac{f(z) - f(w)}{z - w}$. Dessa forma,

$$\forall \epsilon > 0, \exists \delta > 0 \text{ tal que } 0 < |z - w| < \delta \Longrightarrow \left| \frac{f(z) - f(w)}{z - w} - f^{'}(w) \right| < \epsilon.$$

Pela desigualdade triangular,

$$\left| \frac{f(z) - f(w)}{z - w} \right| - |f^{'}(w)| \leq \left| \frac{f(z) - f(w)}{z - w} - f^{'}(w) \right| < \epsilon \text{ logo } \left| \frac{f(z) - f(w)}{z - w} \right| < \epsilon + |f^{'}(w)|.$$

Tomando $\epsilon = r - |f^{'}(w)| > 0$. Com isto,

$$\left| \frac{f(z) - f(w)}{z - w} \right| < r \text{ para todo } z \in A - \{w\} \text{ com } |z - w| < \delta.$$

Consequentemente, para $z \in A$ com $|z - w| < \delta$, temos

$$|f(z) - w| \leq r|z - w| < r\delta, \text{ onde } 0 < r < 1. \tag{2}$$

Isto significa que $f(z)$ está mais próxima de w do que z. Portanto, $f(z) \in B_\delta(w) \cap A$.

Observe que $f^2(z) = f(f(z))$, com $f(z) \in B_\delta(w) \cap A$. Aplicando a desigualdade (2)

duas vezes obtemos:

$$|f^2(z) - w| = |f(f(z)) - w| \leq r|f(z) - w| \leq r.r|z - w| < r^2\delta, \text{ onde } 0 < r < 1.$$

Logo, $f^2(z) \in B_\delta(w) \cap A$. Por indução em k, para $z \in A$ com $|z - w| < \delta$, temos $|f^k(z) - w| < r^k.\delta$, onde $0 < r < 1$. Portanto, $f^k(z) \in B_\delta(w) \cap A$. Quando $k \longrightarrow \infty$, $r^k \longrightarrow 0$ e, consequentemente, $f^k(z) \longrightarrow w$. ■

Teorema 3.2 (do ponto fixo repulsor)**.** *Sejam $A \subset \mathbb{C}$ e w um ponto fixo repulsor da função $f : A \longrightarrow A$. Então, existe uma bola aberta de centro em w e raio δ na qual a seguinte condição é satisfeita: se $z \in B_\delta(w) \cap A$ e $z \neq w$, então existe $n_0 \in \mathbb{N}^\star$ tal que $f^n(z) \notin B_\delta(w) \cap A$ para todo $n \geq n_0$.*

A demonstração é análoga a feita para o teorema anterior.

Proposição 3.1. *$A_f(w)$ é um conjunto aberto.*

De fato, como w é um ponto fixo atrator de f, pelo Teorema 3.1 existe um conjunto aberto V contendo w tal que $V \subset A_f(w)$ (se $w = \infty$, pode-se tomar $\{z : |z| > r\}$ para r suficientemente grande). Isto implica que $A_f(w)$ é aberto. Com efeito, se $z \in A_f(w)$, então $f^k(z) \in V$ para algum k, logo $z \in f^{-k}(V)$, o qual é aberto. ■

Ainda considerando o exemplo da função $f(z) = z^2$ observe que os pontos $z \in \mathbb{C}$, tais que $|z| \leq 1$ possuem órbitas limitadas, também são denominados prisioneiros e os pontos tais que $|z| > 1$ possuem órbitas ilimitadas, ou seja, órbitas que escapam para o infinito. Desta forma, tem-se a definição:

Definição 3.1. *Seja f uma função polinomial complexa de grau $n \geq 2$. O conjunto prisioneiro de o conjunto de escape para a função f são definidos, respectivamente, por:*

$$K(f) = \{z \in \mathbb{C} : |f^k(z)| \longrightarrow \infty \text{ quando } k \longrightarrow \infty\} \tag{3}$$

e

$$E(f) = \{z \in \mathbb{C} : |f^k(z)| \nrightarrow \infty \text{ quando } k \longrightarrow \infty\}. \tag{4}$$

Observe que $\mathbb{C}$ é a união disjunta entre $K(f)$ e $E(f)$. A partir daí tem-se uma dicotomia: o plano complexo é dividido em dois subconjuntos cuja intersecção é vazia.

