

# Pesquisa de Órbitas Indiferentes

Bruno Ferreira Mariani  
Gabriel Alves de Lima  
Luiz Alberto Thomé Speltz Filho  
Matheus Moraes Santos

A motivação principal deste capítulo é estudar pontos de um sistema dinâmico a partir da classificação de suas órbitas, em especial, estamos interessados em delinear comportamentos para pontos cujas órbitas são neutras (ou indiferentes), uma vez que tais órbitas não assumem uma convergência em todos os casos. Além disso, buscamos obter um método para encontrar pontos periódicos de um período  $n$  dado. Salientamos que o referencial teórico é totalmente baseado em [2]. De outra parte, a posterior aplicação computacional e as imagens usadas para exemplificação dos conteúdos foram elaboradas, com auxílio da referência [1], em *Python* (versão 3.8.0), uma linguagem de programação criada por Guido van Rossum, e plotadas utilizando o software livre *PyCharm*.

# 1 Órbitas e autovalores

## 1.1 Órbitas

Inicialmente, discutiremos conceitos elementares, necessários para a formalização teórica do tema.

Consideremos o sistema dinâmico gerado por uma função  $f : \mathbb{C} \rightarrow \mathbb{C}$  qualquer (aqui não é exigido condições sobre  $f$ ). A órbita progressiva, ou somente órbita, de um ponto  $z_0 \in \mathbb{C}$  por  $f$  é o conjunto de pontos obtidos a partir das iterações consecutivas de  $z_0$ , isto é:

$$O(z_0) = \{z_0, f(z_0), f^2(z_0), \dots, f^k(z_0), \dots\}$$

Ainda, se denotarmos  $z_1 = f(z_0), z_2 = f(z_1), \dots, z_k = f(z_{k-1}), \dots$ , podemos reescrever a órbita de  $z_0$  em termos da sequência  $(z_n)_{n \in \mathbb{N}}$ , como

$$O(z_0) = \{z_0, z_1, z_2, \dots, z_k, \dots\}$$

Um exemplo clássico para ilustrar esta definição é tomar a função

$$\begin{aligned} f : \mathbb{C} &\longrightarrow \mathbb{C} \\ z &\longmapsto z^2 \end{aligned}$$

Vamos testar alguns valores de  $z$  e descrever suas respectivas órbitas por  $f$ .

- Para  $z_0 = 1$ , temos

$$\begin{aligned} z_0 &= 1 \\ z_1 &= f(z_0) = f(1) = 1^2 = 1 \\ z_2 &= f(z_1) = f(1) = 1^2 = 1 \\ &\dots \\ z_k &= f(z_{k-1}) = f(1) = 1^2 = 1 \\ &\dots \end{aligned}$$

Logo, a órbita de 1 por  $f$  é composta somente pelo 1.

- Para  $z_0 = 2$ , temos

$$\begin{aligned} z_0 &= 2 \\ z_1 &= f(z_0) = f(2) = 2^2 = 4 \\ z_2 &= f(z_1) = f(4) = 4^2 = 16 \\ &\dots \\ z_k &= f(z_{k-1}) = f(2^{2^{k-1}}) = 2^{2^k} \\ &\dots \end{aligned}$$

Assim, temos que  $O(2) = \{2, 4, 16, \dots, 2^{2^k}, \dots\}$ .

É fácil ver que as duas órbitas exemplificadas acima possuem comportamentos bastante distintos. É isso que abordaremos a seguir.

Podemos classificar as órbitas de pontos do sistema dinâmico gerado por uma função  $f$  em dois casos disjuntos:

i. A órbita de um ponto  $z_0$  é dita *confinada*, se existe  $\delta \in \mathbb{R}$ , com  $\delta > 0$  tal que

$$\lim_{k \rightarrow \infty} |z_k| = \lim_{k \rightarrow \infty} |f^k(z_0)| < \delta$$

isto é,  $f^k(z_0) \in B_{0,\delta}$  (Bola de centro 0 e raio  $\delta$ ),  $\forall k \in \mathbb{N}$ .

ii. A órbita de um ponto  $z_0$  é dita *fugitiva*, se, para todo  $\epsilon \in \mathbb{R}$  com  $\epsilon > 0$ , existe um  $k \in \mathbb{N}$  tal que

$$|f^k(z_0)| > \epsilon$$

Note que, nos exemplos anteriores,  $O(1)$  se comporta como uma órbita confinada, uma vez que tomando  $\delta = 2$  temos  $1 = f^k(1) \in B_{0,\delta}$ ,  $\forall k \in \mathbb{N}$ , e  $O(2)$  tem caráter fugitivo, pois  $\lim_{k \rightarrow \infty} |f^k(2)| = \lim_{k \rightarrow \infty} 2^{2^k} = \infty$ , ou seja, para todo  $\epsilon > 0$ , existe  $k \in \mathbb{N}$  tal que  $|f^k(2)| > \epsilon$ .

