Revista da Graduação

Vol. 4 No. 1 2011 24

Seção: FACULDADE DE INFORMÁTICA

Título: Um estudo do método de Newton fractal para resolução de equações

Autor: Artur Luiz Silva da Cunha Freitas

Este trabalho está publicado na Revista da Graduação.
ISSN 1983-1374
http://revistaseletronicas.pucrs.br/ojs/index.php/graduacao/article/view/8807/6171

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

ARTUR LUIZ SILVA DA CUNHA FREITAS

UM ESTUDO DO MÉTODO DE NEWTON FRACTAL PARA RESOLUÇÃO DE EQUAÇÕES

Porto Alegre
2010

ARTUR LUIZ SILVA DA CUNHA FREITAS

UM ESTUDO DO MÉTODO DE NEWTON FRACTAL PARA RESOLUÇÃO DE EQUAÇÕES

Trabalho de conclusão de curso de graduação apresentado à Faculdade de Informática da Pontifícia Universidade Católica do Rio Grande do Sul, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientadora: Profª. Beatriz Regina Tavares Franciosi

Porto Alegre

2010

Dedico este trabalho a minha
curiosidade que me motiva nesta jornada
rumo ao desconhecido.

AGRADECIMENTOS

Agradeço a minha família e amigos pelo apoio e incentivo. Agradeço mais ainda a eles por serem do jeito que são, pois caso contrário eu não seria do jeito que sou. Agradeço também aos meus professores e a minha orientadora. Agradeço a todos que, de alguma forma, aumentaram o conhecimento da humanidade.

A ciência serve para nos dar uma ideia de quão extensa é a nossa ignorância.

Félicité Robert de Lamennais

RESUMO

Este trabalho investiga a possibilidade de utilizar a exploração visual na identificação da estabilidade de métodos numéricos. Para conduzir a investigação foi escolhido o tema resolução de equações e o método numérico de Newton para cálculo de raízes de equações. De fato, foi utilizado o método de Newton fractal para gerar imagens cujas formas gráficas identificam regiões de atratores ou bacias de convergência. Para entender o significado destas imagens, no contexto de estabilidade de algoritmos numéricos, são utilizadas estratégias mais elaboradas de pensamento do que simplesmente “olhar”; são estratégias que possibilitam “ver”. Informações transportadas por formatos, cores e sombras são objetos de estudo deste trabalho. Este texto é constituído pela fundamentação teórica de métodos numéricos, teoria dos fractais e de ambientes de visualização científica, seguindo pela descrição do projeto de um ambiente para a exploração visual da estabilidade do algoritmo numérico fundamentado na teoria dos fractais. Uma síntese provisória dos resultados da utilização do sistema implementado é apresentada em um capítulo a parte assim como o encaminhamento para trabalhos futuros e perguntas formuladas a partir de explorações visuais realizadas através da utilização do sistema.

Palavras-chave: Fractais. Métodos Numéricos. Newton Fractal. Métodos Computacionais. Resolução de equações.

ABSTRACT

This work investigates the possibility of using the visual exploration in the identification of the stability of numerical methods. To conduct the research was chosen the theme of solving equations and the numerical method of Newton for calculate roots of equations. In fact, we used Newton's method to generate fractal images whose graphical shapes identify regions of attractors or basins of convergence. To understand the significance of these images, in the context of stability of numerical algorithms, are used more elaborate strategies of thought than simply "look"; are strategies that enable "see." Information carried by shapes, colors and shadows are objects of this study. This text comprises the theoretical foundation of numerical methods, fractal theory and scientific visualization environments, followed by the description of a design environment for visual exploration of the stability of numerical algorithm based on fractal theory. An interim summary of results using the implemented system is presented in a separate chapter as well as referral for further work and questions from visual explorations performed using the system.

Keywords: Fractals. Numerical Methods. Newton Fractal. Computational Methods. Solving equations.

LISTA DE ILUSTRAÇÕES

SUMÁRIO

1. INTRODUÇÃO

> "*A geometria fractal vai fazer você ver tudo de forma diferente. Existe um risco em aprendê-la. Você arrisca perder sua visão de infância das nuvens, galáxias, florestas, flores, montanhas, e muito mais. Nunca mais você interpretará estas coisas do mesmo jeito*" (BARNSLEY, 1993).

Segundo Euclides, matemático grego que viveu há dois milênios atrás, um ponto não tem dimensão, ou melhor, tem dimensão zero. Uma linha, a distância entre dois pontos quaisquer, é algo com uma dimensão. A capa de um livro tem duas dimensões, pois para conhecer sua área é preciso multiplicar dois números, o do comprimento pelo da largura. Do mesmo modo, um bloco possui três dimensões, portanto precisamos multiplicar três dimensões para saber o seu volume. Euclides estava certo, mas não resolveu todo o problema. Os contornos das montanhas, a superfície dos pulmões humanos, a trajetória das gotículas de água quando penetram na terra, existe uma infinidade de fenômenos na natureza que não podem ser descritos pela geometria euclidiana em um nível adequado de exatidão. É preciso apelar para complicados cálculos que resultam nas chamadas dimensões fracionárias, como a dimensão 0.5, por exemplo, típica de um objeto que é mais do que um simples ponto com dimensão zero, porém menos que uma linha com uma dimensão. Só a chamada geometria dos fractais consegue descrevê-lo.

Os fractais deram origem a um novo ramo da matemática, muitas vezes designado como a geometria da natureza. As formas estranhas e caóticas dos fractais descrevem alguns fenômenos naturais, como o desenvolvimento das árvores, a forma de algumas raízes, a linha de costa marítima, a anatomia dos animais. Estes fenômenos eram considerados anomalias do ponto de vista da geometria euclidiana, que trabalha com esferas perfeitas, linhas retas, triângulos, etc. Além disto, a geometria fractal se aplica na astronomia, na meteorologia, na economia, nas artes, na análise numérica e em diversos outros campos ainda nem imaginados [7]. Para os biólogos, ajuda a compreender o crescimento das plantas. Para os físicos, possibilita o estudo de superfícies intrincadas. Para os médicos, dá uma nova visão da anatomia interna do corpo. Enfim, não faltam exemplos. Um dos mais belos - e, sem dúvida, o mais colorido - é o uso dos fractais na arte. Quando os computadores são alimentados com equações, eles criam magníficos desenhos abstratos.

Neste trabalho pretendemos investigar a aplicação da teoria dos fractais na análise numérica. Em especial, na resolução de equações através do método numérico de Newton. Para isto, será realizado um estudo do método de Newton, que quando ampliado para o plano complexo gera interessantes padrões fractais. Além disto, será apresentada uma fundamentação teórica sobre os fractais, contendo características, exemplos e aplicações. Uma vez que os fractais e o método de Newton estiverem entendidos, estaremos aptos a entrar em um novo tópico que combina estes dois assuntos, chamado **Newton Fractal**.

O trabalho relaciona-se com diversas disciplinas do curso de Ciência da Computação e é também interdisciplinar por envolver outras disciplinas, como a disciplina de cálculo, por exemplo. Entretanto, a ênfase do trabalho consiste na modelagem computacional da resolução de problemas matemáticos através de métodos numéricos.

1.1. MOTIVAÇÃO

Suponha que o problema em questão consiste em calcular o valor das raízes de f(x) utilizando algoritmos numéricos e que isto será realizado utilizado o método de Newton. Este método produz uma sequência de aproximações para o valor da raiz a partir de um valor inicial (chamado também de estimativa inicial). Se o valor inicial estiver perto o suficiente de uma raiz, ele pode ser melhorado aplicando uma função e usando o resultado dessa função como um novo valor inicial para uma estimativa ainda melhor, e assim por diante. Isto leva a um processo iterativo que deve convergir para a raiz [6]. Dependendo do valor inicial a sequência de valores pode convergir ou não para a raiz, assim como alguns valores convergem mais rapidamente que outros [4]. Portanto, resumidamente falando, o método de Newton se fundamenta no refinamento sucessivo da estimativa inicial através da obtenção de uma sequência de aproximações convergente para a raiz da equação.

Talvez a escolha do melhor valor inicial possa ser baseada em uma evidência numérica ou gráfica. Disto decorrem duas perguntas:

- Dado um valor inicial z_0, para qual das raízes do polinômio o método vai convergir?
- Dado um valor inicial z_0, o quão rápido ele vai convergir para uma raiz, em comparação com outros valores iniciais?

Estas perguntas podem ser respondidas quando “ampliamos” o método de Newton para o plano complexo e assim obtemos um padrão fractal.

Portanto, o Trabalho de Conclusão visa responder estas perguntas através da utilização do método de Newton Fractal. Pretendemos explorar possibilidades do método de Newton Fractal para definir regiões de estimativas iniciais ótimas, ou seja, regiões cujos valores produzem sequências de aproximações convergentes e que possuem o menor número possível de elementos. Desta forma, teremos valores que quando usados como estimativas iniciais do método de Newton vão convergir para uma raiz com um mínimo de iterações possíveis.

1.2. OBJETIVO GERAL

O objetivo geral do Trabalho de Conclusão é realizar a análise gráfica da estabilidade do algoritmo numérico para cálculo de raízes através do método de Newton. Para isto, são utilizadas imagens para visualizar a região de atratores, segundo a teoria dos fractais. As imagens são geradas através de software específico, cuja implementação faz parte deste Trabalho de Conclusão.

1.3. OBJETIVOS ESPECÍFICOS

Para atingir o objetivo geral foram definidos os seguintes objetivos específicos:

- Realizar uma exploração gráfica da região de convergência das raízes de um polinômio e direcionar esta exploração para a visualização científica.

- Investigar possibilidades de uso de bibliotecas numéricas já existentes no desenvolvimento do projeto de software específico para visualização de imagens geradas a partir da aplicação do método de Newton fractal.
- Analisar possibilidades de otimização do algoritmo Newton Fractal.
- Investigar sobre relações de interesse na área de métodos numéricos como, por exemplo, precisão e tempo para gerar as imagens fractais; bacias de convergência e raízes de equações.
- Prospectar exemplos onde a exploração visual do comportamento de algoritmos numéricos justifique sua necessidade de uso.

2. FUNDAMENTAÇÃO TEÓRICA

2.1. MÉTODOS NUMÉRICOS PARA RESOLUÇÃO DE EQUAÇÕES

> "Métodos analíticos são usados para resolver equações, porém apenas uma parte destes problemas pode ser resolvida desta forma. A outra parte destes problemas é abordada com estratégias numéricas que tentam chegar a soluções aproximadas." (KOLBERG, BOCIAN e CLÁUDIO, 2001)

Com Bhaskara, tornou-se conhecida o algoritmo para resolução de uma equação de 2º grau. A resolução de equações do 3º grau foi encontrada por Nicolo Fontana, cognominado Tartaglia, cujo trabalho foi publicado por Jerônimo Cardano [8]. Por muito tempo procuraram-se fórmulas para a equação do 5º grau e graus superiores. Contudo, em 1824, Niels Abel provou que não é possível resolver equações de grau superior a quatro por radicais e combinação dos coeficientes. Este teorema, juntamente com o teorema fundamental da Álgebra, enunciado por D'Alembert em 1746, que afirma: "Toda equação polinomial de grau *n* possui exatamente *n* raízes" e que foi demonstrado por Gauss em 1799, conclui a pesquisa das fórmulas de equações de grau *n* [8].

Desde então, são utilizados métodos numéricos para cálculo das *n* raízes de um polinômio de grau *n*. Os métodos numéricos para o cálculo das raízes de equações algébricas são classificados em três tipos [8]:

- **Métodos de quebra** (Exemplo: método da bissecção)
  - ✓ Partimos de um intervalo [a, b] onde a função troca de sinal.
  - ✓ Quebramos o intervalo em dois intervalos e prosseguimos com o intervalo que mantém a troca de sinal (ou seja, contém a raiz).
  - ✓ São os mais intuitivos geometricamente, contudo são os que convergem mais lentamente.

- **Métodos de ponto fixo** (Exemplo: método de Newton)
  - ✓ Começamos com uma aproximação inicial x_0 e construímos iterativamente uma sequência de aproximações da raiz.
  - ✓ Dependendo da escolha do ponto inicial, x_0, o método pode convergir ou não. Quando converge é um dos métodos mais eficientes.

- **Métodos de múltiplos passos** (Exemplo: método das secantes)
  - ✓ São generalizações dos métodos de ponto fixo, que utilizam vários pontos anteriores (ao invés de um só) para determinar o próximo.
  - ✓ Utilizam os valores de *f* e suas derivadas.

Qualquer método numérico para o cálculo das raízes de equações algébricas é formado de quatro partes [8]:

- **Estimativa inicial:** uma ou mais aproximações para a raiz desejada.
- **Atualização:** uma fórmula que atualize a solução aproximada.
- **Critério de parada:** uma forma de estabelecer quando parar o processo iterativo em qualquer caso.
- **Estimador de exatidão:** está associado ao critério de parada e provê uma estimativa do erro cometido.

2.2. MÉTODO DE NEWTON

> "Se eu vi mais longe, foi por estar de pé sobre ombros de gigantes." (Isaac Newton, 1676)

O método de Newton é um dos melhores métodos conhecidos para calcular sucessivas aproximações dos zeros polinomiais. Este método pode convergir “muito rapidamente”, especialmente se a iteração começa “suficientemente próxima” da raiz desejada. O quão perto ("suficientemente próximo") deve ser e quão rapidamente ("muito rapidamente") pode ser, depende do problema. Infelizmente, quando a iteração começa longe da raiz desejada, o método de Newton pode facilmente levar um usuário desavisado a se perder sem aviso prévio [3].

Dada uma função f(x) e sua derivada f’(x), partimos de um valor inicial, denominado x_0, preferencialmente próximo a uma raiz que desejamos descobrir e calculamos as sucessivas aproximações da raiz x_1, x_2 da raiz através da fórmula [8] [11]:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

A ideia geométrica do método (Figura 1) é a partir de uma estimativa inicial x_0, identificar o ponto $(x_0, f(x_0))$, traçar a reta tangente que passa por este ponto e obter a estimativa x_1 a partir da identificação do ponto de interseção da reta tangente com o eixo x. Assim, sucessivamente, são gerados os valores x_{n+1} da sequência iterativa até que os sucessivos valores estejam extremamente próximos da raiz [11]. Ou seja, o método é repetido usando a aproximação recém calculada para gerar outra aproximação, e depois outra aproximação, até que os sucessivos valores estejam extremamente próximos e você conclua que possui uma ótima aproximação de uma das raízes da função.

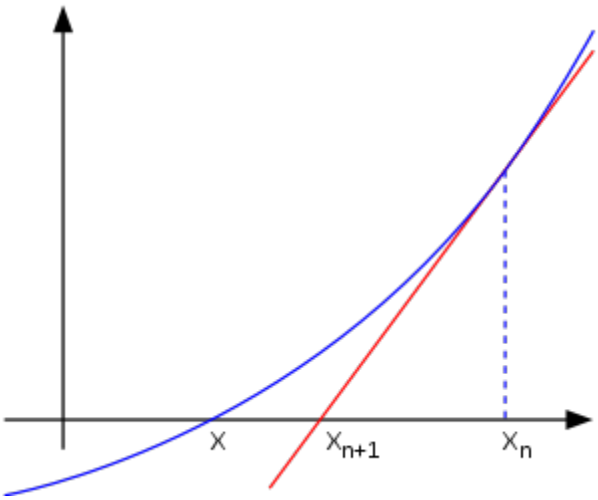

Figura 1: Uma ilustração de uma iteração do método de Newton. A função *f* é mostrada em azul e a linha tangente a $f(x_n)$ em vermelho. Vemos que x_{n+1} é uma aproximação melhor que x_n da raiz x.
Fonte: http://en.wikipedia.org/wiki/Newton's_method

A ordem de convergência do método de Newton é quadrática para uma raiz de multiplicidade um e linear para uma raiz de multiplicidade maior que um [3]. Quando a convergência é quadrática, o número exato de dígitos dobra (aproximadamente) a cada iteração [11]. Além disto, existem algumas outras observações importantes sobre o método:

- Requer o cálculo da derivada.
- A fórmula para o método de Newton irá falhar nos casos em que a derivada é zero. Da mesma forma, quando a derivada é próxima de zero, a tangente é quase horizontal e, portanto, pode ultrapassar a raiz desejada.
- Se o valor inicial está muito longe do valor da raiz e dependendo do comportamento da função, o método de Newton pode não convergir para a raiz.
- Quando existir duas ou mais raízes juntas, podem ser necessárias muitas iterações antes que seja possível chegar perto o suficiente de uma das raízes para a convergência quadrática ser aparente.
- Funciona melhor para as funções com curvatura baixa.