Definição 3.2. *Seja f uma função polinomial complexa de grau $n \geq 2$. O conjunto de Julia preenchido da função f é definido por*

$$K(f) = \{z \in \mathbb{C} : |f^k(z)| \nrightarrow \infty \text{ quando } k \longrightarrow \infty\}. \tag{5}$$

Observação 3.1. *Como $J(f) = \partial A_f(\infty) = \partial E(f) = \partial K(f)$, segue que o conjunto de Julia de f é a fronteira do conjunto de Julia preenchido.*

Afirmação 3.1. *O conjunto de Julia e o conjunto de Julia preenchido são fechados.*

Demonstração: O conjunto de Julia é fechado por ser fronteira de um conjunto. Pela Proposição 3.1, $A_f(\infty)$ é aberto. Como $A_f(\infty) = E(f)$ e $K(f)$ são complementares, segue que o conjunto de Julia preenchido é fechado.

3.1 O Conjunto de Mandelbrot

Seja $\mathcal{F}$ a família das funções quadráticas da forma $f_c : \mathbb{C} \longrightarrow \mathbb{C}$ dada por $f_c(z) = z^2 + c$, onde c é uma constante complexa.

É possível provar que toda função quadrática é topologicamente conjugada a algum membro da família quadrática $\mathcal{F}$. Isto significa que ao serem estudados os conjuntos de Julia de f_c, com $c \in \mathbb{C}$, são estudados os conjuntos de Julia de todas as funções quadráticas.

Observa-se a partir das definições que foram apresentadas até o momento e das que se seguirão que ao fazer o estudo do conjunto de Julia é estudar o conjunto de Mandelbrot.

Definição 3.3. *O conjunto de Mandelbrot $\mathcal{M}$ é o conjunto dos parâmetros c tais que o conjunto de Julia $J(f_c)$ é conexo, isto é,*

$$\mathcal{M} = \{c \in \mathbb{C} : J(f_c) \text{ é conexo}\}. \tag{6}$$

Definição 3.4. *O conjunto de Mandelbrot é o conjunto dos parâmetros c tais que a órbita do ponto crítico de f_c é limitada*

$$\mathcal{M} = \{c \in \mathbb{C} : \{f_c^k(0)\}_{k\geq 1} \text{ é limitado}\}. \tag{7}$$

Observação 3.2. *A órbita do ponto crítico ser limitada é equivalente à órbita do ponto crítico não tender ao infinito. Desta forma, tem-se a igualdade entre os seguintes conjuntos:*

$$\mathcal{M} = \{c \in \mathbb{C} : \{f_c^k(z)\}_{k\geq 1} \text{ é limitado }\} = \{c \in \mathbb{C} : f_c^k(0) \nrightarrow \infty \text{ quando } k \longrightarrow \infty\}. \tag{8}$$

Lema 3.1. *Sejam $z \in \mathbb{C}$ e $f_c(z) = z^2 + c$, onde c é uma constante complexa. Se $|z| \geq c$ e $|z| > 2$, então existe número real positivo ϵ tal que $|f_c^n(z)| \geq (1+\epsilon)^n|z|$.*

Demonstração: De fato, fazendo por indução sobre n, de $|z| > 2$, segue que existe número real positivo ϵ com $|z| = 2 + \epsilon$. Consequentemente, $|z| - 1 = 1 + \epsilon$.

- Para $n = 0,\ |f_c^0(z)| = |z| \geq (1+\epsilon)^0|z|$.

- Para $n = 1,\ |f_c(z)| = |z^2 + c|$.

Da desigualdade triangular, tem-se $|z^2| = |z^2 + c - c| \leq |z^2 + c| + |-c| = |z^2 + c| + |c|$. Assim, $|f_c(z)| = |z^2 + c| \geq |z^2| - |c| = |z|^2 - |c| \geq |z|^2 - |z| = (|z| - 1)|z| = (1 + \epsilon)|z|$.

- Para $n = 2,\ |f_c^2(z)| = |f_c(f_c(z))|$.