## 1.2 Pontos periódicos

Um ponto  $z \in \mathbb{C}$  é dito ponto fixo do sistema dinâmico gerado pela função  $f$  se

$$f(z) = z$$

Além disso, podemos encontrar pontos fixos resolvendo a seguinte equação,

$$f(z) - z = 0 \tag{1}$$

Este conceito pode ser estendido para a  $n$ -ésima iteração de  $f$ , isto é, um ponto  $z \in \mathbb{C}$  é dito ponto periódico de período  $n$ , se após  $n$  iterações de  $f$  voltamos a  $z$ , ou seja,

$$f^n(z) = z$$

Analogamente, para encontrar pontos periódicos de período  $n$ , resolvemos a equação:

$$f^n(z) - z = 0 \tag{2}$$

Encontrar soluções analíticas de (1) e (2) é, muitas vezes, inviável. Desta forma, recorre-se a métodos numéricos para obter soluções que, dentro de um limite computacional, são boas aproximações. Discutiremos mais sobre estes métodos na subseção 1.4.

Quando nos limitamos à função  $f(z) = z^2 + c$ , onde  $c \in \mathbb{C}$ , podemos descrever seus pontos fixos de forma analítica, do seguinte modo:

$$\begin{aligned} f(z) - z = 0 &\Rightarrow (z^2 + c) - z = 0 \\ &\Rightarrow z^2 - z + c = 0 \\ &\Rightarrow z = \frac{1 \pm \sqrt{1 - 4c}}{2} \end{aligned}$$

Logo, os pontos fixos de  $f(z) = z^2 + c$  são:  $z_1 = \frac{1 + \sqrt{1 - 4c}}{2}$  e  $z_2 = \frac{1 - \sqrt{1 - 4c}}{2}$ .

Se restringirmos ainda mais a função  $f$ , fixando  $c = 0$ , podemos facilmente estudar seus pontos periódicos de período  $n$ , fazendo:

$$\begin{aligned} f^n(z) - z = 0 &\Rightarrow z^{2^n} - z = 0 \\ &\Rightarrow z(z^{2^{n-1}} - 1) = 0 \\ &\Rightarrow z = 0 \text{ ou } z = \xi_{2^{n-1}} \end{aligned}$$

onde  $\xi_{2^{n-1}}$  é uma raiz da unidade<sup>1</sup> de ordem  $2^{n-1}$ .

Note que todo ponto fixo é ponto periódico de período  $n$ , mais ainda, se  $z_0$  é ponto periódico de período  $p$ , então também é ponto periódico de período  $p \cdot q$ ,  $\forall q \in \mathbb{N}$ .

### 1.3 Classificação das órbitas periódicas

Seja  $z_0$  um ponto periódico de período  $n$  de uma função  $f$  de classe  $C^1$ . Definimos o seu autovalor como:

$$\begin{cases} \lambda_{z_0} = (f^n)'(z_0), & \text{se } |z_0| < \infty; \text{ ou} \\ \lambda_{z_0} = \frac{1}{(f^n)'(z_0)}, & \text{se } |z_0| = \infty \end{cases}$$

Uma observação relevante é que, como todo ponto na órbita de  $z_0$  tem período  $n$ , o autovalor  $\lambda_{z_0}$  é, na verdade, autovalor de todos os pontos da órbita de  $z_0$ .

De fato,

$$\begin{aligned} z_1 &= f(z_0), \text{ então } z'_1 = f'(z_0) \\ z_2 &= f(z_1) = f^2(z_0), \text{ então } z'_2 = (f^2)'(z_0) = f'(z_1) \cdot z'_1 = f'(z_1) \cdot f'(z_0) \\ &\dots \\ z_n &= f(z_{n-1}) = f^n(z_0), \text{ então} \\ z'_n &= (f^n)'(z_0) = f'(z_{n-1}) \cdot z'_{n-1} = f'(z_{n-1}) \cdot f'(z_{n-2}) \cdot \dots \cdot f'(z_0) \end{aligned}$$

Ou seja,  $\lambda_{z_0} = (f^n)'(z_0) = f'(z_{n-1}) \cdot f'(z_{n-2}) \cdot \dots \cdot f'(z_0)$ .

Calculemos agora o autovalor  $\lambda_{z_k}$ , para um ponto  $z_k$  na órbita de  $z_0$  (isto é,  $1 < k < n$ ). Por simplicidade, faremos para  $k = 1$ , isto é, estamos interessados em  $z_1 = f(z_0)$ .

Por um raciocínio análogo, teremos que

$$\lambda_{z_1} = (f^n)'(z_1) = f'(z_n) \cdot f'(z_{n-1}) \cdot \dots \cdot f'(z_1)$$

Agora, do fato que as órbitas destes pontos tem período  $n$ , sabemos que  $z_n = z_0$ , e, portanto, segue que  $\lambda_{z_0} = \lambda_{z_1}$ . Da mesma forma, argumentamos que os outros pontos da órbita possuem o mesmo autovalor, assim diremos que a **órbita periódica** de  $z_0$  tem autovalor  $\lambda$ .

Da observação anterior, podemos associar os pontos  $z \in \mathbb{C}$  de uma órbita  $O(z)$  a um mesmo autovalor  $\lambda$ , logo, podemos realizar a seguinte classificação acerca de  $O(z)$ :

$$\begin{aligned} & \textit{superatratora}, \text{ se } \lambda = 0 \\ & \textit{atratora}, \text{ se } 0 < |\lambda| < 1 \\ & \textit{indiferente}, \text{ se } |\lambda| = 1 \\ & \textit{repulsora}, \text{ se } |\lambda| > 1 \end{aligned}$$

<sup>1</sup>Uma raiz  $n$ -ésima da unidade é um número complexo  $\xi$  satisfazendo  $\xi^n = 1$ .