Vamos analisar, por exemplo, o problema de calcular a raiz quadrada de 612. Existem muitos métodos de calcular a raiz quadrada e o método de Newton é um deles. Então, para calcular o valor da raiz será realizado o seguinte procedimento:

Considere que $x = \sqrt{612}$, o que é o mesmo que $x^2 = 612$. Portanto,

$$f(x) = x^2 - 612$$

e sua derivada é calculada por

$$f'(x) = 2x$$

Utilizando o valor 10 como estimativa inicial, obtemos a seguinte sequência iterativa:

$$\begin{array}{rcccccl}
x_1 & = & x_0 - \dfrac{f(x_0)}{f'(x_0)} & = & 10 - \dfrac{10^2 - 612}{2 \cdot 10} & = & 35.6 \\
x_2 & = & x_1 - \dfrac{f(x_1)}{f'(x_1)} & = & 35.6 - \dfrac{35.6^2 - 612}{2 \cdot 35.6} & = & \underline{2}6.3955056 \\
x_3 & = & \vdots & = & \vdots & = & \underline{24.}7906355 \\
x_4 & = & \vdots & = & \vdots & = & \underline{24.7386}883 \\
x_5 & = & \vdots & = & \vdots & = & \underline{24.7386338}
\end{array}$$

cujos dígitos significativos exatos estão sublinhados. Assim como pode ser constado através dos valores da sequência, a solução e obtida com apenas algumas poucas iterações.

2.3. FRACTAIS

> "*Nuvens não são esferas, montanhas não são cones, linhas costeiras não são círculos, cascas de árvores não são suaves, nem o raio se propaga em linha reta." (MANDELBROT, 1982)*

A ideia dos fractais teve a sua origem entre 1875 e 1925, mas foi preciso alguns anos para que os matemáticos reconhecessem seu valor científico. Atualmente, os fractais constituem uma área importante de investigação matemática [5].

Como os fractais podem assumir uma infinidade de formas, é difícil criar uma definição para eles. Mandelbrot elaborou um conceito mais matemático, que diz: "Um conjunto é dito Fractal se a dimensão Hausdorff deste conjunto for maior do que

sua dimensão topológica". Outra definição mais simples diz que: "Fractais são objetos gerados pela repetição de um mesmo processo recursivo, apresentando auto-semelhança e complexidade infinita." Os fractais são caracterizados por possuir pelo menos alguma das seguintes propriedades:

● **Auto-semelhança:** um pequeno pedaço do fractal é similar ao todo, ou seja, visto em diferentes escalas a imagem de um fractal parece similar [2].

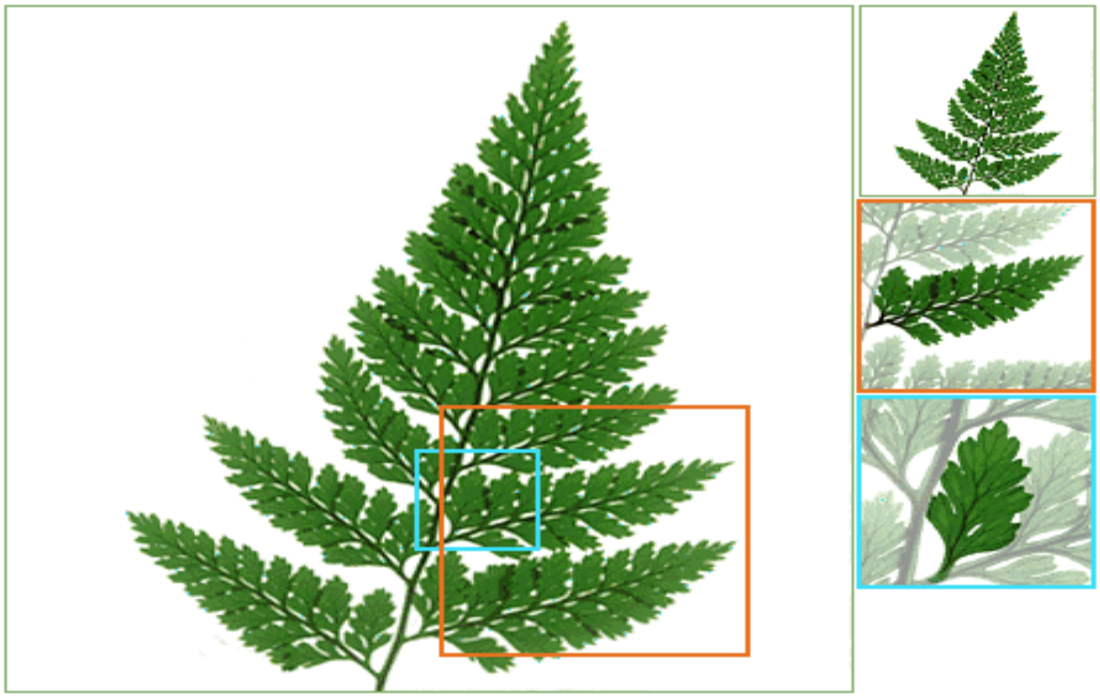

Figura 2: Auto-similaridade em uma folha de samambaia: as ampliações, sinalizadas na imagem laranja e azul são muito semelhantes à imagem original.
Fonte: http://www.educ.fc.ul.pt/icm/icm99/icm43/exempl_f.htm

● **Complexidade infinita:** propriedade relacionada com o fato do processo gerador do fractal ser recursivo, com um número infinito de iterações. Isso significa que não conseguiremos representar completamente um fractal, pois a quantidade de detalhes é infinita.

● **Dimensão fractal:** ao contrário do que acontece na geometria euclidiana, nem sempre um fractal possui uma dimensão inteira, ou seja, a dimensão pode ser fracionária. A dimensão de um fractal representa o seu grau de ocupação no espaço, o que está relacionado com seu grau de irregularidade.

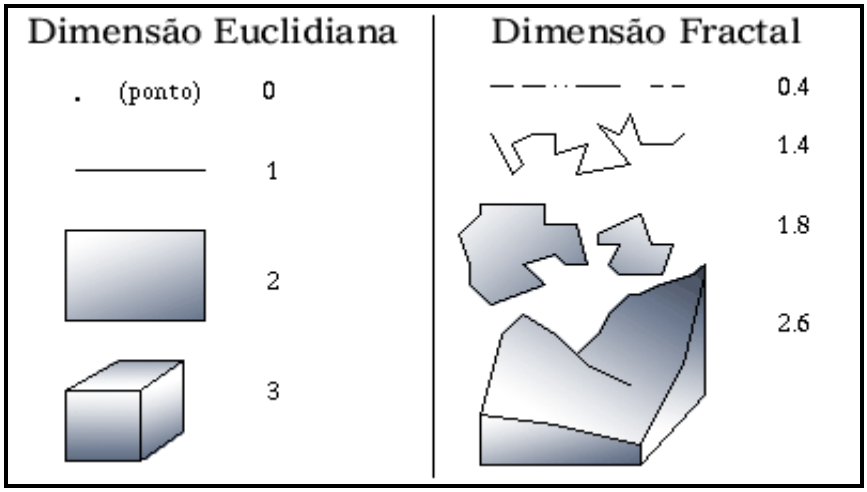

Figura 3: Dimensão Fractal: uma nova geometria e um novo conceito de dimensão precisaram ser criados para explicar a geometria das formas intrincadas.
Fonte: http://www.insite.com.br/fractarte/artigos.php

Para introduzir a ideia de que a dimensão fractal não é necessariamente inteira, Mandelbrot utilizou o seguinte exemplo: Qual é a dimensão de um novelo de fio? Mandelbrot respondeu que isso depende do ponto de vista. Visto de grande distância, o novelo não é mais do que um ponto, com dimensão zero. Visto mais de perto, o novelo parece ocupar um espaço periférico, assumindo assim três dimensões. Visto ainda mais de perto, o fio torna-se visível, e o objeto torna-se de fato unidimensional, ainda que essa dimensão única se enovele em volta de si mesma de tal forma que ocupe um espaço tridimensional [5]. Como podemos perceber, Mandelbrot estava salientando que um mesmo objeto pode ser diferentemente descrito e compreendido quando observado em escalas espaciais distintas.

Existem diversos tipos de fractais, que podem ser classificados segundo o modo como são formados ou gerados [7]:

● Fractais definidos por uma relação de recorrência em cada ponto do espaço. É o caso do Conjunto de Mandelbrot (Figura 4), fractal definido como o conjunto de pontos *c* no plano complexo para o qual a sequência abaixo definida iterativamente não tende ao infinito:

$$z_0 = 0$$

$$z_{n+1} = z_n^2 + c$$

Ou seja, para desenhar **cada ponto *c*** deste este fractal é preciso calcular, se naquele ponto *c* a sequência abaixo tende ou não para o infinito:

$c = x + yi$

$z_0 = 0$

$z_1 = z_0^2 + c$, substituindo temos $z_1 = x + yi$

$z_2 = z_1^2 + c$, substituindo temos $z_2 = x^2 - y^2 + x + (2xy + y)i$

$z_3 = z_2^2 + c$, e assim por diante.

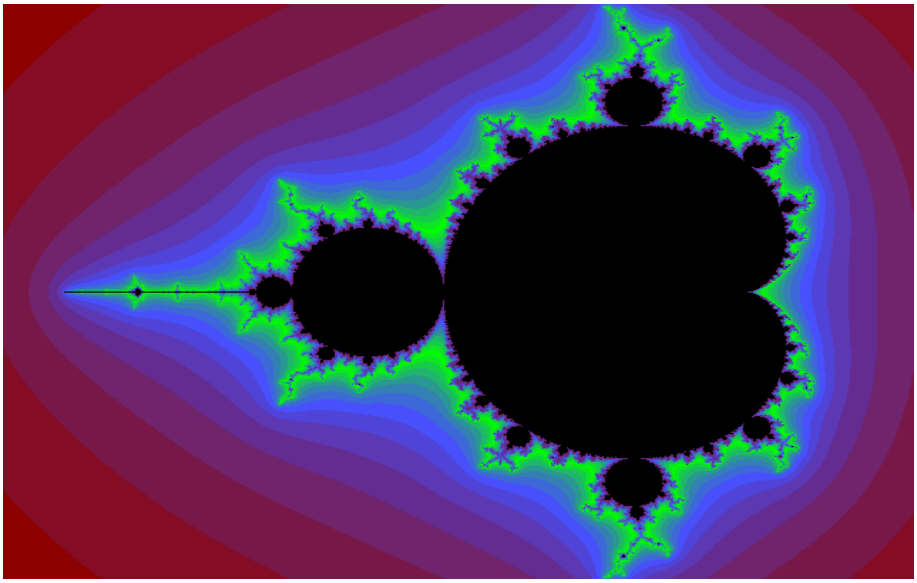

Figura 4: Conjunto de Mandelbrot. Adicionam-se cores para tornar a desenho mais bonito, mas este mesmo conjunto pode ser colorido de infinitas formas.
Fonte: http://commons.wikimedia.org/wiki/Mandelbrot_set

- Fractais aleatórios, gerados por processos estocásticos ao invés de determinísticos (Figura 5).

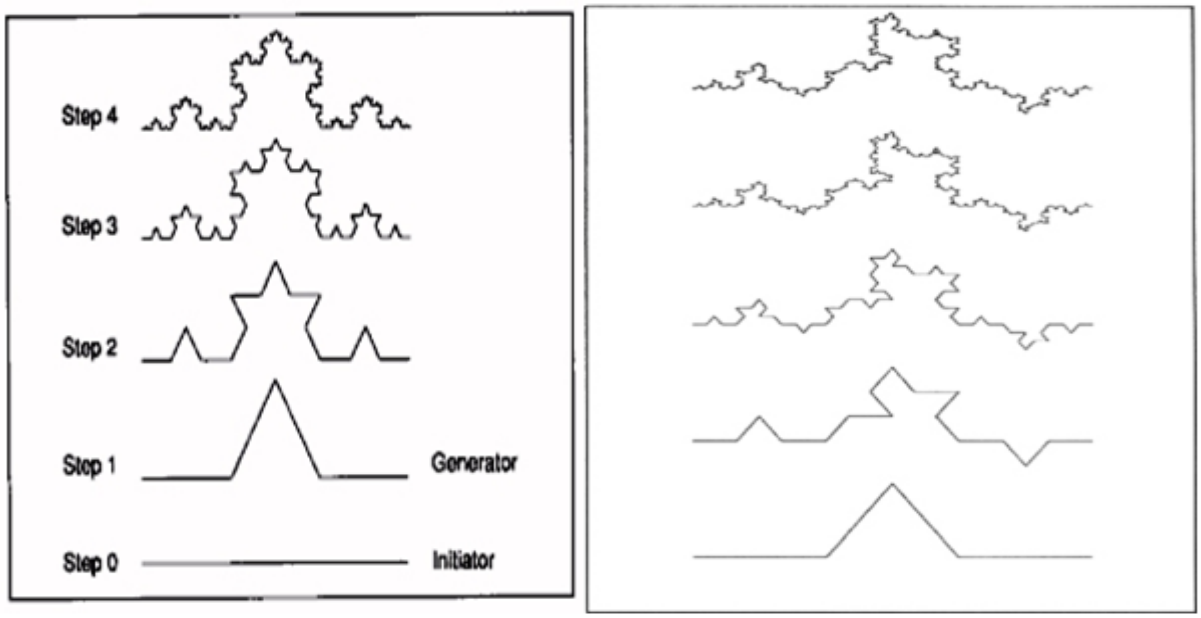

Figura 5: Na esquerda, o fractal da curva de Koch. Na direita, o mesmo fractal gerado de forma aleatória, a cada iteração, por exemplo, é jogada uma moeda para decidir o lado do segmento.
Fonte: http://algos.inesc.pt/~pjjr/MoiSelf.html

● Ponto fixo de um sistema de funções iteradas: um conjunto de funções é aplicado sucessivamente, a um subconjunto compacto de um espaço métrico, um número infinito de vezes (Figura 6).

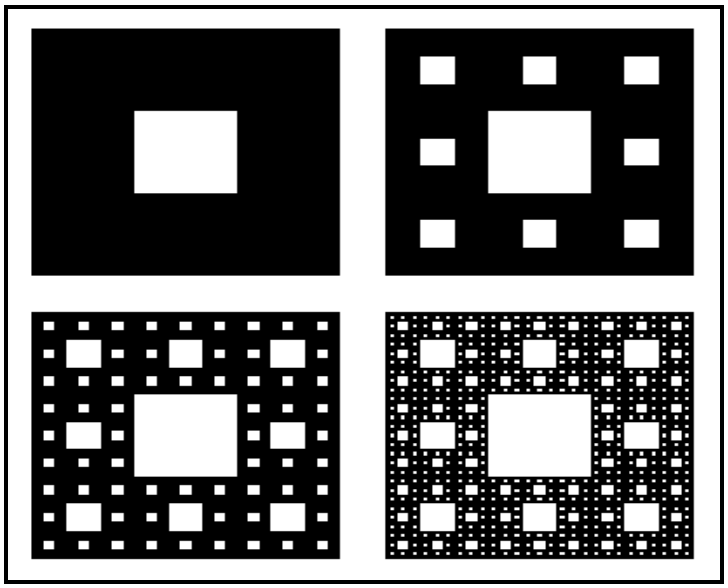

Figura 6: Sierpinski carpet: começa com um quadrado preto que é divido em nove quadrados. Remove-se o do meio e continua o mesmo processo nos oito quadrados restantes.
Fonte: http://mathforum.org/advanced/robertd/carpet.html

Sobre as características da geometria fractal na natureza, Alves [7] (p. 34) diz:

> "Não há fractais 'puros' na Natureza, tal como nela também não existem esferas perfeitas. Os fractais podem ser úteis para modelar objetos e

> fenômenos naturais, desde a escala atômica (na turbulência de ruídos, por exemplo) até ao tamanho do Universo (constituição de galáxias, por exemplo), porém, em cada caso, a sua auto-semelhança não será repetida infinitamente, mas apenas num determinado intervalo de escalas."

A aplicação dos fractais está sempre limitada a um intervalo de escalas, fora do qual a propriedade da auto-semelhança, seja exata ou estatística, já não se verifica. Como não se pode aplicar diretamente a teoria às situações práticas, a modelagem de objetos ou fenômenos com a geometria fractal é feita considerando uma série de aproximações que dependerão do grau de exatidão que se pretende nos resultados finais (Figura 7) [7]. Mas estas aproximações também são feitas na geometria euclidiana, que estuda o planeta Terra como se fosse uma esfera perfeita.

Figura 7: Relâmpago com formato fractal. "... nem os relâmpagos viajam em linha reta" disse Mandelbrot para ilustrar as propriedades fractais da natureza. Fonte: Alves, 2007.

Muitos vegetais crescem de forma ramificada, ou seja, o tronco subdivide-se em vários ramos que, por sua vez, se subdividem em ramos mais estreitos e assim por diante, como se fosse uma função recursiva. Além disso, geralmente uma parte da planta tem uma forma semelhante à planta completa, como pode ser observado na Figura 8.