De acordo com o caso anterior, $|f_c(z)| \geq (1 + c)|z| > |z| = 2 + \epsilon$. Como $|z| \geq |c|$ e $|z| > 2$, segue que $|f_c(z)| \geq |c|$ e $|f_c(z)| > 2$. Assim sendo, pode-se usar o caso anterior para o complexo $f_c(z)$. Desse modo,

$$|f_c^2(z)| = |f_c(f_c(z))| \geq (1 + \epsilon)f_c(z)| \geq (1 + \epsilon).(1 + \epsilon)|z| = (1 + \epsilon)^2|z|. \tag{9}$$

Suponha que a afirmação é verdadeira para algum $n = k$. Ou seja, $|f_c^k(z)| \geq (1 + \epsilon)^k|z|$. Desde que $|f_c^k(z)| \geq c$ e $|f_c^k(z)| > 2$, pode-se escrever

$$|f_c^{k+1}(z)| = |f_c(f_c^k(z))| \geq (1 + \epsilon)f_c^k(z)| \geq^{H.I.} (1 + \epsilon).(1 + \epsilon)^k|z| = (1 + \epsilon)^{k+1}|z|. \tag{10}$$

Portanto, pelo Princípio da Indução Finita, a afirmação é verdadeira para todo $n \in \mathbb{N}$. ■

Proposição 3.2. *Sejam $z \in \mathbb{C}$ e $f_c(z) = z^2 + c$, onde c é uma constante complexa. Se $|z| \geq |c|$ e $|z| > 2$, então $\lim_{n \longrightarrow \infty} |f_c^n(z)| = \infty$ (ou seja, a órbita de z tende ao infinito).*

Demonstração: De fato, de acordo com o Lema 3.1, existe um número real positivo ϵ tal que $|f_c(z)s| \geq (1 + \epsilon)^n|z|$. Como $\lim_{n \longrightarrow \infty} |f_c^n(z)| = \infty$, tem-se

$$\lim_{n \longrightarrow \infty} |f_c^n(z)| \geq \lim_{n \longrightarrow \infty} (1 + \epsilon)^n|z| = |z| \lim_{n \longrightarrow \infty} (1 + \epsilon)^n = \infty. \tag{11}$$

Portanto, a órbita de z tende ao infinito se $|z| \geq c$ e $|z| > 2$. ■

Corolário 3.1. *O conjunto de Julia de f_c está contido no círculo de centro 0 e raio $r_c = max\{|c|, 2\}$.*

Demonstração: De fato, de acordo com a definição, $J(f_c) = \partial A_f(\infty)$, onde $A_f(\infty)$ é aberto. Logo, os pontos do plano complexo que são iterados para o infinito não pertencem a $J(f_c)$. Desta forma, por Proposição 3.2, o conjunto de Julia de f_c está contido no círculo de centro 0 e raio $r_c = max\{|c|, 2\}$. ■

Corolário 3.2. *O conjunto de Julia preenchido de f_c, está contido no círculo de centro 0 e raio $r_c = max\{|c|, 2\}$.*

Demonstração: De fato, pela Proposição 3.2, os pontos fora do círculo de centro 0 e raio r_c são iterados para o infinito, logo não pertencem a K_c. ■

Afirmação 3.2. *O conjunto de Julia preenchido de f_c é compacto.*

De fato, pela proposição anterior, K_c é limitado e pela Afirmação 3.1 é fechado.

Corolário 3.3. *Sejam $z \in \mathbb{C}$, $f_c(z) = z^2 + c$, onde c é uma constante complexa e $r_c = max\{|c|, 2\}$. Se para algum $n \in \mathbb{N}$ for satisfeita a expressão $|f_c^n(z)| > r_c$, então a órbita de z tende para o infinito.*

Demonstração: De fato, seja $n_0 \in \mathbb{N}$ tal que $|f_c^{n_0}(z)| > r_c$.