## 1.4 Resolução numérica de $f^n(z) - z = 0$

Como comentado anteriormente, obter soluções analíticas para a equação  $f^n(z) - z = 0$  é, muitas vezes, inviável. Desta forma, é preciso usar métodos numéricos para o cálculo das soluções.

Uma das formas mais clássicas na abordagem de problemas deste gênero é o *método de Newton*, o qual se caracteriza por um sistema iterativo de aproximações. De forma sucinta, a construção deste método para a solução do problema em questão acontece da seguinte forma:

1. Define-se uma nova função,  $g(z) = f^n(z) - z$ ;
2. Calcula-se sua derivada,  $g'(z) = (f^n)'(z) - 1$ ;
3. A partir de um ponto inicial  $z_0$ , define-se a relação de recorrência:

$$z_{k+1} = z_k - \frac{g(z_k)}{g'(z_k)}, \quad (k = 0, 1, 2, \dots)$$

4. As iterações terminam em qualquer um dos subcasos abaixo:
  - (a) Encontramos um ponto  $z_{k+1}$  tal que  $|g(z_{k+1})| < \epsilon$ , para algum  $\epsilon > 0$  suficientemente pequeno fixado previamente;
  - (b) Chegamos no número máximo de iterações,  $k = k_{max}$ .
5. Se ocorre (a), dizemos que  $z_{k+1}$  é uma solução procurada, e se ocorre (b) dizemos que o número de iterações máximo estipulado foi insuficiente para encontrar uma solução.

Em (3.3), abordamos mais detalhadamente a computação por trás deste processo, explicando um dos possíveis algoritmos e dando alguns exemplos.

## 2 Pesquisa de Órbitas indiferentes

Falaremos agora sobre uma estratégia para a pesquisa de órbitas indiferentes, que são aquelas cujo autovalor  $\lambda$  satisfaz  $|\lambda| = 1$ . Antes, vale dizer, a importância do estudo dessas órbitas está no fato de que elas não apresentam tendência nem de atração nem de repulsão, comportando-se de diferentes maneiras e, em alguns casos, traçando figuras incomuns.

Porém, nem toda função  $f$  admite uma órbita indiferente de período  $n$ . Assim, para que consigamos trabalhar com funções adequadas ao nosso estudo, escolheremos aquelas cujo parâmetro possa ser determinado no sentido de possibilitar órbitas indiferentes.

Num primeiro momento, é conveniente usar a função  $f(z) = z^2 + c$ .

Mas como determinaremos  $c$  a ponto de orientar  $f$  a nos mostrar o que buscamos?

Uma solução é conjugar  $f$ , expressa em termos do parâmetro  $c$ , com uma função  $g$ , expressa em termos do autovalor  $\lambda$ . Esse processo de conjugação pode ser aplicado à função  $f$  através da transformação de Möbius, conforme o diagrama abaixo:

$$\begin{array}{ccc} \mathbb{C} & \xrightarrow{f} & \mathbb{C} \\ M \downarrow & & \downarrow M \\ \mathbb{C} & \xrightarrow{g} & \mathbb{C} \end{array}$$

em que  $M = \frac{\alpha z + \beta}{\gamma z + \delta}$ , com  $\alpha, \beta, \gamma, \delta \in \mathbb{C}$  e  $\alpha\delta - \beta\gamma \neq 0$ .

Note que o diagrama indica  $M \circ f(z) = g \circ M(z)$ .

Dizemos, nesse caso, que  $f$  e  $g$  são analiticamente conjugadas, isto é, são dinamicamente equivalentes, pois seus sistemas dinâmicos comportam-se de modo topologicamente equivalentes, particularmente no que concerne às órbitas e às bacias de atração.

Mas quem poderia ser  $g$ , para que consigamos pesquisar pontos fixos de autovalor  $\lambda$  em conformidade com o que fora dito acima?

Uma possibilidade é  $g(z) = \lambda z + \lambda z^2$ .

Nesse caso,  $g$  tem a origem como ponto fixo, pois  $g(0) = 0$ , ficando preservado o autovalor do ponto fixo da função  $f$ , já que  $g'(0) = \lambda$ .

Além disso, para a conjugação, podemos usar a transformação  $M$  da forma:  $M(z) = \frac{1}{\lambda} - \frac{1}{2}$ .

Assim, de  $M \circ f(z) = g \circ M(z)$ , isto é, de

$$\frac{1}{\lambda}(z^2 + c) - \frac{1}{2} = \lambda\left(\frac{1}{\lambda} - \frac{1}{2}\right) + \lambda\left(\frac{1}{\lambda}z - \frac{1}{2}\right)^2$$

obtemos  $c = \frac{2\lambda - \lambda^2}{4}$ .

Com esse parâmetro, fica fácil aplicar em  $f(z) = z^2 + c$  valores de  $\lambda$  que possibilitem a observação de órbitas periódicas indiferentes, como nos propusemos no início.