Figura 8: Elementos da flora que exibem a repetição da mesma forma em várias escalas. Fonte: Alves, 2007.

Alguns dos exemplos de fractais com estrutura mais intrincada e interessante encontram-se na fisiologia animal, nos sistemas respiratório, circulatório e nervoso, que possuem uma estrutura altamente ramificada (Figura 9). A ramificação fractal, além de proporcionar mais tolerância a defeitos no crescimento e a danos, amplifica a área da superfície de um tecido, com os objetivos de absorção (pulmão, intestino) ou distribuição e colheita (vasos sanguíneos, bronquíolos) ou para processamento de informações (neurônios) [7].

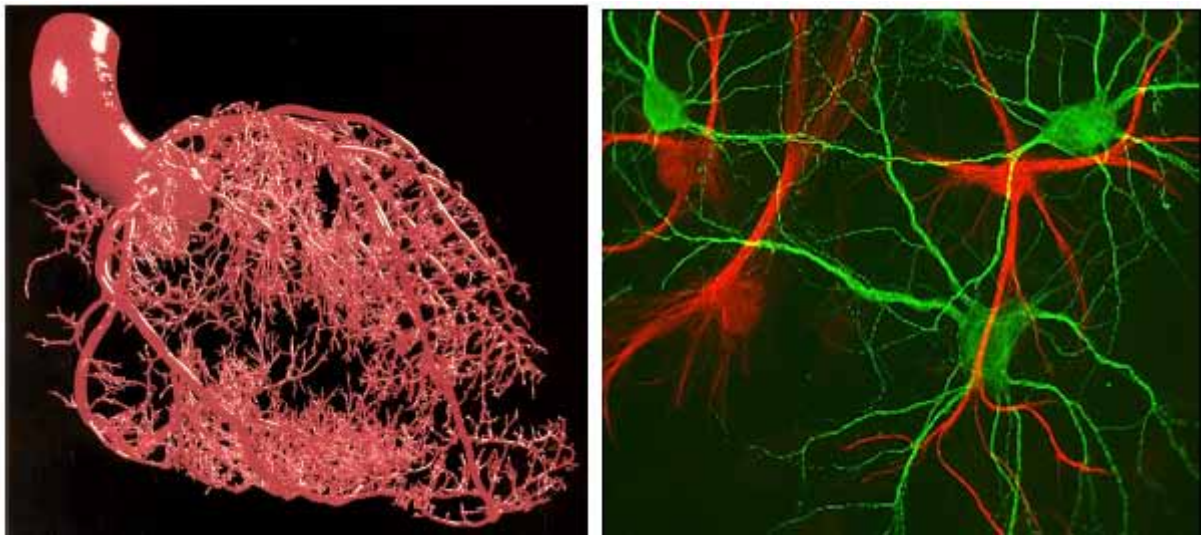

Figura 9: Respectivamente, modelo do sistema de irrigação do coração humano e neurônios do hipocampo (em verde) acompanhados das células da glia (em vermelho). Fonte: Alves, 2007.

Os rios são bons exemplos de fractais naturais, dada a sua forma ramificada e tortuosa (Figura 10). As cascatas também apresentam efeitos visuais de aspecto fractal, com ramificações sucessivas do curso da água, criadas pela combinação da

força da gravidade e a forma da superfície da rocha por onde cai a água (Figura 10) [7].

Figura 10: Respectivamente, vista área de uma rede fluvial e fotografia de uma cascata: em ambas as imagens são visíveis estruturas ramificadas auto-semelhantes. Fonte: Alves, 2007.

As montanhas são construídas por forças tectônicas e esculpidas pela água e pelo vento, que atuam de formas semelhantes, em escalas diferentes. Nas duas imagens seguintes podemos observar o formato fractal nas montanhas, que poderiam ser geradas por um algoritmo baseado na geometria fractal (Figuras 11 e 12) [7].

Figura 11: Montanhas onde facilmente se percebe a presença da geometria fractal.
Fonte: Alves, 2007.

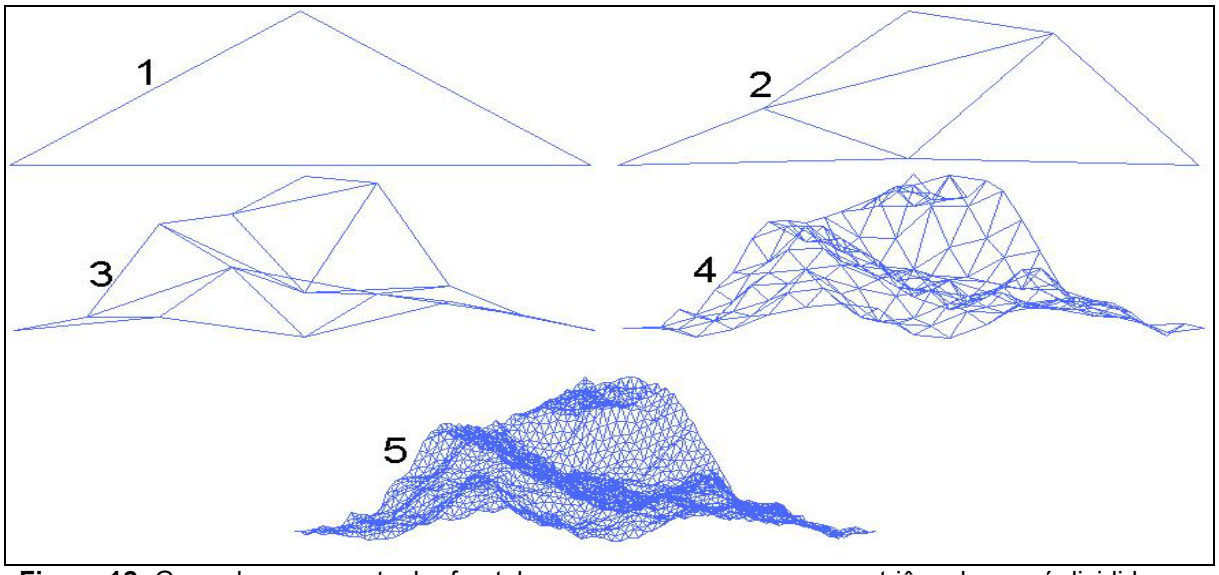

Figura 12: Gerando uma montanha fractal: o processo começa com um triângulo que é dividido em quatro triângulos e em cada um dos novos triângulos se aplica o mesmo processo.
Fonte: http://pt.wikipedia.org/wiki/Fractal

Fractais também podem ser criados pelo homem para diversos fins. Antenas com formato comum são sensíveis apenas a um número limitado de frequências e não são eficientes se a frequência for menor que um quarto do comprimento de onda (Figura 13). Isto transforma o desenho de antenas portáteis para celulares num desafio. Mas se a antena tiver um formato fractal, ela poderá ultrapassar algumas destas dificuldades: experiências demonstraram que as antenas cuja forma corresponde à imagem obtida após um pequeno número de iterações da construção de um fractal conseguem detectar várias frequências. À medida que o número de iterações aumenta, o campo de frequências detectáveis pela antena também aumenta. Além disso, as antenas fractais possuem um tamanho de um quarto das antenas comuns [7].

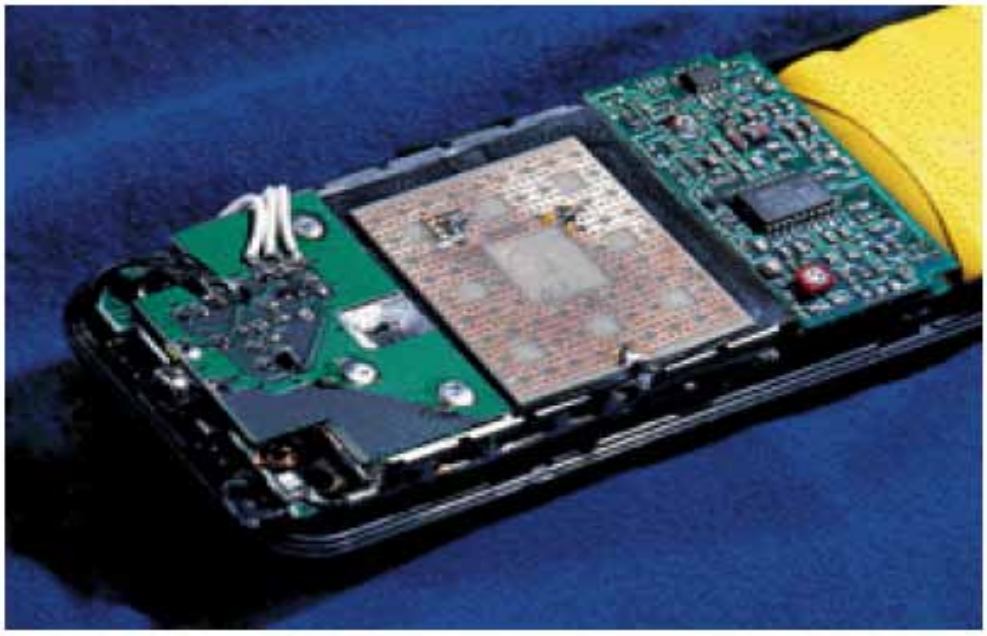

Figura 13: Exemplo de antena com o formato de um fractal. Fonte: Alves, 2007.

Outros exemplos de aplicações de fractais, segundo Alves [7]:

● Diagnóstico de células cancerosas: as células doentes tendem a ser mais ramificadas, com uma dimensão fractal maior que a das células sãs;

● Caracterização de padrões gerados por eletro encefalogramas em estudos sobre os diversos estados do cérebro durante o sono;

● O crescimento de populações urbanas em torno de um núcleo pode formar padrões com características fractais;

● Geração de música fractal: um processo iterativo é aplicado a uma nota musical inicial para produzir a nota seguinte, em seguida aplica-se a mesma fórmula à nota obtida para gerar a terceira nota, e assim sucessivamente;

● Análise de ações da bolsa: o mercado pode ser analisado como um fractal, pois o comportamento é estatisticamente igual em grande escala (um ano, por exemplo) e em escalas menores (uma semana ou um dia);

A geometria baseada em fractais é interdisciplinar, como mostram os exemplos. Embora tenham sido apresentados diversos exemplos e aplicações de fractais, é preciso salientar que nem tudo na natureza é fractal, e a geometria euclidiana continuará sendo útil e necessária [7].

2.4. NÚMEROS COMPLEXOS

> "Por muitas razões o conceito de número teve que ser aumentado para além dos números reais através da introdução dos chamados números complexos." (COURANT, ROBBINS e STEWART, 1996)

Os números complexos encontram aplicação em diversos problemas da matemática, física quântica, engenharia, teoria do caos, sobretudo da solução de equações algébricas e equações diferenciais. Os números complexos pertencem a um conjunto que é uma extensão do conjunto dos números reais, e são representados na forma $x + yi$, onde x e y são números reais e i representa a unidade imaginária igual a $\sqrt{-1}$ [9]. Ou seja:

$$i^2 = -1$$

Historicamente, a representação geométrica de um número complexo como simplesmente um ponto no plano foi importante, pois fez a idéia de um número complexo mais aceitável. Em particular, números "imaginários" passaram a ser aceitos, em parte, por meio de sua visualização (Figura 14) [9]. Os números complexos começaram a ser representados visualmente no plano complexo, também chamado de Plano de Argand-Gauss ou Diagrama de Argand, que consiste em um plano cartesiano usado para representar geometricamente os números complexos. Neste plano, a parte imaginária de um número complexo é representada pela ordenada (eixo y) e a parte real pela abscissa (eixo x).

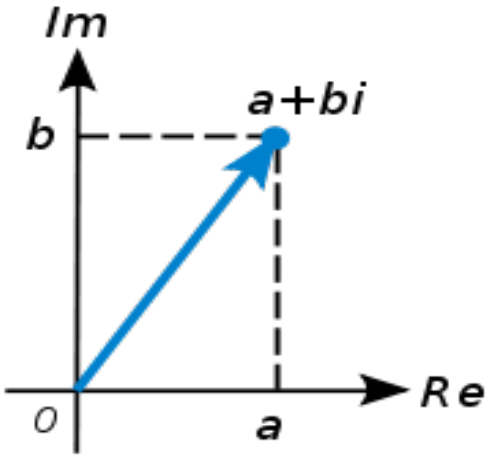

Figura 14: Um número complexo pode ser visualmente representado como um par de números formando um vetor em um diagrama chamado plano de Argand.
Fonte: http://en.wikipedia.org/wiki/Complex_number

Diferentemente dos números reais, números complexos não têm uma ordem natural, portanto, não há analogia das desigualdades de valores complexos. Esta propriedade não é tão surpreendente quando eles são vistos como sendo elementos no plano complexo, pois os pontos de um plano também não têm uma ordenação natural [9].

Algumas operações elementares com números complexos são exemplificadas a seguir [9]:

- Adição: $(a+bi) + (c+di) = (a+c) + i(b+d)$
- Subtração: $(a+bi) - (c+di) = (a-c) + i(b-d)$
- Multiplicação: $(a+bi) * (c+di) = (ac-bd) + i(ad+bc)$
- Divisão: $\frac{a+bi}{c+di} = \frac{(ac+bd)+i(bc-ad)}{c^2+d^2}$

2.5. RELAÇÃO ENTRE FRACTAIS E O MÉTODO DE NEWTON

> "Embora o método de Newton seja uma aplicação antiga do cálculo, apenas recentemente foi descoberto que ampliando ele para o plano complexo obtemos um interessante padrão fractal." (MCCLURE, 2010).

Vamos supor que desejamos analisar a função $f(x) = x^3 - 2$ (Gráfico 1). Podemos perceber no gráfico seguinte que este polinômio possui uma raiz real um pouco menor que 1.5. Usando um valor inicial de 1.5 nós geramos as seguintes aproximações usando o método de Newton: (1.5, 1.2963, 1.26093, 1.25992, 1.25992). Estamos na verdade calculando a raiz cúbica de 2, e o resultado 1.25992 é uma boa aproximação da resposta.

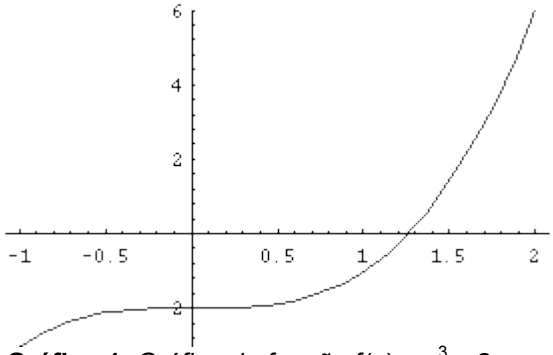

Gráfico 1: Gráfico da função $f(x) = x^3 - 2$.
Fonte: http://facstaff.unca.edu/mcmcclur/mathematicaGraphics/Newton/

A análise anterior não precisa ficar restrita aos números reais. O valor inicial poderia ser um número complexo (geralmente é usada a variável *z* para indicar que a função aceita valores complexos). A função anterior $f(x) = x^3 - 2$ possui apenas uma raiz. No entanto, quando esta função é considerada como uma função de variável complexa, $f(z) = z^3 - 2$, ela possui três raízes (Gráfico 2) [6].

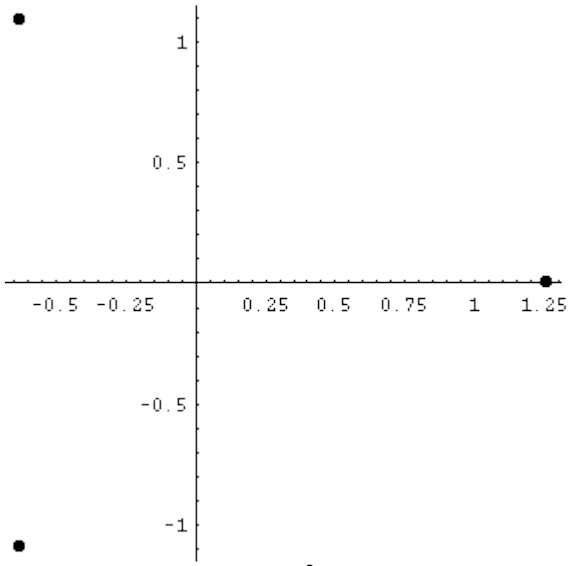

Gráfico 2: Posição das três raízes da função $f(z) = z^3 - 2$ representadas no plano complexo (parte real dos números no eixo x e parte imaginária no eixo y). Uma raiz é real e duas são imaginárias: 1.25992, -0.629961 - 1.09112i e -0.629961 + 1.09112i.