Como $r_c = max\{|c|, 2\}$, segue que $r_c \geq |c|$ e $r_c \geq 2$. Portanto, $|f_c^{n_0}(z)| > |c|$ e $|f_c^{n_0}(z)| > 2$. Logo, $w = f_c^{n_0}(z)$ satisfaz as hipóteses da Proposição 3.2. Assim,

$$\lim_{n\longrightarrow\infty} |f_c^n(w)| = \lim_{n\longrightarrow\infty} |f_c^n(f_c^{n_0}(z))| = \lim_{n\longrightarrow\infty} |f_c^{n+n_0}(w)| = \infty. \tag{12}$$

Portanto, a órbita de w tende para o infinito. O mesmo acontece com a órbita de z, uma vez que a órbita de z contém a órbita de w. ■

Corolário 3.4. *O conjunto de Mandelbrot pertence ao círculo de raio 2.*

Demonstração: De fato, de acordo com a Observação 3.2,

$$\mathcal{M} = \{c \in \mathbb{C} : f_c^k(0) \nrightarrow \infty \text{ quando } k \longrightarrow \infty\}. \tag{13}$$

Tomando $c \in \mathbb{C}$ tal que $|c| > 2$ e $z = 0$. Observe que $r_c = max\{|c|, 2\} = |c|$.

Como $f_c(0) = 0^2 + c = c$, tem-se $|f_c(0)| = |c| = r_c$. Mas, $f_c^2(0) = f_c(f_c(0)) = f(c) = c^2 + c$, e neste caso,

$$|f_c^2(0)| = |c^2 + c| \geq |c^2| - |c| = |c|^2 - |c| = |c|(|c| - 1) \overset{|c|>2}{>} |c| = r_c.$$

De acordo com o Corolário 3.3, a órbita de $z = 0$ tende ao infinito.

A partir desta definição foi desenvolvido o programa na linguagem Python $3.8.3$ com o auxílio das bibliotecas *N*umpy versão 1.19.0 e *M*atplotlib versão 3.3.1 para a construção do conjunto de Mandelbrot.

3.2 Análise do Código em *Python*

Para demonstrar como a imagem do conjunto de Mandelbrot foi gerada, abaixo segue o código em *Python* que realiza os cálculos e projeta o gráfico:

```
1  import numpy as np
2  import matplotlib.pyplot as plt
3  from numpy import newaxis
4  x = np.linspace(-2, 1, 3000)
5  y = np.linspace(-1, 1, 2000)
6  c = x[:,newaxis] + 1j*y[newaxis,:]
7  z = c
8  for i in range(100):
9  z = (z*z) + c
10 conj = (abs(z) < 2)
11 plt.imshow(conj.T, extent=[-2, 1, -1, 1])
12 plt.show()
```

Nas linhas $1, 2$ e 3 são feitas as importações de bibliotecas e funções que serão utilizadas.

Sabendo que um número complexo possui formato $z = x + yi$ nas linhas 4 e 5 são gerados vetores para os valores que x e y devem assumir. A função *linspace* da biblioteca *Numpy* gera um vetor de pontos igualmente distanciados para um período específico, neste caso para x foram escolhidos três mil pontos entre -2 e 1, e para y foram gerados dois mil pontos entre -1 e 1.

Para então unir os vetores x e y em um número complexo utilizou-se a função *newaxis* da biblioteca *Numpy* em cada vetor, dessa maneira aumentou-se em 1 o número de eixos: x então torna-se uma matriz dois por dois, onde o primeiro eixo está preenchido com os valores gerados pela função *linspace* e o segundo eixo está vazio, o mesmo acontece em y, porém com os eixos invertidos, o primeiro vazio e o segundo com os valores previamente obtidos. Multiplica-se então a matriz y pelo valor j que para *Python* funciona como o número complexo i, e soma-se as matrizes em $\mathbb{C}$.