É preciso destacar, no entanto, que a maneira como uma órbita indiferente influencia a órbita de um ponto do conjunto de confinamento  $K$  depende primordialmente do autovalor  $\lambda$ , de modo que podemos classificar as órbitas periódicas com autovalor  $\lambda$  de módulo unitário de duas formas:

- i. *Racionalmente indiferente*, se  $\lambda$  for uma raiz da unidade e o seu argumento for expresso como um múltiplo racional de  $2\pi$ , isto é,  $\lambda = e^{s2\pi i}$ , com  $s \in \mathbb{Q}$ .
- ii. *Irracionalmente indiferente*, se o argumento de  $\lambda$  for múltiplo irracional de  $2\pi$ , ou seja,  $\lambda = e^{s2\pi i}$ , com  $s \in \mathbb{R} \setminus \mathbb{Q}$ .

## 2.1 Órbitas periódicas racionalmente indiferentes

Discutiremos agora com mais detalhes algumas particularidades das órbitas periódicas racionalmente indiferentes, isto é, órbitas cujo autovalor  $\lambda$  é da forma  $e^{s2\pi i}$ , com  $s \in \mathbb{Q}$ . A primeira propriedade interessante é que toda órbita periódica indiferente está no conjunto de Julia, o que computacionalmente exige um número elevado de iterações para ser observado.

O conjunto de Julia pode assumir diversas formas, dependendo da função escolhida, aqui iremos nos concentrar em conjuntos formados por *lóbulos*: figuras simples, topologicamente equivalentes à circunferência. A imagem abaixo mostra o conjunto de Julia da função  $g(z) = \lambda z + \lambda z^2$ , com  $\lambda = e^{\frac{2\pi i}{3}}$ :

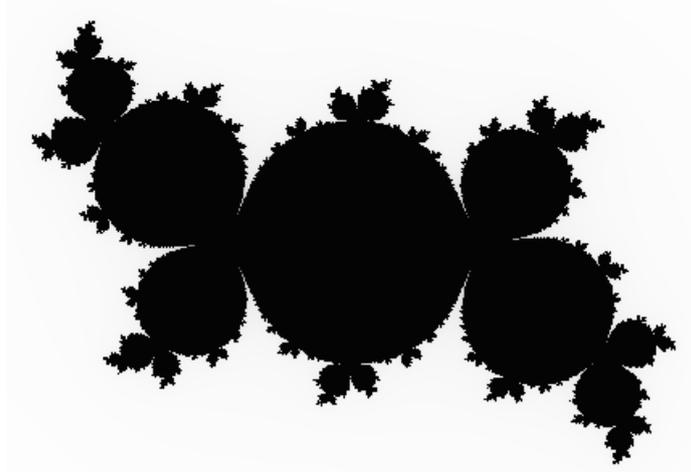


Figura 1: Conjunto de Julia da função  $g$  com  $\lambda = e^{\frac{2\pi i}{3}}$

Podemos observar que o conjunto de Julia se constitui de pontos de junção de lóbulos (neste caso de 3 em 3) - lembrando que se trata apenas da fronteira da imagem (a parte interna constitui o conjunto de confinamento). Notemos que, como dito anteriormente, o processo de formação da imagem exige muitas iterações, o que dificulta observar a junção dos lóbulos (obtemos, porém, uma boa aproximação).

As figuras (2) e (3) constituem outros exemplos de Conjuntos de Julia com valores de  $\lambda$  múltiplos racionais de  $2\pi$ :

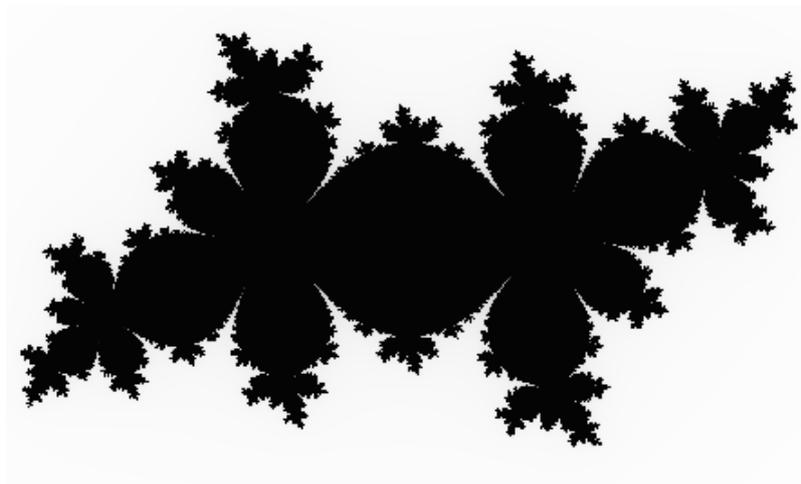


Figura 2: Conjunto de Julia da função  $g$  com  $\lambda = e^{\frac{4\pi i}{5}}$