Em 1879, Cayley fez a seguinte pergunta: “Dada uma entrada inicial z0, para qual raiz o método de Newton irá convergir?” A resposta só foi plenamente entendida recentemente e leva a um belo padrão fractal. Podemos desenhar este padrão em um computador, da seguinte forma (Figura 15) [6]:

- Divida o plano, que no exemplo abaixo vai de [-2; 2] na parte real (eixo x) por [-2; 2] na parte imaginária (eixo y) em um grande número de pequenos pontos. Cada ponto corresponde a um número complexo.

- Execute o método de Newton em cada ponto para determinar para qual das três raízes o valor inicial vai convergir naquele ponto.

- É preciso escolher uma cor para cada raiz, neste caso são três cores porque o polinômio é de grau três (e por isso existem três raízes). Pinte o ponto dependendo do resultado do passo anterior.

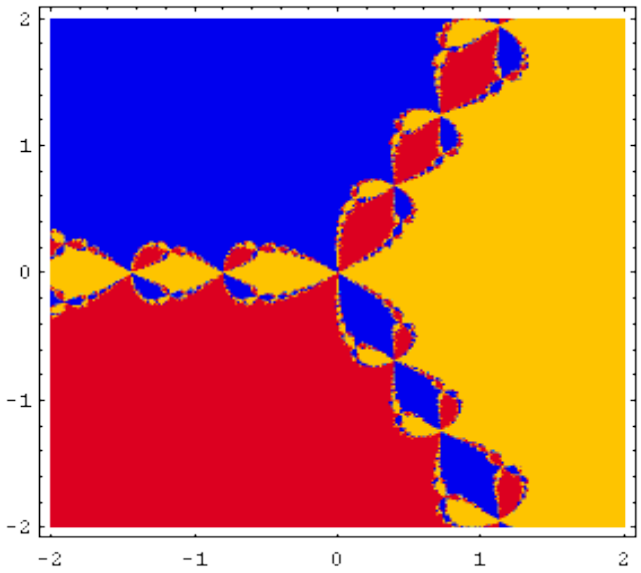

Figura 15: Fractal da função f(z) = z^3 – 1 no plano complexo. Cada ponto do gráfico é um número e sua cor representa para qual raiz aquele número converge quando usado como valor inicial no método de Newton. Fonte: http://facstaff.unca.edu/mcmcclur/mathematicaGraphics/Newton/

Suponha que não estivéssemos interessados em saber para qual raiz cada estimativa inicial vai convergir, mas que desejamos conhecer quanto tempo ou quantas iterações são necessárias para se obter uma boa aproximação das raízes. Neste caso, podemos gerar uma imagem sombreada que destaca a velocidade de convergência da sequência de Newton em cada ponto do plano (Figuras 16, 17 e 18) [6].

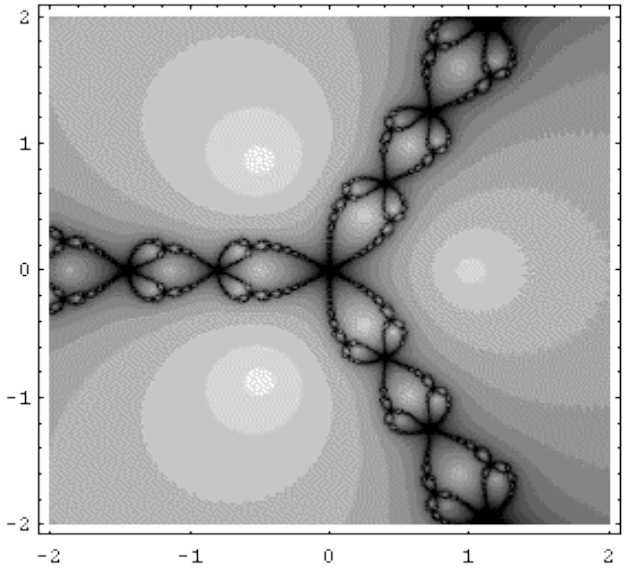

Figura 16: Fractal da função f(z) = z^3 – 1 no plano complexo. Cada ponto do gráfico é um número e quando mais claro ele for, mais rápido ele converge para uma das raízes quando usado como valor inicial no método de Newton. Fonte: http://facstaff.unca.edu/mcmcclur/mathematicaGraphics/Newton/

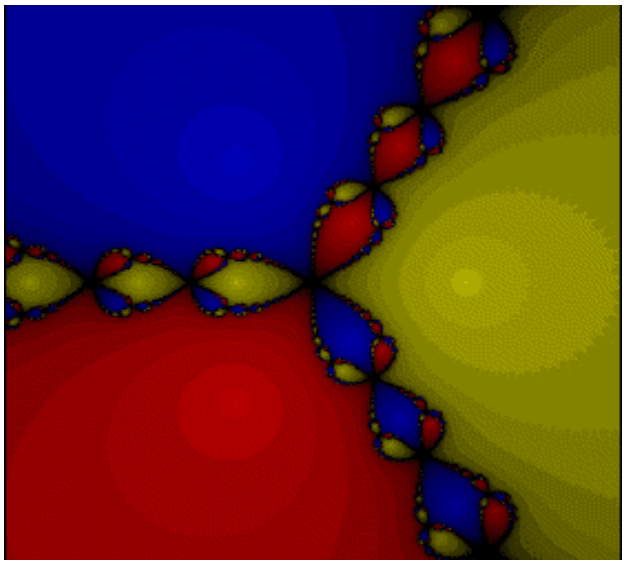

Figura 17: Fractal da função f(z) = z^3 – 1 no plano complexo. Combina as cores e o sombreamento e por isso leva em conta para qual raiz cada número converge e o quão rápido é essa convergência. Fonte: http://facstaff.unca.edu/mcmcclur/mathematicaGraphics/Newton/

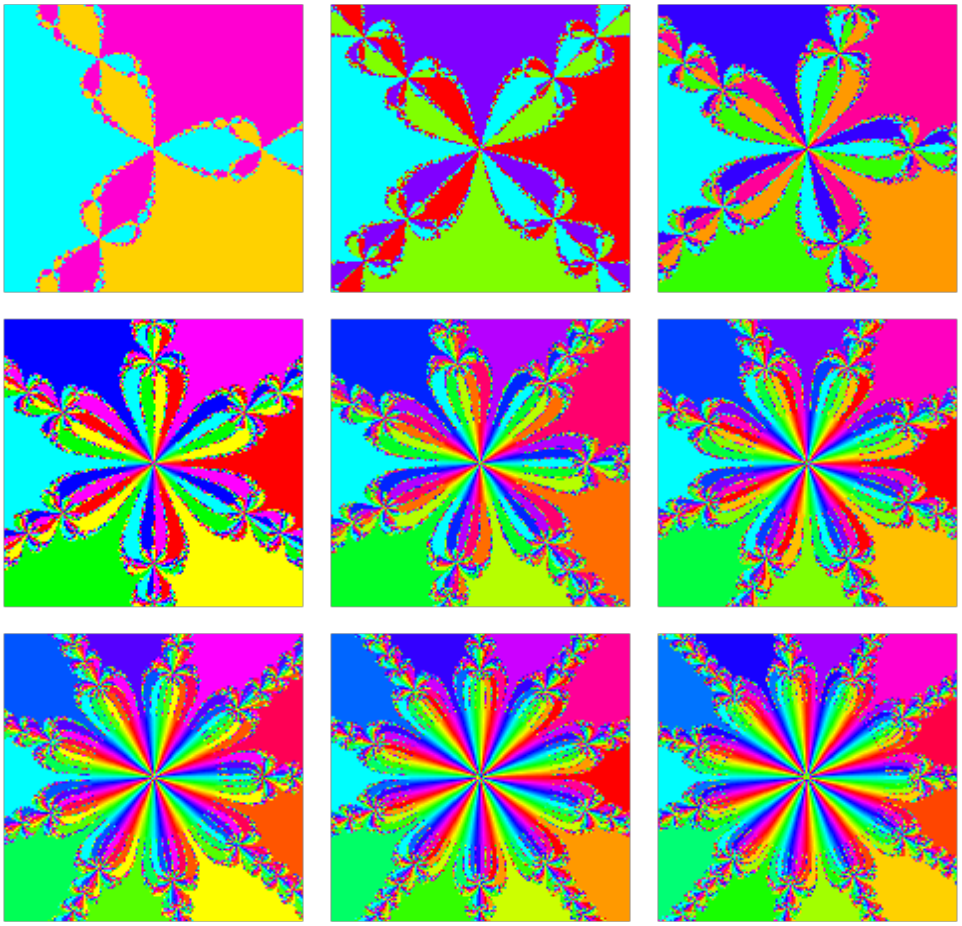

Figura 18: O primeiro fractal, na parte superior esquerda, usa três cores por se tratar de uma função de grau três. Cada imagem que segue aumenta um grau na equação e por isso usa uma cor a mais. Fonte: http://mathworld.wolfram.com/NewtonsMethod.html

Fractais também surgem de equações não-polinomiais [3]. A imagem abaixo mostra o número de iterações necessárias para o método de Newton para convergir para as funções $z^2 - 2^z$ e $z^3 - 3^z$ (Figura 19).

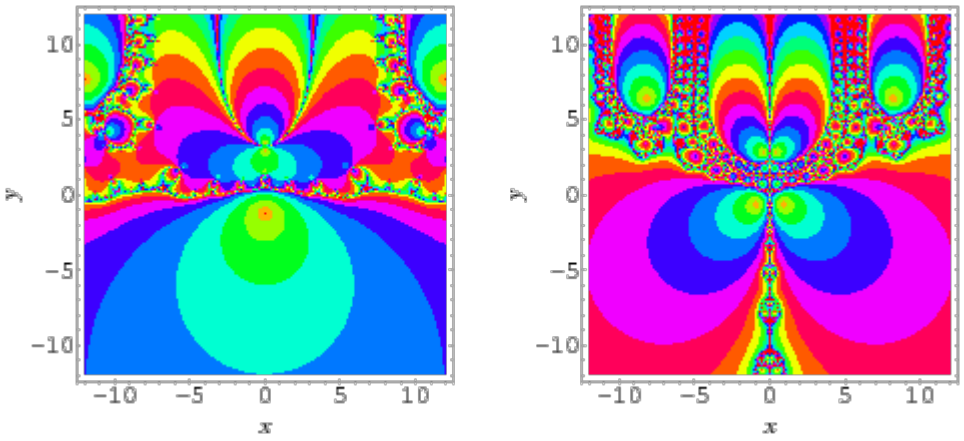

Figura 19: Número de iterações necessário para o método de Newton convergir nas funções $z^2 - 2^z$ e $z^3 - 3^z$, respectivamente. Fonte: http://mathworld.wolfram.com/NewtonsMethod.html

3. DOCUMENTAÇÃO DO SISTEMA DESENVOLVIDO

Um sistema de software se caracteriza por um conjunto de componentes abstratos (estrutura de dados e algoritmo) descritos na forma de procedimentos, funções, módulos e objetos que são organizados de forma a definir a arquitetura do sistema. Neste item estão descritos estes componentes.

3.1. LEVANTAMENTO DE REQUISITOS

Para definir o escopo do projeto de sistema foi realizada uma pesquisa visando identificar softwares para desenhar fractais Nestes softwares foram analisadas as visando a definição de funcionalidades do sistema a ser implementado.

O levantamento de funcionalidades possibilitou a construção do Quadro 1 onde foi utilizada a seguinte convenção:

Sistema A: http://vlab.infotech.monash.edu.au/simulations/fractals/fractals-on-the-complex-plane/demo/ - *Applet*, acessado em 13 de junho de 2010.
Sistema B: **Fractal Explorer** – disponível para download na web neste endereço: http://www.eclectasy.com/Fractal-Explorer/
Sistema C: **Dynamical Systems** – disponível para download na web neste endereço: http://archives.math.utk.edu/software/msdos/complex.variables/dy-syst/.html
Sistema D: **Fractal eXtreme** – disponível para download na web neste endereço: http://www.cygnus-software.com/downloads/downloads.htm

Funcionalidade 1 - Alterar o limite do plano complexo.
Funcionalidade 2 - Alterar o número limite de iterações do método de Newton.
Funcionalidade 3 - Alterar a paleta de cores.
Funcionalidade 4 - Zoom in, zoom out e mover a visão do plano.
Funcionalidade 5 - Exemplos de fractais.
Funcionalidade 6 - Alterar a equação geradora do fractal.

Funcionalidade 7 - Mostrar a posição das raízes no plano complexo.

Funcionalidade 8 - Dispor de sistema de ajuda.

	Software A	Software B	Software C	Software D
Funcionalidade 1	✔	✔	✔	✔
Funcionalidade 2	✔	✔	✔	✔
Funcionalidade 3	✔	✔	✔	✔
Funcionalidade 4	✔	✔	✔	✔
Funcionalidade 5	✔	✔	✔	✔
Funcionalidade 6	✘	✔	✔	✘
Funcionalidade 7	✘	✔	✔	✔
Funcionalidade 8	✘	✘	✔	✔

Quadro 1: Quadro comparativo das funcionalidades dos softwares visualizadores de Fractal

3.2. DESCRIÇÃO DO SISTEMA

O sistema implementado consiste em um ambiente para exploração visual da estabilidade do algoritmo que codifica o método numérico de Newton para cálculo de raízes. Para isto, o sistema implementa o método de Newton fractal. Foi desenvolvido um software para gerar imagens fractais associadas a equações. As imagens são coloridas a fim de identificar para qual raiz converge cada estimativa inicial ao utilizar o método de Newton. As imagens também podem ser sombreadas visando identificar quais valores convergem mais rapidamente. A forma gráfica das imagens possibilita visualizar bacias de atração, segundo a teoria dos fractais, e, portanto realizar a exploração visual da estabilidade do algoritmo numérico do método de Newton e também regiões do plano cartesiano onde os valores sempre produzem sequências convergentes e aquelas onde esta propriedade não se verifica.

A primeira versão do software foi um *applet* em Java, que consiste em um aplicativo que é incorporado em páginas *web* e executado pelo navegador. Ao longo do trabalho, sentimos a necessidade de realizar cálculos matemáticos e a solução adotada foi executar comandos do Octave em Java. Para isso, o programa em Java executa o Octave, envia comandos e recebe respostas. Entretanto, devido a

restrições de segurança, os *applets* por padrão não possuem permissão para acessar recursos do cliente, como o sistema de arquivos ou arquivos executáveis [13]. Por este motivo, optamos por transformar nosso *applet* em uma aplicação desktop normal, por exemplo, como o editor de texto Notepad.

O software recebe como entrada uma equação e gera o desenho do Newton Fractal correspondente. Para isso, os parâmetros de entrada são os coeficientes do polinômio, o intervalo de valores reais associado ao eixo x (parte real da raiz) e ao eixo y (parte imaginária da raiz), um valor de tolerância para cada raiz e o limite de iterações do método de Newton. Consideramos que a estimativa inicial convergiu à raiz quando o valor obtido pelo método numérico está dentro do seguinte intervalo:

[raiz – tolerância; raiz + tolerância]

O limite de iterações do método de Newton é outro parâmetro que pode ser alterado para a geração da imagem fractal. No Quadro 2 é apresentada a descrição detalhada dos parâmetros de entrada do sistema.

Parâmetro de entrada	**Descrição**	**Valor padrão (default)**
[minX; maxX]	Intervalo de valores do eixo x (parte real do número).	[-2; 2]
[minY; maxY]	Intervalo de valores do eixo y (parte imaginária do número).	[-2; 2]
limiteIteracoes	Número limite de iterações do método de Newton.	15
tolerancia	Limite de erro do valor obtido para cada raiz.	0.1
equacao	Equação que será usada para gerar o Newton fractal.	
raizes	Lista das s raízes da equação. Parâmetro opcional. Contém a multiplicidade.	

Quadro 2: Parâmetros de entrada do algoritmo de Newton Fractal, significados e valor padrão. Os valores padrões ainda não foram testados e, por isto, estão sujeitos a alteração. O objetivo deles é abstrair o nível de detalhe da execução do algoritmo e o usuário

3.3. DIAGRAMA DE CASOS DE USO

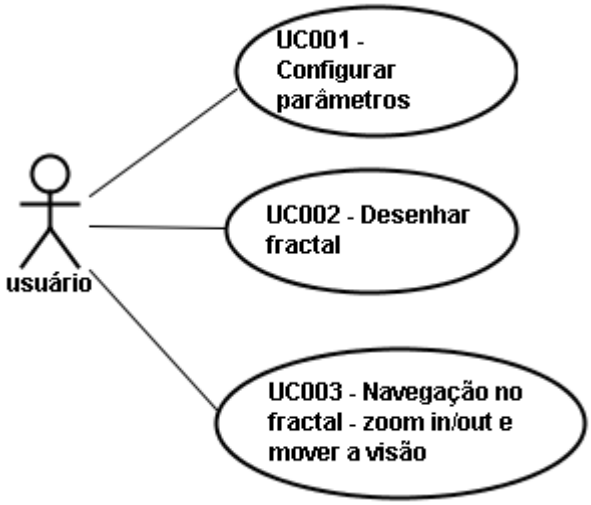

Figura 20: Diagrama de casos de uso exemplificando as funcionalidades do software.