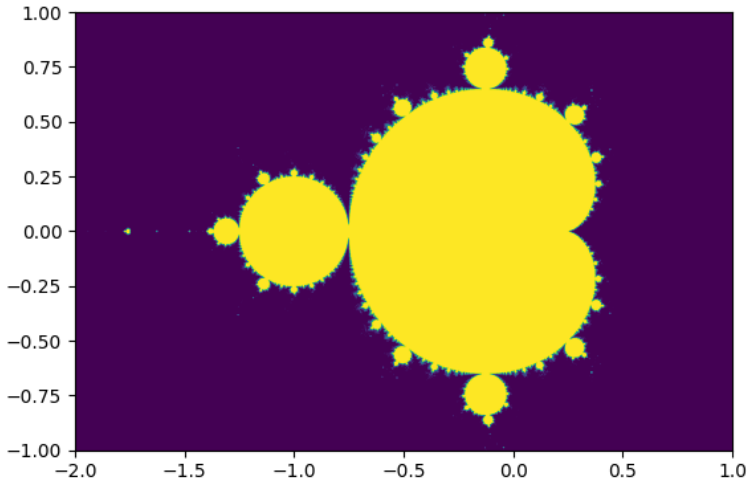
Na sequência, a linguagem automaticamente propaga os valores por todas as dimensões já que estão alternadamente vazias em x e y, obtem-se então todas as combinações possíveis entre os valores de x e y, com y multiplicado por i.

Nas linhas 8 e 9 tem-se um laço que repetirá a execução da função $f(z) = z^2 + c$ cem vezes e atribuirá o valor em z. Como o objeto utilizado é uma matriz o símbolo $\star$ não fará uma multiplicação de matrizes, mas sim uma multiplicação ponto a ponto, assim o cálculo vetorizado diminui o tempo de execução e obtém os valores procurados.

Então na linha 10 atribuí-se à variável $conj$ o valor 0 aos números complexos que após cem iterações tem seu módulo maior ou igual a 2, e o valor 1 para os que possuem módulo menor que 2. Na linha 11 determina-se o tamanhos dos eixos x e y do gráfico a ser exibido, e na linha 11 executa-se a função que exibe o gráfico.

A partir do exposto acima foi obtido

Figura 1: Fractal de Mandelbrot.



Fonte: Imagem elaborada pelo próprio autor

4 CONCLUSÃO

Devido a complexidade de algumas figuras fractais é inviável criá-las manualmente. Portanto, sem o auxílio de um *software* ou da linguagem *Python* não seria possível criar imagens do Conjunto de Mandelbrot estudado e assim fazer uma análise mais profunda de conteúdos matemáticos associados a esse objeto. Como trabalho futuro pretende-se fazer uma análise da complexidade do algoritmo utilizado na linguagem *Python* e na linguagem *Scilab* para a criação do mesmo conjunto de Mandelbrot.

REFERÊNCIAS

[1] R. Barbosa, Descobrindo a Geometria Fractal, 3^a ed., Autêntica, Belo Horizonte, 2007.

[2] M.F. Barnsley, Fractals Everywhere, second ed, Academic Press Professional, New York, 1993.

[3] A. Lins Neto, Funções de uma Variável Complexa, 4^a Edição, IMPA (Projeto Euclides), Rio de Janeiro, 1993.

[4] R. A. Uceda, Propriedades Topológicas de conjuntos de Julia. Dissertação de mestrado, Universidade Estadual Paulista, Júlio Mesquista Filho, 81f, 2008.

[5] F. Nelly, Python Data Analytics, second ed, Apress, New York, 2018.

[6] W. McKenney, Python for Data Analysis, first ed, O'Reilly Media, United States of America, 2012.

[7] Chudoba, Rostislav & Sadílek, Václav & Rypl, Rostislav & Vorechovsky, Miroslav. Using Python for scientific computing: Efficient and flexible evaluation of the statistical characteristics of functions with multivariate random inputs. Computer Physics Communications. 184. 414–427, 2013. [CrossRef]

BREVE BIOGRAFIA

João Otávio Rodrigues Ferreira Frediani https://orcid.org/0000-0002-6544-9066

Estudante do curso Bacharelado em Ciência da Computação na Universidade Estadual Paulista Júlio de Mesquita Filho. Tem interesse nas áreas de geometria computacional, inteligência artificial e deep learning.

Tatiana Miguel Rodrigues de Souza https://orcid.org/0000-0001-8203-915X

Possui graduação em Licenciatura em Matemática pela Universidade Estadual Paulista Júlio de Mesquita Filho (2000) e mestrado em Matemática pela Universidade Estadual Paulista Júlio de Mesquita Filho (2003),fez doutorado em sistemas dinâmicos simbólicos, área de seu interesse. Atualmente trabalha como professora doutora no departamento de Matemática da UNESP, campus Bauru.