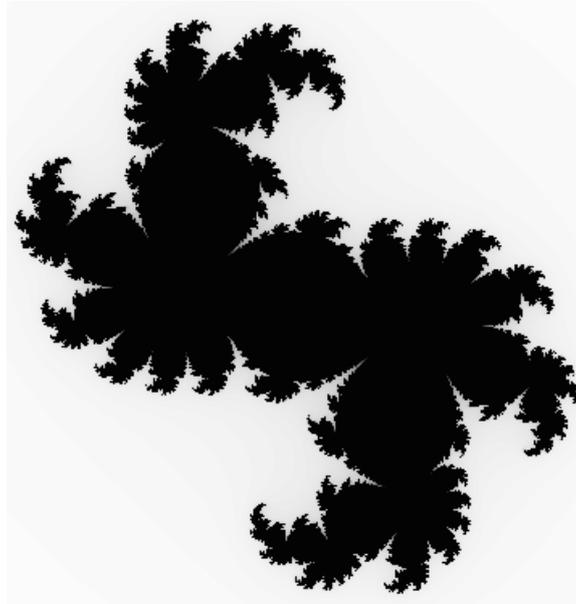


Figura 3: Conjunto de Julia da função  $g$  com  $\lambda = e^{\frac{2\pi i}{7}}$

Vemos que, se  $\lambda$  for uma raiz primitiva de ordem  $q$  da unidade, então os pontos da órbita racionalmente indiferente serão uma junção de  $kq$  lóbulos, para algum  $k \in \mathbb{N}$ . Na figura (1),  $\lambda$  é uma raiz primitiva de ordem 3 da unidade, e neste caso  $k = 1$ .

Temos também que, dado um ponto no interior do conjunto  $K$  (conjunto de confinamento), a órbita deste ponto será formada por uma figura de  $q$  pétalas em torno dos pontos  $n$ -periódicos.

Vale notar que, para gerar essas pétalas, é necessário um número extremamente elevado de iterações. Iremos agora mostrar imagens de algumas aproximações (aqui os pontos tiveram sua espessura aumentada para ajudar na visualização).

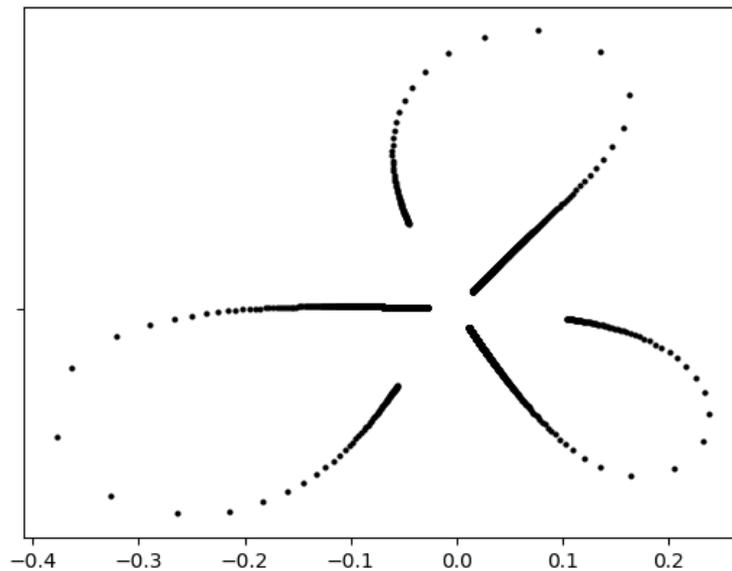


Figura 4: Órbita em formato de pétala ao redor da origem no conjunto de Julia de  $g$  com  $\lambda = e^{\frac{2\pi i}{3}}$

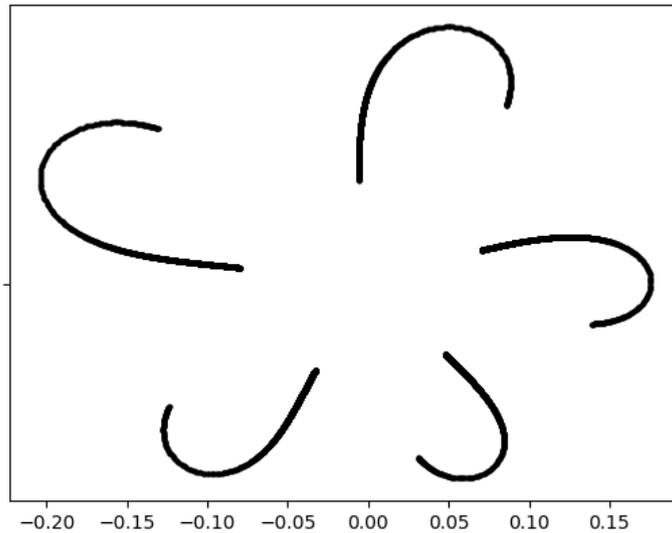


Figura 5: Órbita em formato de pétala ao redor da origem no conjunto de Julia de  $g$  com  $\lambda = e^{\frac{4\pi i}{5}}$

Se tivéssemos uma capacidade computacional bem elevada, veríamos as pétalas das figuras (4) e (5) “completas” e dentro dos respectivos conjuntos de Julia.

Por último, notamos que, para um ponto dentro de um lóbulo, sua órbita será formada por  $q$  pétalas, portanto em cada lóbulo existe um infinidade de pétalas.

## 2.2 Órbitas periódicas irracionalmente indiferentes

Agora, discutiremos órbitas periódicas com autovalor da forma  $\lambda = e^{s2\pi i}$ , onde  $s \in \mathbb{R} \setminus \mathbb{Q}$ . A primeira grande diferença é que não podemos afirmar que tais órbitas estarão no conjunto de Julia da função trabalhada, de outro modo, a chance de estar em seu complementar (conjunto de Fatou) é maior.