3.4. PROJETO DE INTERFACE

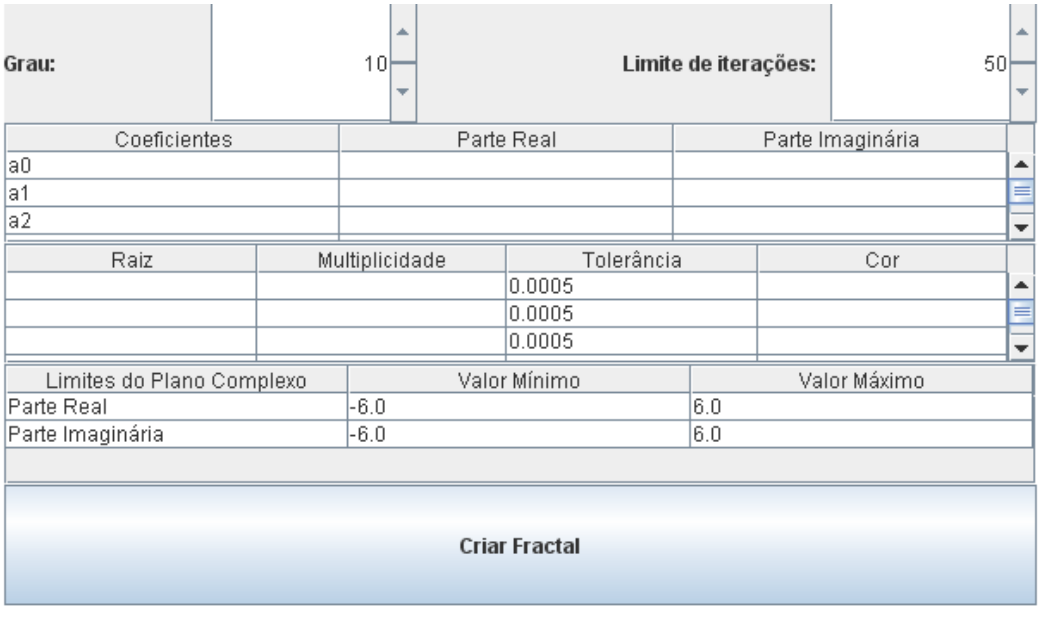

Figura 21: Primeira versão da interface do sistema.

A primeira versão de interface do sistema exigia a entrada de muitos parâmetros assim como consta na Figura 21. Visando tornar mais intuitiva e melhorar a programação visual da tela foi elaborada uma segunda e última versão onde a entrada de dados é separada em módulos e são utilizas abas para apresentar contextos (Figura 22).

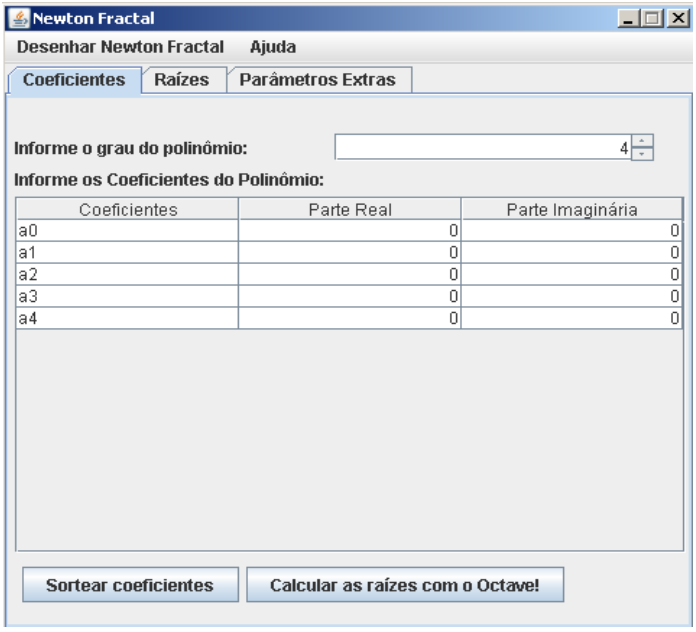

Figura 22: Versão final da interface, utilizando abas (JTabbedPane).

As abas apresentam os seguintes contextos:

- **Coeficientes:** Nesta aba são definidos os coeficientes do polinômio, lembrando que cada coeficiente possui parte real e parte imaginária (Figura 22).
- **Raízes:** Informações referentes às raízes estão apresentadas na Figura 23. O valor das raízes pode ser calculado pelo Octave, mas, além disso, é preciso definir a cor e a tolerância de erro de cada raiz. Para a escolha da cor, usamos a classe JColorChooser, ilustrada na figura 24.

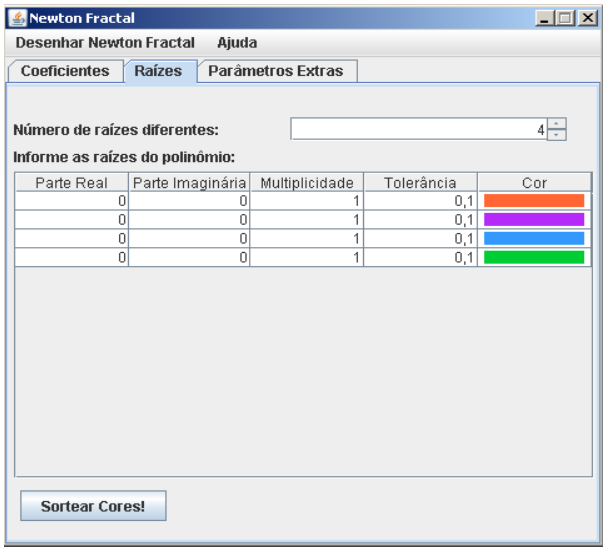

Figura 23: Aba Raízes para entrada de parâmetros referentes às raízes do polinômio.

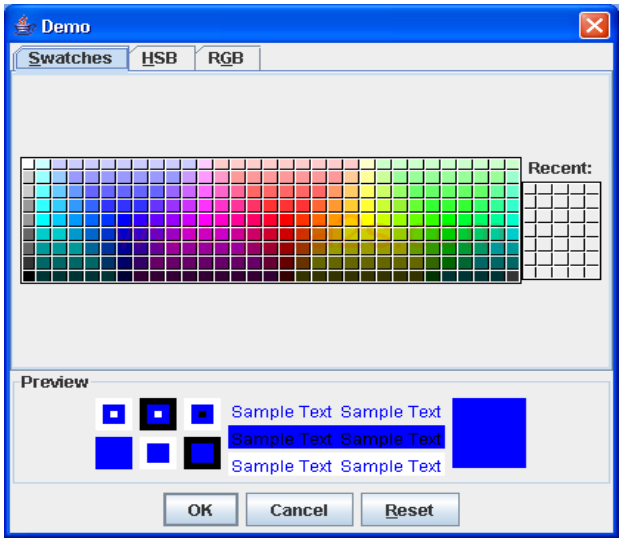

Figura 24: Exemplo de interface para seleção de cor (*JColorChooser*)

- **Parâmetros extras:** Nesta aba é informado o número limite de iterações do método de Newton e os limites do plano complexo, ou seja, os valores máximos e mínimos do eixo dos x e y (Figura 25).

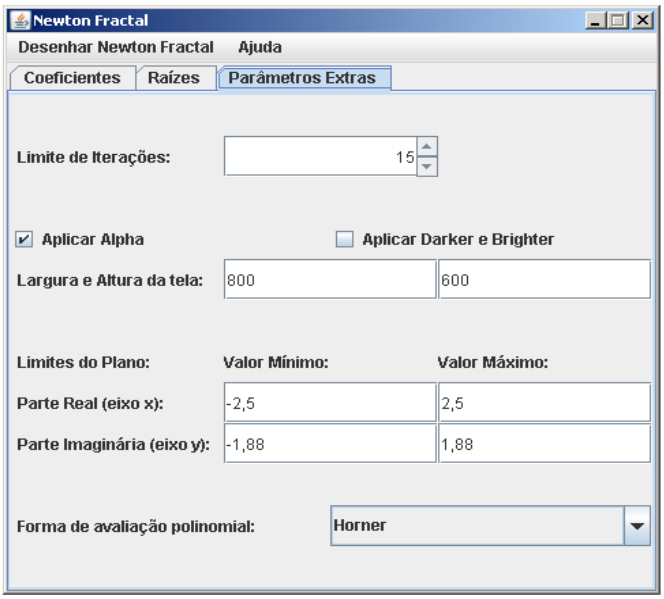

Figura 25: Aba Parâmetros Extras com informações adicionais para o desenho do Newton Fractal

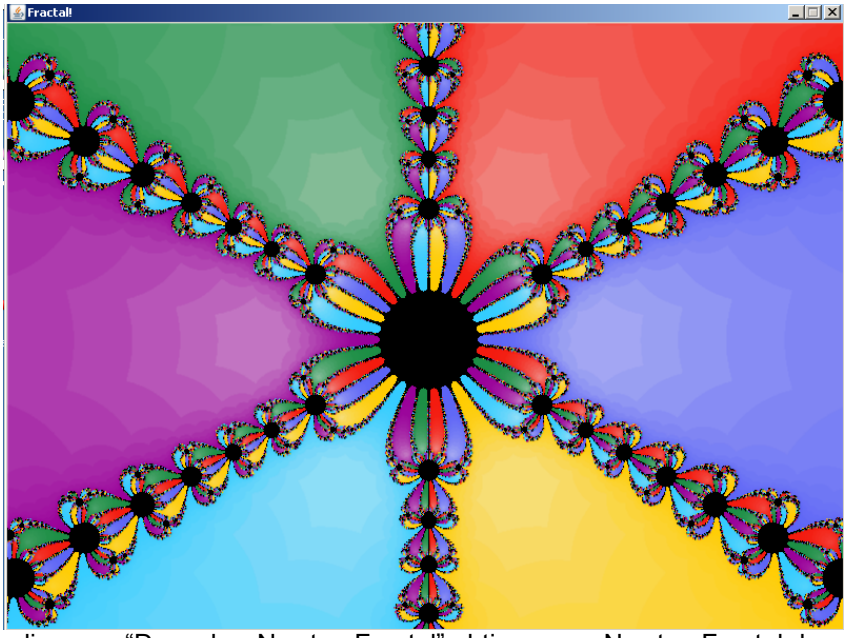

Figura 26: Ao clicar em “Desenhar Newton Fractal” obtivemos o Newton Fractal do polinômio $z^6 - 1$

3.5. OCTAVE

A possibilidade de usar o Octave para realizar computação matemática é importante, pois garante a qualidade dos resultados obtidos e libera da necessidade de implementar algoritmos numéricos. Também é importante explorar a possibilidade de integração entre Octave e a linguagem de programação Java, pois é bastante interessante dispor de aplicações matemáticas que executam via browser.

Nesta seção serão apresentadas características e funcionalidade do ambiente para programação matemática Octave e como a execução de comandos do ambiente foi integrada ao sistema implementado.

Octave é um software *open-source* para computação numérica e gráfica. O sistema foi especialmente projetado para computações de matrizes, resolução de sistema de equações, autovetores e autovalores [14].

Inicialmente, o Octave era um software que acompanhava um livro, em nível de graduação, sobre projeto de reatores químicos. Segundo Long [14]:

> "Atualmente o Octave está sendo desenvolvido sob a liderança do Dr. J.W. Eaton e liberado sob a licença GNU General Public Licence. A utilidade Octave é reforçada porque sua sintaxe é semelhante e compatível com a sintaxe do MATLAB, que é comumente usado na indústria e na academia."

De acordo com Long [14], a NASA usa o Octave para desenvolver sistemas de ancoragem de naves espaciais (*spacecraft docking systems*), Jaguar Racing utiliza o software para exibir e analisar dados transmitidos por seus carros de Fórmula 1 e a Universidade de Sheffield utiliza o Octave em um software para reconhecimento de células cancerosas.

Para calcular as raízes de polinômios, o Octave dispõe de uma função, chamada *roots*. A equação é inserida no Octave usando um vetor para representar o *array* de coeficientes. Por exemplo, na imagem seguinte (Figura 27), o vetor $[1.0\ \ 0\ \ 2-3i\ \ 5]$ corresponde ao polinômio $f(z) = z^3 + (2-3i)z + 5$.

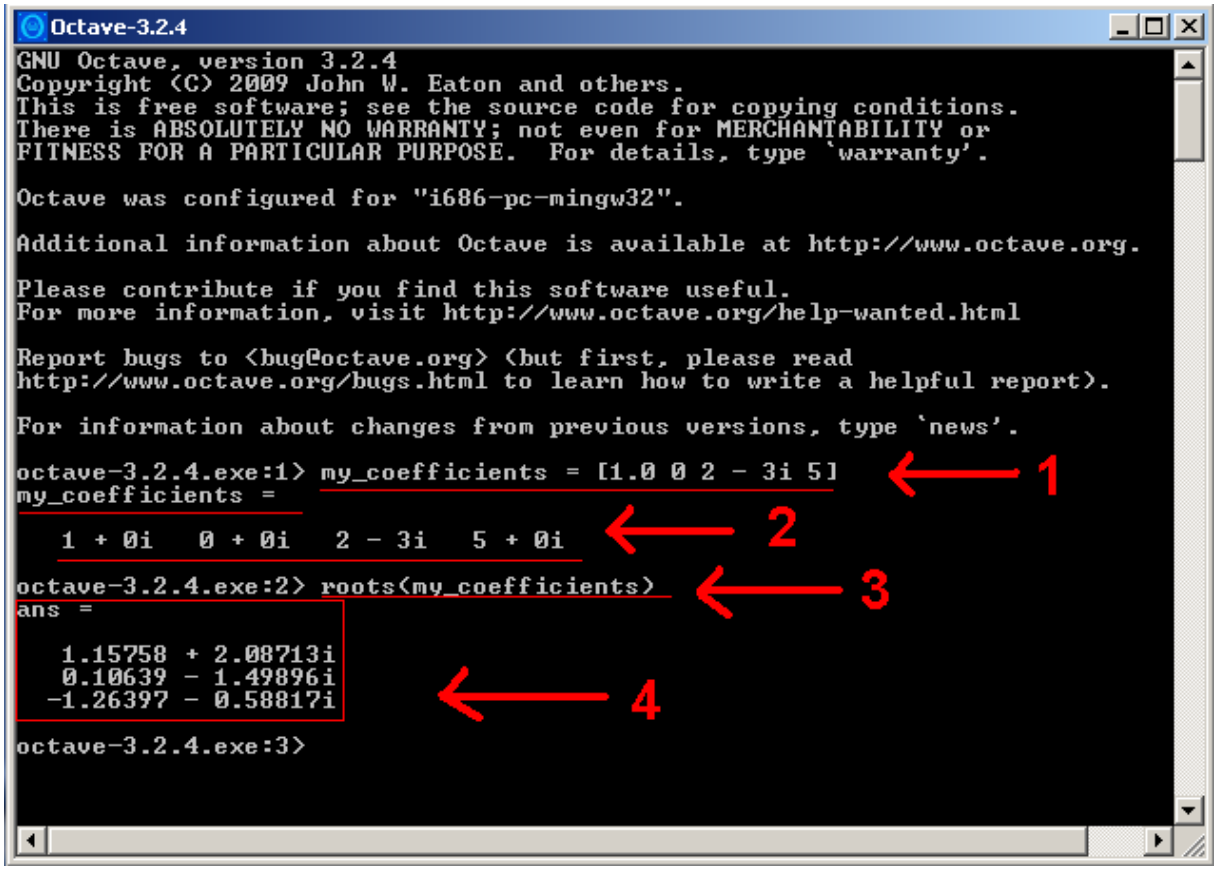

Figura 27: Exemplo de como calcular raízes polinomiais usando o Octave. Os comandos executados estão marcados com os números 1 e 3. As respostas do Octave estão associadas aos números 2 e 4.

Os comandos ilustrados na Figura 27 devem ser formados em Java e enviados ao Octave para que ele compute o resultado. Esse processo de integração Java-Octave será explicado com mais detalhes no capítulo a seguir, que descreve as principais classes usadas no trabalho.

3.6. DESCRIÇÃO DAS CLASSES

Usaremos os diagramas de classes da UML para explicar e ilustrar as principais classes do software desenvolvido. Nessa notação, por exemplo, o vermelho representa o modificador de acesso privado, o verde significa o modificador de acesso público, a letra ‘C’ nos métodos identifica construtores, a letra ‘S’ significa estático, ‘F’ significa final, entre outros símbolos usados na notação.

A classe ComplexNumber (Figura 28) representa um número complexo usando o tipo primitivo de dado *float* tanto para a parte real, quanto para a parte imaginária deste número. Esta classe também implementa as operações aritméticas entre números complexos. A primeira versão implementada do software utilizava a classe BigDecimal, disponibilizada pela *Sun* (empresa criadora do Java), para computar essas operações aritméticas. O BigDecimal representa um número real com precisão arbitrária e implementa operações aritméticas com estes números. Com isso, é possível exigir que a operação seja realizada com uma determinada precisão e se for impossível que isso aconteça (dízima periódica infinita, por exemplo), a operação não é executada. Assim sendo, é possível garantir que uma resposta foi calculada usando a precisão desejada, seja ela qual for.

Para computar o desenho do polinômio $f(z) = z^4 – 1$, considerando 800 *pixels* de largura por 600 *pixels* de altura, limite de iterações do método de Newton igual a 15 e o plano complexo compreendido pelo intervalo [-2; 2] tanto na parte real quanto na imaginária, obtivemos os seguintes tempos de execução:

- **Usando *float*:** aproximadamente 1 segundo
- **Usando BigDecimal com precisão 32:** aproximadamente 180 segundos
- **Usando BigDecimal com precisão 8:** aproximadamente 60 segundos

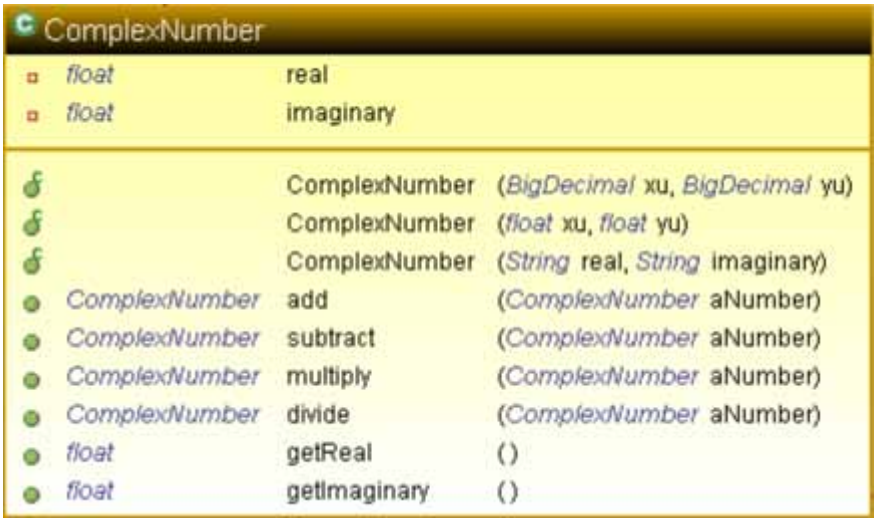

Figura 28: Atributos e métodos da classe ComplexNumber

Além de ponderar o tempo de execução, consideramos também que só era perceptível uma degradação na qualidade da imagem quando a precisão dos cálculos realizados com o BigDecimal era inferior a 4. Por outro lado, diminuir a precisão dos cálculos com BigDecimal aumenta o desempenho, mas ainda assim o desempenho obtido pelo *float* continua superior. Portanto, por esses motivos, a aritmética de *float* foi utilizada ao invés de BigDecimal e foi usado o Octave para a computação numérica das raízes polinomiais.

A classe que representa uma função polinomial é ilustrada pela Figura 29. Essa classe representa os polinômios usando um vetor (*array*) de ComplexNumber, onde cada posição desse vetor é um coeficiente do polinômio. Os coeficientes, neste vetor, são ordenados pelo grau, por exemplo, coefficients[0] é a constante e o coefficients[n] é o coeficiente de x^n, onde *n* é o grau do polinômio. Essa classe também possui métodos para derivar o polinômio e avaliar o valor da função em um dado valor, usando o método de Horner, a forma Fatorada e o método da Potência. Para desenvolver a classe PolynomialFunction, estudamos alguns *frameworks* e bibliotecas matemáticas que foram encontrados ao longo desta pesquisa. A biblioteca chamada Apache Commons Math [12] possui um conjunto de métodos para funções polinomiais reais que foram adaptados para os polinômios complexos.

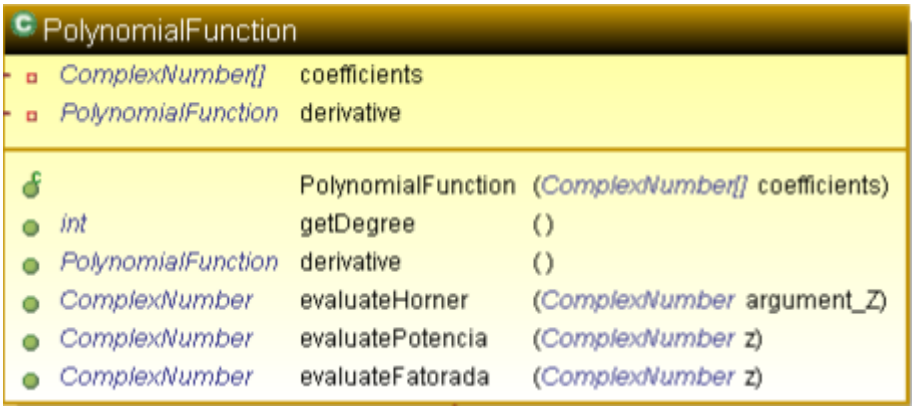

Figura 29: Atributos e métodos da classe PolynomialFunction

A classe responsável por representar os limites do plano complexo é chamada de PlaneLimits (Figura 30). Seu objetivo é informar os valores do plano complexo onde o Newton Fractal será desenhado.

Figura 30: Atributos e métodos da classe PlaneLimits

A classe que mantém informações sobre uma raiz de um polinômio é chamada de RootInformation (Figura 31). Cada raiz possui informações sobre qual seu valor, que é um número complexo, com qual cor está associada, qual a multiplicidade e qual a tolerância. A tolerância é usada para decidir o quão perto o método de Newton deve ir para concluir que houve convergência para aquela raiz.

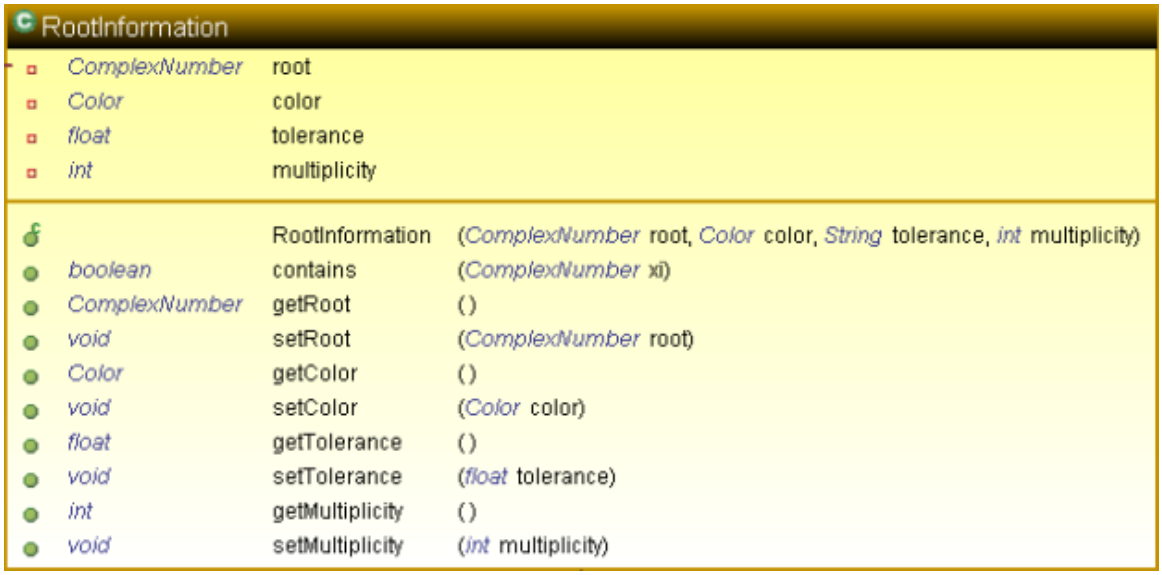

Figura 31: Atributos e métodos da classe RootInformation

Para calcular qual cor será usada em um dado valor z0 usado como estimativa inicial, foi criado um método na classe NewtonSolver que recebe como parâmetros:

- a função polinomial (PolynomialFunction),
- o numero complexo representando o valor inicial,
- o valor limite de iterações,
- informações sobre as raízes do polinômio,
- a forma de avaliação polinomial,
- e se deve ser aplicado transparência ou escurecimento na cor resultante.

O resultado (retorno) desse método é a cor que será usada no polinômio em questão, com o valor inicial passado por parâmetro.

A classe Jopas (Figura 32) executa o Octave, criando um novo processo. Para isso, é necessário conhecer o caminho do arquivo executável do Octave e possuir acesso ao sistema de arquivos. Neste caso específico, a classe Jopas recebe o JPanelRoots que contém o JTable onde as respostas do cálculo efetuado pelo Octave devem ser colocadas. Também é importante salientar que o método *Execute(String code)* da classe Jopas recebe um comando que será enviado à entrada padrão do Octave para ser executado.

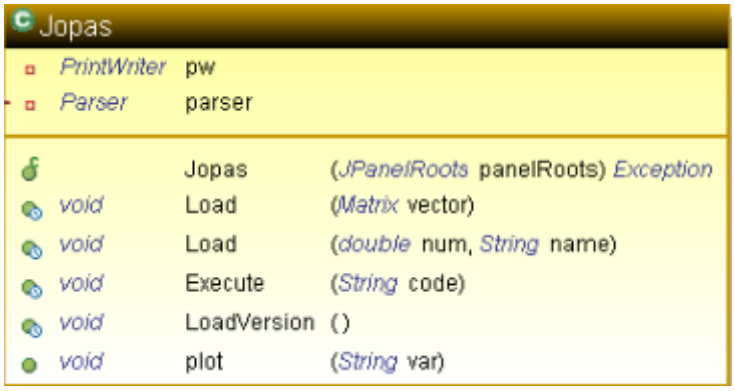

Figura 32: Atributos e métodos da classe Jopas

A classe Jopas (Figura 32) também cria e coloca em execução uma *thread* da classe Parser (Figura 33), responsável por ficar lendo a saída padrão do Octave, interpretando as respostas lá exibidas e extraindo os respectivos valores.

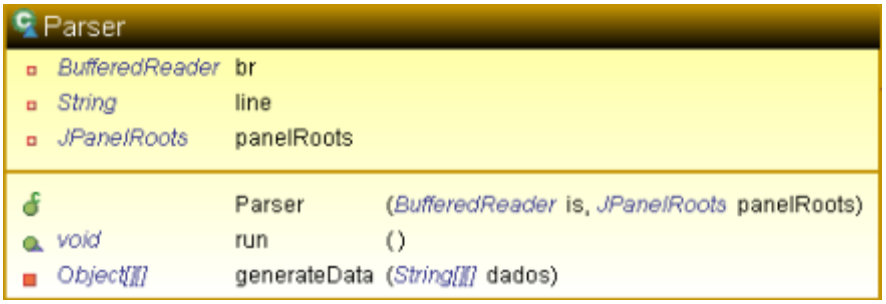

Figura 33: Atributos e métodos da classe Parser

Para auxiliar na parte gráfica, foram utilizadas as classes do pacote **javax.swing**. São exemplos de classes deste pacote: JButton, JTextField, JPanel e outros elementos gráficos. As seguintes classes fazem parte da interface gráfica do software:

JPanelCoefficients → Classe que extende o JPanel. É o painel exibido ao usuário quando a aba “Coeficientes” é selecionada. Acrescentamos nesse painel um botão que faz um sorteio de coeficientes, de forma pseudo-aletória, o que talvez possa facilitar a exploração de diferentes fractais.

JPanelRoots → Classe que extende o JPanel. É o painel exibido ao usuário quando a aba “Raízes” é selecionada. Acrescentamos nesse painel um botão para

sortear, de forma pseudo-aletória, as cores de cada raiz do polinômio, para facilitar a entrada de parâmetros do desenho.

JPanelExtra → Classe que extende o JPanel. É o painel exibido ao usuário quando a aba “Parâmetros Extras” é selecionada.

JPanelFractal → Classe que extende o JPanel. É responsável também por computar as cores do Newton Fractal e desenhar o resultado na tela. Essa computação das cores é realizada na classe interna *ThreadComputaCores* (Figura 34), que calcula e exibe uma barra de progresso informando a porcentagem do cálculo já realizado. Como o cálculo do Newton Fractal pode levar alguns segundos, é interessante que o usuário tenha uma noção de quanto falta de processamento. Ou seja, enquanto a computação vai sendo feita, o usuário pode ver a porcentagem do processamento que já foi computada.

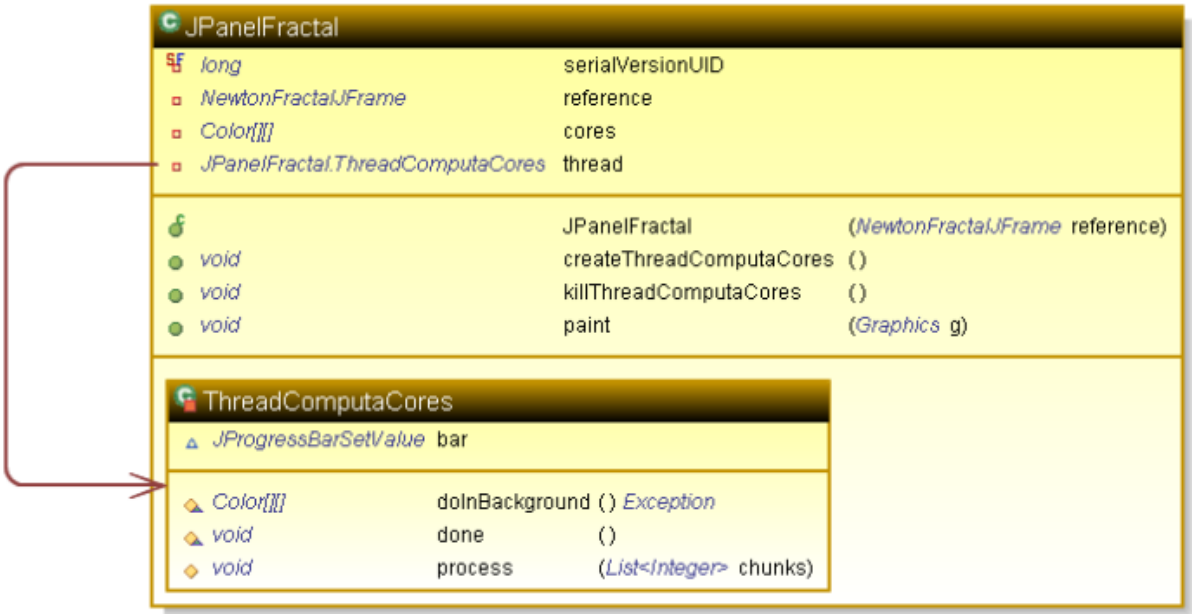

Figura 34: Atributos e métodos da classe JPanelFractal

ColorEditor → Classe que possibilita editar as cores das raízes na tabela de raízes com um clique do mouse, de forma que seja aberto uma janela para escolha da cor.

ColorRenderer → Classe responsável por desenhar (renderizar) uma cor nas células de uma tabela.

NewtonFractalJFrame → Esta classe é a janela (JFrame) principal do software, que contém as abas com os outros painéis (JPanel) e a os itens de menu. Essa classe também é a classe principal (main), ou seja, é a classe que deve ser executada primeiro (Figura 35).