Um ponto interessante é que, se a órbita periódica estudada estiver no conjunto de Fatou, então os pontos que se encontram no interior de lóbulos do conjunto de confinamento irão formar uma curva simples ao redor dos pontos periódicos. O conjunto dessas curvas simples é denominado *disco de Siegel* e preenche todo o interior do lóbulo. A figura (6) é um exemplo de conjunto de Julia da função dada por  $g(z) = \lambda z + \lambda z^2$ , onde  $\lambda = e^{1,6i}$ ; e a figura (7) representa órbitas de pontos no interior de um dos lóbulos da figura (6).

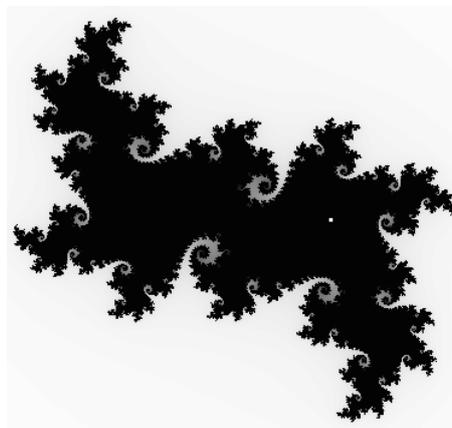


Figura 6: Conjunto de Julia de  $g$ , com  $\lambda = e^{1,6i}$

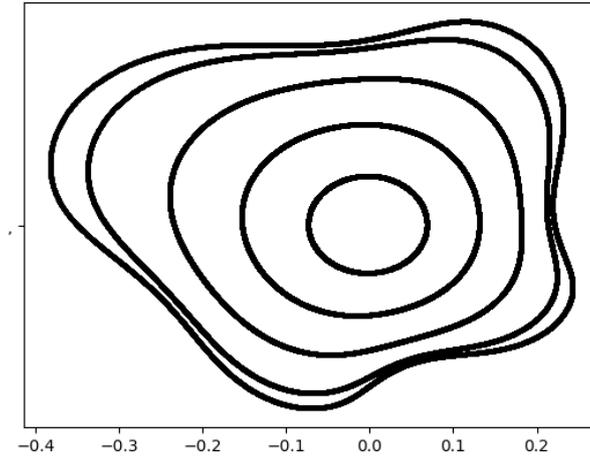


Figura 7: Curvas simples ao redor da origem em um dos lóbulos da figura anterior

### 3 Programando

A parte da programação foi dividida em blocos, todos com base na função  $g(z) = \lambda z + \lambda z^2$ : dois deles abordaram os valores de  $\lambda$  como raízes da unidade, ou seja,  $\lambda = e^{s2\pi i}$ , com  $s \in \mathbb{Q}$ ; outros dois abordaram os valores de  $\lambda$  da forma  $\lambda = e^{si}$ , sendo  $s \in \mathbb{Q}$  um decimal racional definido pelo usuário (note que de fato  $s$  gera um múltiplo irracional de  $2\pi$ , pois  $s = \left(\frac{s}{2\pi}\right) \cdot 2\pi$ , e para todo  $s \in \mathbb{Q}$  o número  $\frac{s}{2\pi}$  é irracional); por fim, um dos blocos de código calculou os pontos periódicos, por meio do método de Newton.

#### 3.1 Programação das figuras aproximadas

Como forma de melhor visualização dos conjuntos e compreensão da parte teórica, resolvemos programar um código que gerasse figuras aproximadas de conjuntos de Julia para a função  $g(z) = \lambda z + \lambda z^2$ .

Para isto, criamos uma matriz com 640 linhas e 640 colunas. Cada valor da matriz representa um ponto no plano complexo, igualmente espaçados, com a parte real e complexa variando de -2 até 2. Cada um desses pontos tem inicialmente designado o valor 255 e, então, cada ponto é testado em iterações consecutivas da função  $g(z) = \lambda z + \lambda z^2$ . Para cada iteração, o valor inicial diminui em um certo valor, que pode variar de acordo com a precisão desejada (os valores usados foram 5 ou 1), até que  $g^n(z)$  atinja um certo limite, em módulo (o valor usado foi 10).

Feito isto, organizamos os valores de modo que representassem seus pontos equivalentes no plano complexo e salvamos a imagem em um formato que converteia os valores numéricos para tons de cinza. No caso, os valores próximos de 255 seriam lidos como próximos de branco e os valores próximos de 0 seriam lidos como pretos. Assim, os pontos da função que vão rapidamente para o infinito são brancos ou quase brancos, e os valores da função que permanecem confinados por muitas iterações são mais escuros.

Os códigos descritos acima para o parâmetro  $\lambda$ , cujo argumento é um múltiplo racional ou irracional de  $2\pi$ , são, respectivamente,

#### Código dos conjuntos de Julia a partir de autovalores racionalmente indiferentes

```
import math

def Julia_raizes_unidade():
    gp = int(input('Coloque o grau do polinomio '))
    pot = int(input('Qual potencia da raiz da unidade? '))
    a = complex(math.cos(pot * 2 * math.pi / gp),
                math.sin(pot * 2 * math.pi / gp))
    print(a)
    vetorReal = numpy.arange(-2, 2, 4 / 640)
    vetorImaginario = numpy.arange(-2, 2, 4 / 640)
    nomeUsuario = input('Coloque o nome do arquivo ' +
                        ' ')
    nomeArquivo = str(nomeUsuario + '.pgm')
    arquivo = open(nomeArquivo, 'w')
    arquivo.write('P2\n#\nJulia_Set_image\n' +
                  '640' + '\n' + '640' + '\n255\n')
    for i in vetorImaginario:
```

```

    for r in vetorReal:
        z = complex(r, i)
        n = 255
        while abs(z) < 5 and n >= 5:
            z = a * z * z + a * z
            n -= 2
        arquivo.write(str(n) + '_')
    arquivo.write('\n')
    arquivo.close()
Julia_raizes_unidade()

```

### Código dos conjuntos de Julia a partir de autovalores irracionalmente indiferentes

```

import numpy
import math

def Julia_raizes_irracionais():

    ang = float(input('Qual o angulo, em radianos,
do ponto do circulo unitario?'))
    lamb = complex(math.cos(ang), math.sin(ang))

    vetorReal = numpy.arange(-2, 2, 4 / 640)
    vetorImaginario = numpy.arange(-2, 2, 4 / 640)

    nomeUsuario = input('Coloque o nome do arquivo')
    nomeArquivo = str(nomeUsuario + '.pgm')

    arquivo = open(nomeArquivo, 'w')
    arquivo.write('P2\n#\nJulia_Set_image\n' +
        '640' + '_' + '640' + '\n255\n')

    for i in vetorImaginario:
        for r in vetorReal:
            z = complex(r, i)
            n = 255
            if abs(z) < 0.01:
                n = 255
            else:
                while abs(z) < 5 and n >= 5:
                    z = lamb * z * z + lamb * z
                    n -= 2
                arquivo.write(str(n) + '_')
        arquivo.write('\n')
    arquivo.close()

Julia_raizes_irracionais()

```

## 3.2 Programação das órbitas

Para enxergar as órbitas de pontos afetados pelas órbitas indiferentes, primeiramente, o usuário deve escolher o valor de  $\lambda$  desejado, e então variar o ponto inicial  $z_0$ . Pela própria escolha da função  $g(z) = \lambda z + \lambda z^2$ , esse trabalho se torna mais simples, já que a origem é um ponto fixo indiferente. Assim, basta escolher alguns pontos próximos da origem e fazer diversas iterações, salvando o valor de cada uma delas em um vetor. É importante notar que, quando se diz “próximos” da origem, foram utilizados pontos  $z_0$  tais que  $|z_0| < 0,3$ .

É preciso, nesse caso, tomar o cuidado de não escolher pontos muito próximos da origem, caso contrário seriam necessárias uma quantidade de iterações extremamente altas, com tempos de compilação muito demorados, para se formar algum tipo de imagem significativa. Por exemplo, ao testar  $z_0 = 0,02 + 0,02i$  na função com  $\lambda = e^{\frac{2\pi i}{3}}$ , obtemos a seguinte figura:

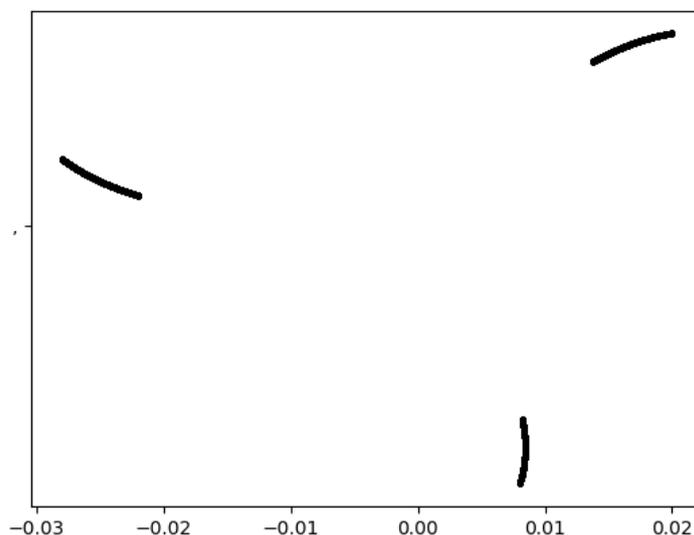


Figura 8: Início da formação das pétalas na função com  $\lambda = e^{\frac{2\pi i}{3}}$  e  $|z_0| = 0,02 + 0,02i$

### Código dos pontos afetados pelas órbitas racionalmente indiferentes

```
import math

def Julia_raizes_unidade():
    gp = int(input('Coloque o grau do polinomio '))
    pot = int(input('Qual potencia da raiz da unidade? '))
    a = complex(math.cos(pot * 2 * math.pi / gp),
                math.sin(pot * 2 * math.pi / gp))
    print(a)
    vetorReal = numpy.arange(-2, 2, 4 / 640)
    vetorImaginario = numpy.arange(-2, 2, 4 / 640)
    nomeUsuario = input('Coloque o nome do arquivo ')
    nomeArquivo = str(nomeUsuario + '.pgm')
    arquivo = open(nomeArquivo, 'w')
```

```

arquivo.write('P2\n#_Julia_Set_image\n' +
              '640' + '_' + '640' + '\n255\n')
for i in vetorImaginario:
    for r in vetorReal:
        z = complex(r, i)
        n = 255
        while abs(z) < 5 and n >= 5:
            z = a * z * z + a * z
            n -= 2
        arquivo.write(str(n) + '_')
        arquivo.write('\n')
arquivo.close()
Julia_raizes_unidade()

```

### Código dos pontos afetados pelas órbitas irracionalmente indiferentes

```

import numpy
import math

def Julia_raizes_irracionais():
    ang = float(input('Qual o angulo, em radianos
    .....do ponto do circulo unitario?_'))
    lamb = complex(math.cos(ang), math.sin(ang))
    vetorReal = numpy.arange(-2, 2, 4 / 640)
    vetorImaginario = numpy.arange(-2, 2, 4 / 640)
    nomeUsuario = input('Coloque o nome do arquivo_')
    nomeArquivo = str(nomeUsuario + '.pgm')
    arquivo = open(nomeArquivo, 'w')
    arquivo.write('P2\n#_Julia_Set_image\n' +
                 '640' + '_' + '640' + '\n255\n')
    for i in vetorImaginario:
        for r in vetorReal:
            z = complex(r, i)
            n = 255
            if abs(z) < 0.01:
                n = 255
            else:
                while abs(z) < 5 and n >= 5:
                    z = lamb * z * z + lamb * z
                    n -= 2
                arquivo.write(str(n) + '_')
                arquivo.write('\n')
    arquivo.close()
Julia_raizes_irracionais()

```

### 3.3 Método de Newton

Para encontrar órbitas de período  $n$  na função  $g(z) = \lambda z + \lambda z^2$ , utilizamos o método de Newton para resolver a função  $g(z) = f^n(z) - z = 0$ . Para calcular esses

valores, aplicamos um método iterativo. Primeiramente, cada iteração do método pode ser calculada como:

$$z_{k+1} = z_k - \frac{g(z_k)}{g'(z_k)}$$

Usamos então as igualdades:

$$\begin{aligned} g(z_k) &= f^n(z_k) - z_k \\ g'(z_k) &= (f^n)'(z_k) - 1 \end{aligned}$$

Podemos obter o valor de  $(f^n)'(z_k)$  a partir das iterações:

$$\begin{aligned} z_{k_1} &= f(z_k), \dots, z_{k_{l+1}} = f(z_{k_l}) \\ z'_{k_1} &= f'(z_k), \dots, z'_{k_n} = (f^n)'(z_k) = f'(z_{k_{n-1}})z'_{k_{n-1}} \end{aligned}$$

Então, ao escolher o valor de  $n$ , podemos procurar órbitas que satisfazem o período desejado. É importante notar que o método de Newton utiliza um ponto inicial (nesse caso,  $z_k$ ) e obtém apenas um resultado para cada ponto. Para encontrar vários pontos de uma órbita, foram testados diversos pontos do plano complexo, com os valores reais e imaginários variando de -2 até 2, igualmente espaçados.

Serão mostrados na figura abaixo os 8 pontos com órbita de período  $n = 3$  da função  $g(z) = \lambda z + \lambda z^2$ , com  $\lambda = e^{\frac{2\pi i}{7}}$

Ponto	Parte real	Parte imaginária
1	-1.5282354110048657	-0.5206205591816541
2	-0.5027317001372349	1.0867692091730783
3	-0.8876091340621606	-1.1225495882132217
4	-1.3973669148333951	0.16440496161983853
5	0.5600409052971207	0.2290245396497188
6	0.1324124528818016	0.9448029194202676
7	-0.37651019814126657	0.7818314824680296
8	0	0

### Código do método de Newton

```
import math
import numpy

gp = int(input('Coloque o grau do polinomio '))
pot = int(input('Qual potencia da raiz da unidade? '))
grau = int(input('Qual o grau de periodicidade do ponto? '))
lamb = complex(math.cos(pot * 2 * math.pi / gp),
               math.sin(pot * 2 * math.pi / gp))
vetorReal = numpy.arange(-2, 2, 4 / 100)
vetorImaginario = numpy.arange(-2, 2, 4 / 100)
zdif = complex(1, 0)
for i in vetorImaginario:
    for r in vetorReal:
        z = complex(r, i)
        z0 = z
        k = 0
```

```

while k < grau:
    z = z * lamb + z * z * lamb
    zdif = (2 * lamb * z + lamb) * zdif
    k += 1
z1 = z - z0
z2 = zdif - complex(1 , 0)
m = 0
k = 0
while m < 2000:
    z0 = z0 - z1/z2
    zk = z0
    zdif = complex(1, 0)
    while k < grau:
        zk = zk * lamb + zk * zk * lamb
        zdif = (2 * lamb * zk + lamb) * zdif
        k += 1
    z1 = zk - z0
    z2 = zdif - complex(1, 0)
    k = 0
    m += 1
if(abs(z0) < 3 and abs(z0) > 0.5):
    print(z0)

```

## Referências

- [1] Synthetic Programming. How to program: A julia set fractal generator in python. <https://youtu.be/fYqBuXc12VM>, 2017. Acesso em: 15 nov. 2019.
- [2] C.P. Serra and E.W. Karas. Fractais gerados por sistemas dinâmicos complexos. Champagnat, Curitiba, Brasil, 1997.