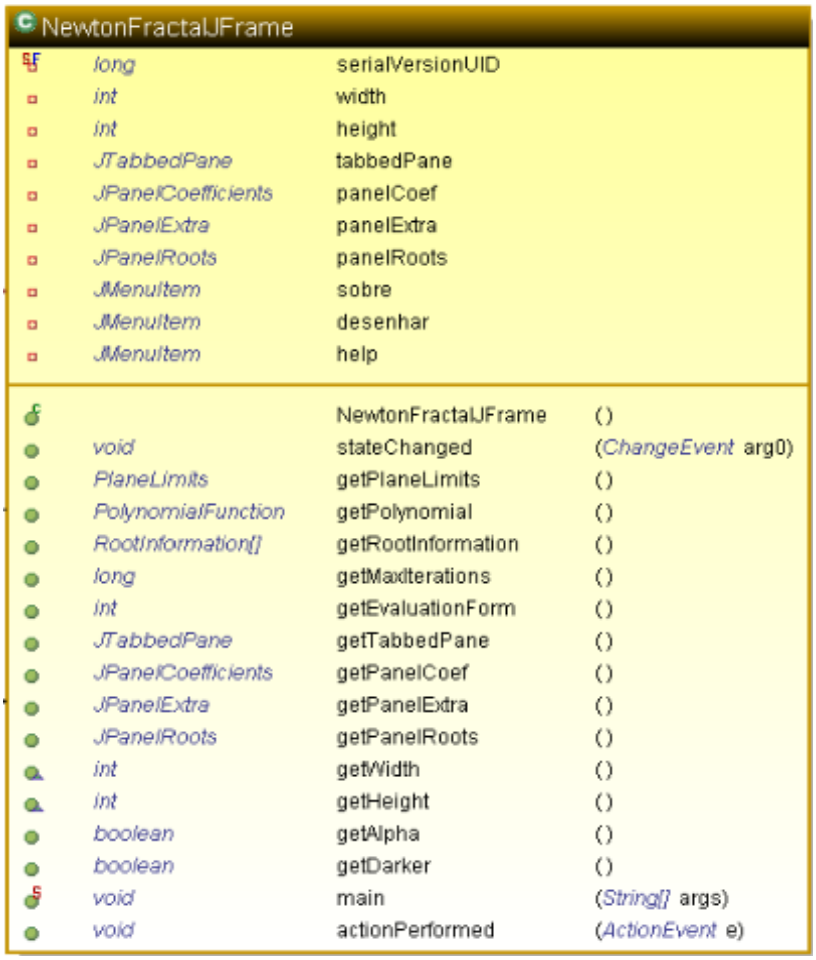

Figura 35: Atributos e métodos da classe NewtonFractalJFrame

A imagem ilustrada pela Figura 36 apresenta as dependências entre as classes do software e além disso, agrupa as classes em três grupos distintos:

- Classes focadas em resolver problemas matemáticos;
- Classes responsáveis pela integração do Octave com Java;
- Classes responsáveis pela interface gráfica.

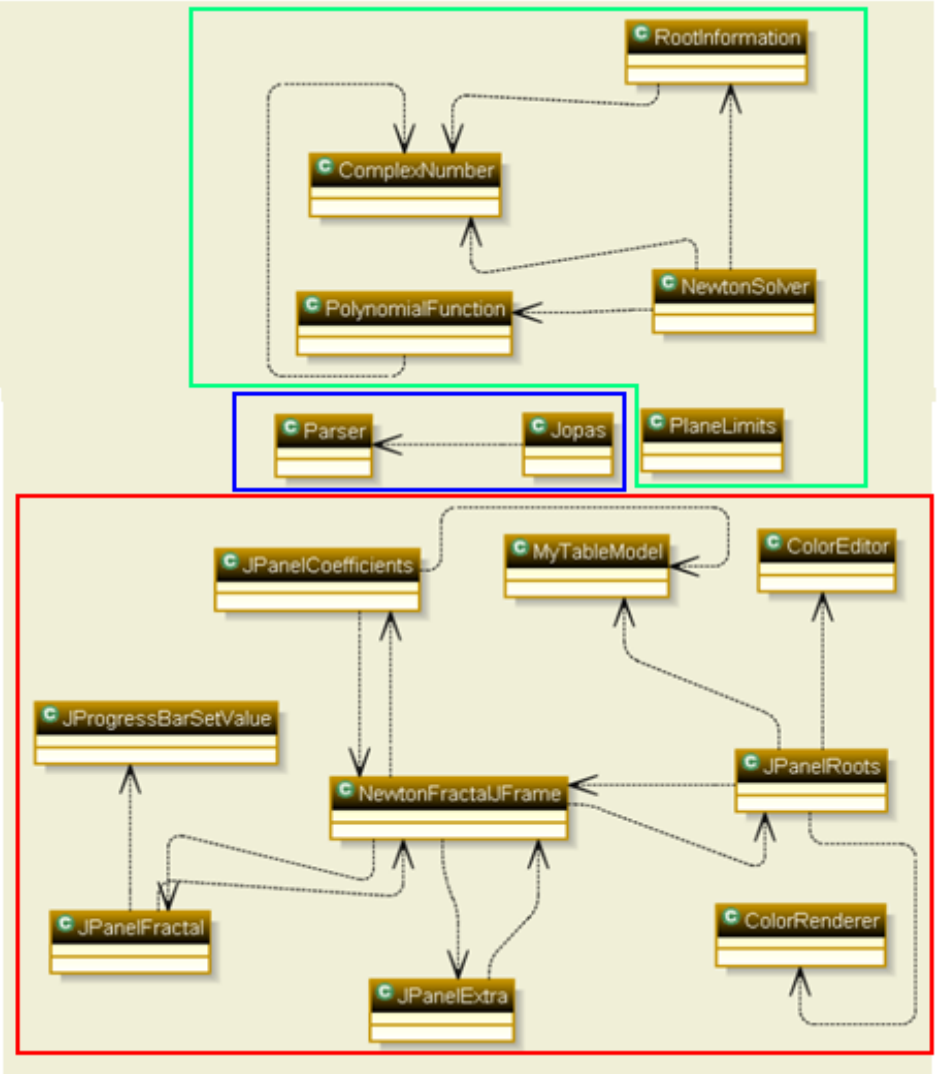

Figura 36: Diagrama de classe com as dependências entre as classes do projeto. A área verde contém as classes que resolvem os problemas matemáticos; a área azul possui as classes que executam comandos Octave em Java e em vermelho as classes responsáveis pela interface gráfica.

3.7. DIAGRAMA DE ATIVIDADES

Neste trabalho, diagramas de atividade da UML são utilizados para facilitar o entendimento das atividades necessárias para encontrar raízes polinômiais (Figura 37) e para desenhar o Newton Fractal (Figura 38). No software desenvolvido, o Octave realiza o cálculo das raízes da equação, enquanto que a computação do Newton Fractal está implementada em Java.

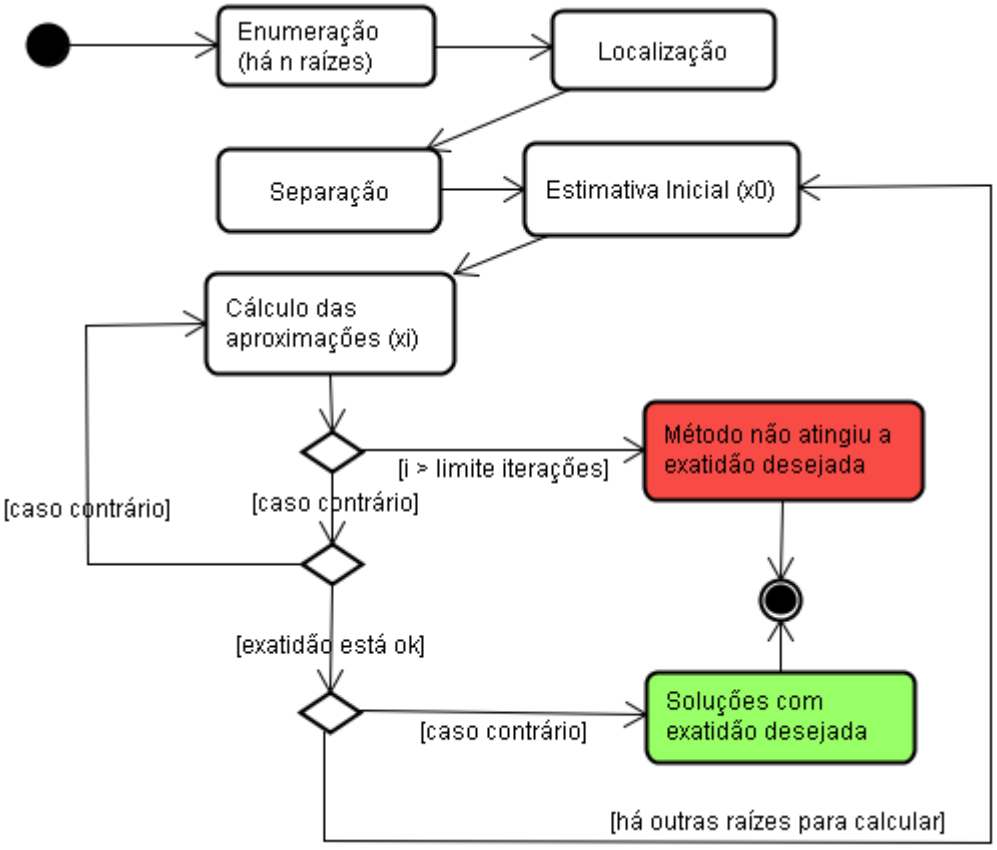

Figura 37: Diagrama de atividades do método para encontrar raízes de polinômios

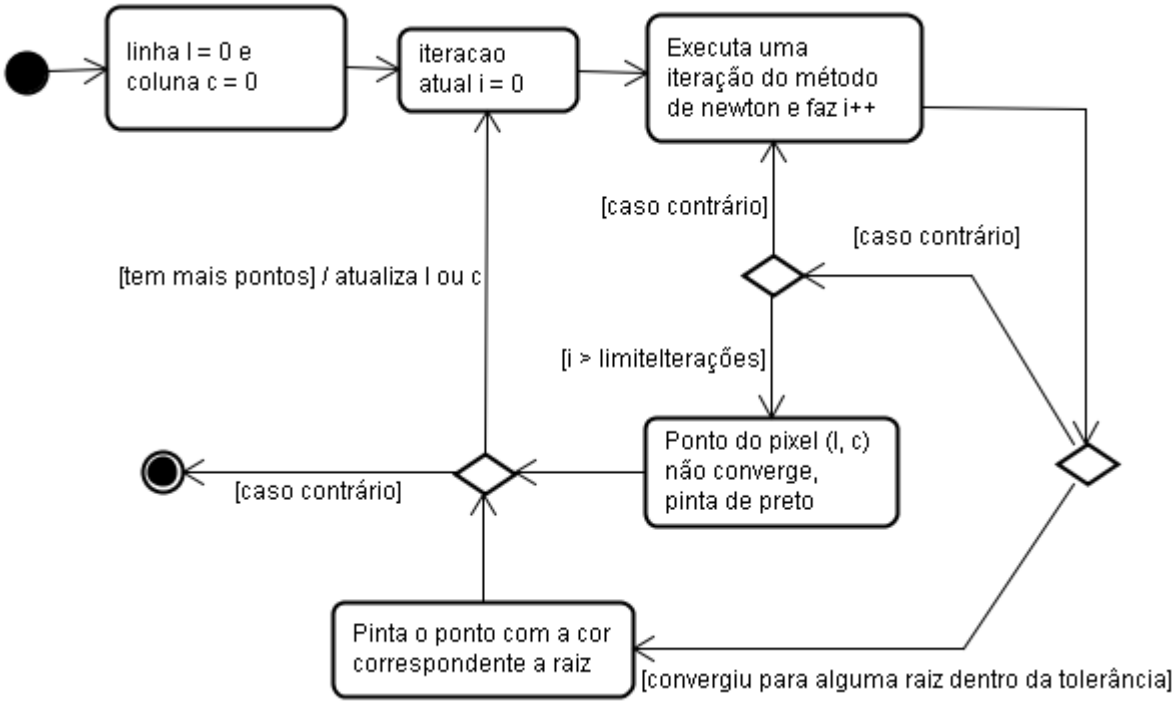

Figura 38: Diagrama de atividades do método de desenho do Newton Fractal

3.8. TESTES REALIZADOS

As imagens apresentadas a seguir (da Figura 39 até a Figura 43) ilustram fractais obtidas a partir da utilização do software desenvolvido neste trabalho, sendo que as duas primeiras possuem coeficientes imaginários e poderiam se enquadrar em uma categoria diferente das três últimas, que possuem apenas coeficientes reais:

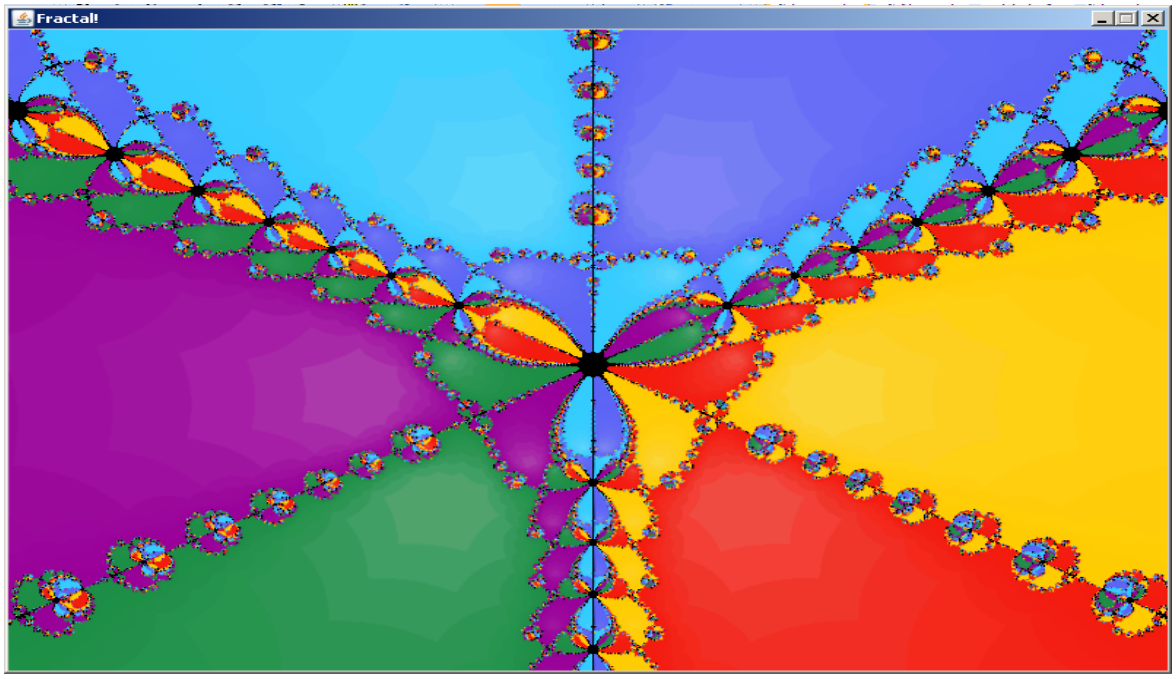

Figura 39: Polinômio: $f(z) = z^6 + iz^3 - 1$

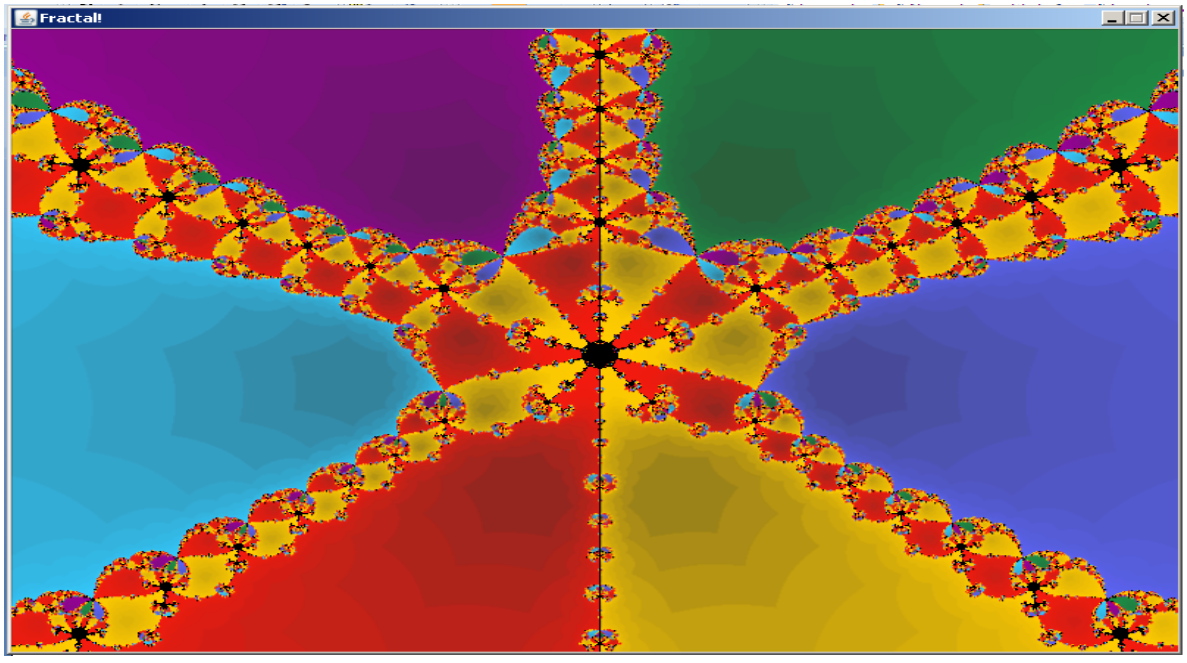

Figura 40: Polinômio: $f(z) = z^6 + iz - 1$

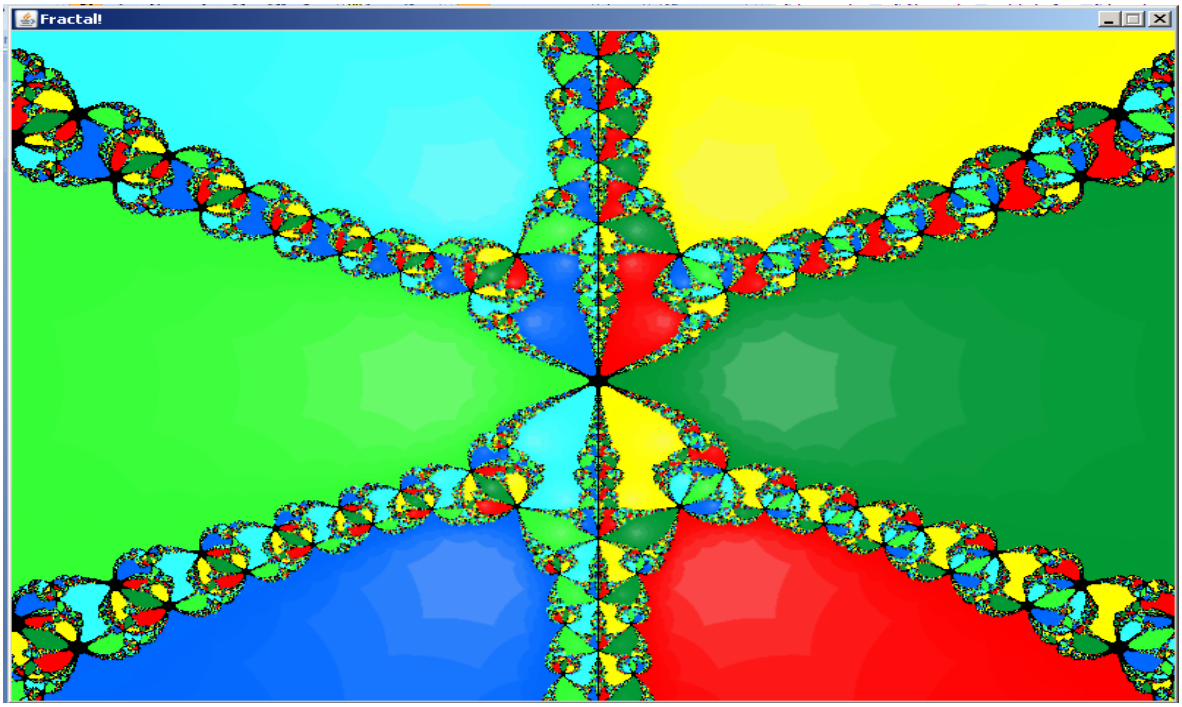

Figura 41: Polinômio: $f(z) = z^6 + z^4 + z^2 - 1$

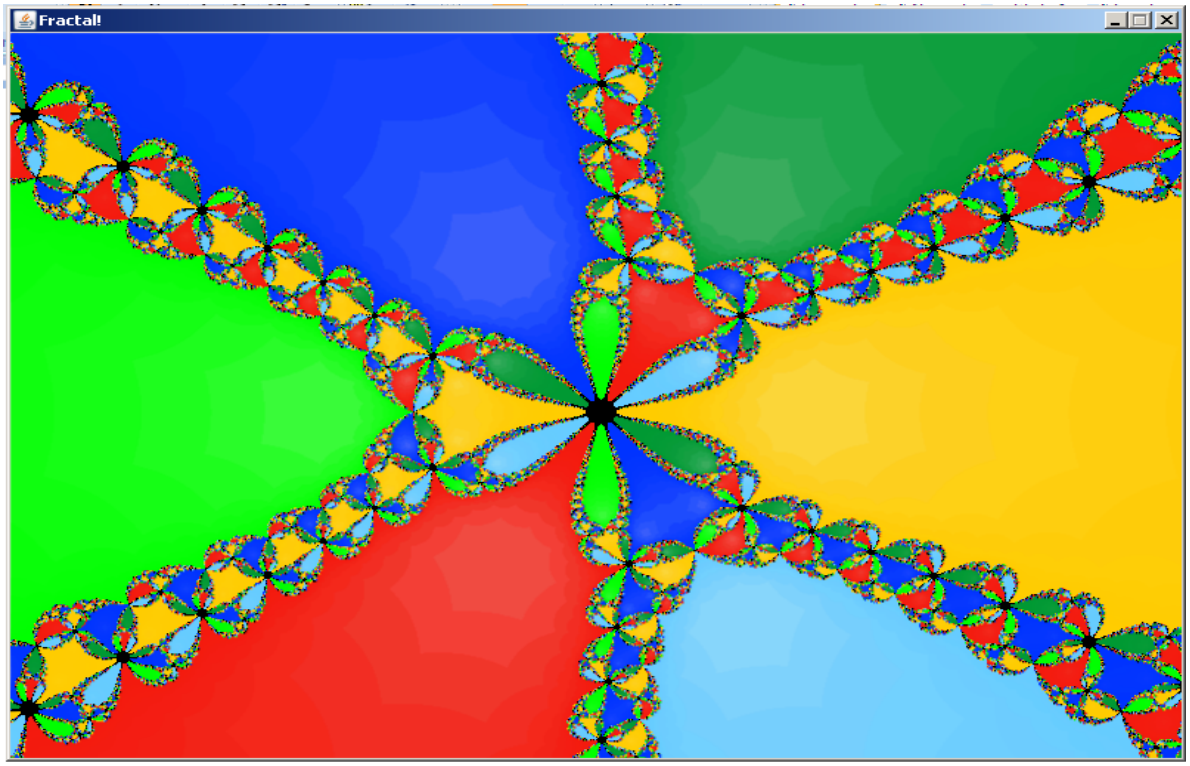

Figura 42: Polinômio: $f(z) = z^6 + z^3 - 1$

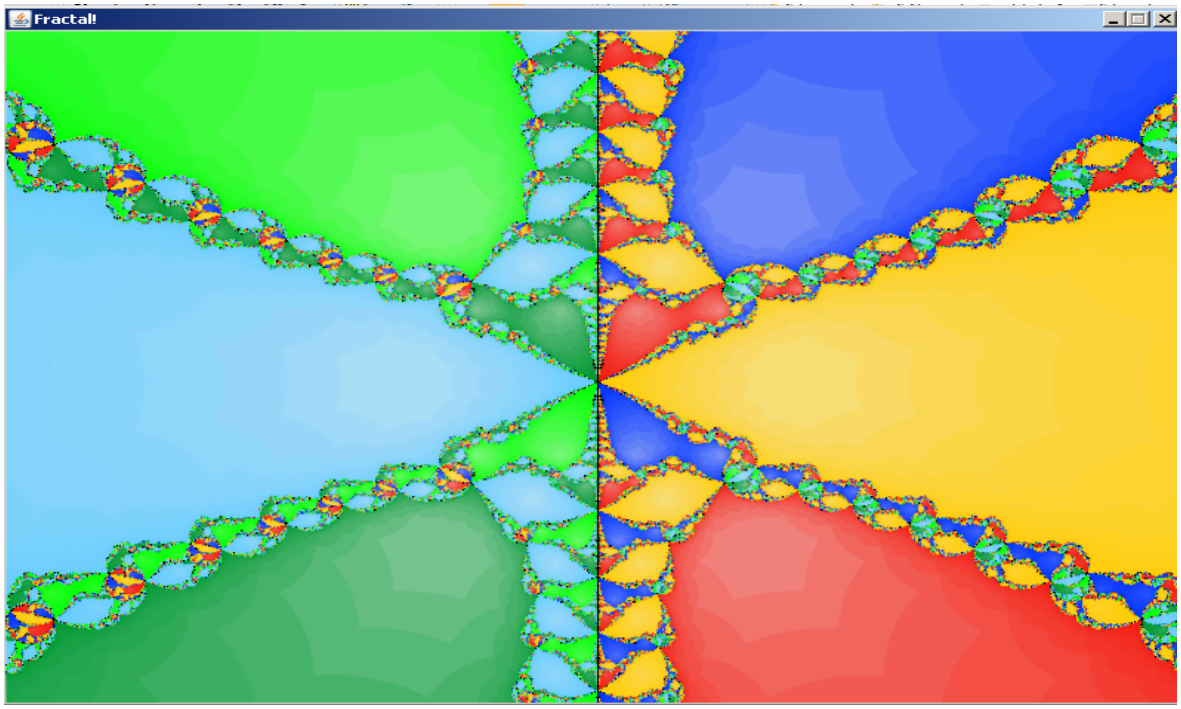

Figura 43: Polinômio: $f(z) = z^6 + z^2 - 1$

3.9. INFRA-ESTRUTURA DE HARDWARE E SOFTWARE

A infraestrutura de hardware para o desenvolvimento do software consiste em:

- Processador de 1.0 GHz ou superior.
- 1GB de memória RAM ou superior.
- Demais periféricos, como monitor, teclado, mouse, disco rígido, placa de rede, placa de vídeo.

Os recursos de software necessários para desenvolvimento e execução do software bem como para a edição do texto do trabalho foram os seguintes:

- Um ambiente de desenvolvimento Java (exemplo: Eclipse).
- Um navegador web (exemplo: Google Chrome).
- Um editor de texto (exemplo: Microsoft Word 2007).
- Um sistema operacional (exemplo: Linux, Windows, ou outro).

4. ATIVIDADES REALIZADAS NO DESENVOLVIMENTO DO TRABALHO

O sistema implementado materializa o trabalho desenvolvimento no período de um ano (TC1 e TC2) e cujas etapas estão listadas a seguir:

A1. Escolha da área temática.

A2. Definição do problema.

A3. Seleção preliminar de referências bibliográficas utilizadas na resolução do problema.

A4. Definição dos objetivos gerais e específicos.

A5. Identificação dos recursos de hardware e software.

A6. Redação da proposta de TC1.

A7. Estudo da linguagem de programação Java e levantamento de bibliotecas numéricas e gráficas existentes.

A8. Revisão bibliográfica para compor o referencial teórico do trabalho de TC1.

A9. Redação da fundamentação teórica sobre números complexos, métodos numéricos e teoria dos fractais.

A10. Definição da modelagem computacional da resolução do problema: levantamento de requisitos, descrição do sistema, diagrama de casos de uso, diagrama de classes, diagrama de atividades, projeto de interface.

A11. Levantamento dos softwares visualizadores de fractais.

A12. Redação da documentação final do TC1.

A13. Implementação da interface gráfica

A14. Pesquisa e implementação da integração Java-Octave

A15. Implementação do cálculo e desenho do Newton Fractal

A16. Otimizações no software

A17. Execução do software para obter desenhos e compará-los

A18. Redação da documentação final de TC2

A19. Revisão e correções do texto com a orientadora

A20. Preparação da apresentação do TC2

Meses / Atividades	Março	Abril	Maio	Junho	Julho
A1. Escolha da área temática	✔				
A2. Definição do problema	✔				
A3. Seleção bibliográfica preliminar	✔	✔	✔		
A4. Definição dos objetivos	✔				
A5. Identificação dos recursos de hardware e software		✔			
A6. Redação da proposta de TC1	✔	✔			
A7. Estudo da linguagem Java			✔	✔	✔
A8. Revisão bibliográfica para compor o referencial teórico		✔	✔	✔	✔
A9. Redação da fundamentação teórica	✔	✔	✔		
A10. Definição da modelagem computacional		✔			
A11. Levantamento dos softwares visualizadores de fractais			✔	✔	
A12. Redação da documentação final de TC1			✔	✔	✔

Quadro 3: Atividades realizadas no TC1 no período de março a julho de 2010.

Meses / Atividades	Ago.	Set.	Out.	Nov.	Dez.
A13. Implementação da interface gráfica	✔	✔			
A14. Pesquisa e implementação da integração Java-Octave			✔		
A15. Implementação do cálculo e desenho do Newton Fractal		✔	✔		
A16. Otimizações no software			✔	✔	
A17. Execução do software para obter desenhos e compará-los				✔	
A18. Redação da documentação final de TC2			✔	✔	
A19. Revisão e correções do texto com a orientadora				✔	
A20. Preparação da apresentação do TC2		✔			✔

Quadro 4: Atividades realizadas no TC2 no período de agosto a dezembro de 2010.

5. RESULTADOS

“Em algum lugar, algo incrível está esperando para ser conhecido.” Carl Sagan

Começamos com o Newton Fractal da função $f_1(z) = z^4 - 1$, que é uma imagem comum de ser encontrada quando se pesquisa por Newton Fractal (Figura 44).

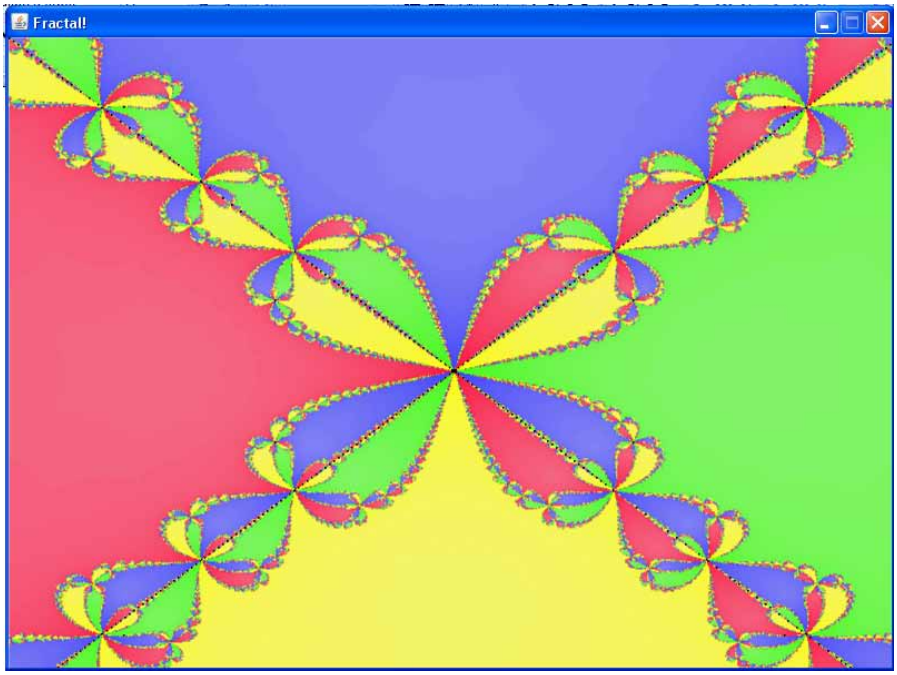

Figura 44: Newton Fractal do polinômio $f_1(z) = z^4 - 1$

Apenas invertendo o valor do coeficiente do termo independe, ou seja, a função $f_1(z) = z^4 - 1$ é substituída por $f_2(z) = z^4 + 1$, obtém-se um gráfico que parece ser a imagem anterior com uma rotação aplicada (Figura 45).

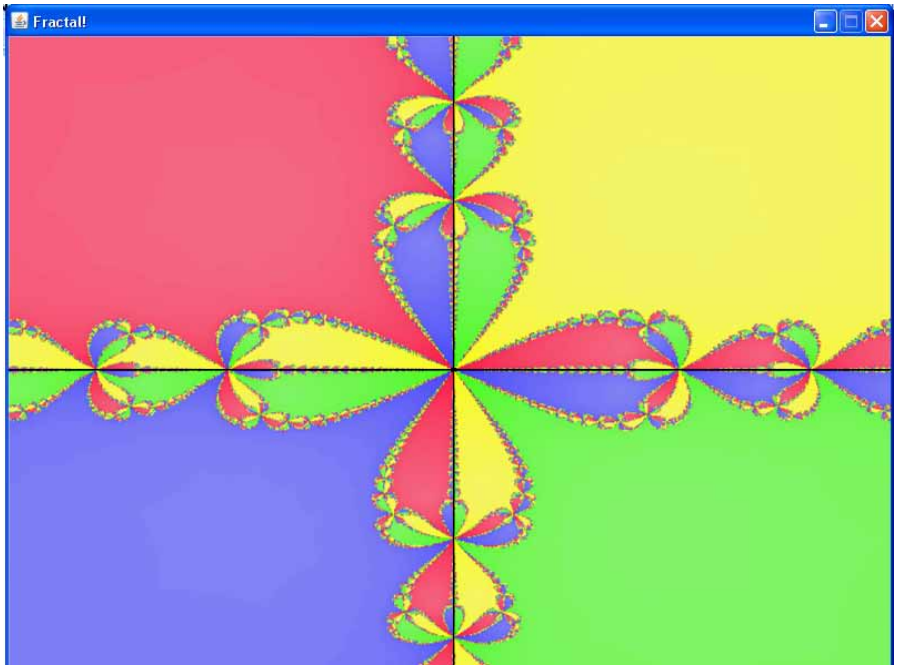

Figura 45: Newton Fractal do polinômio $f_2(z) = z^4 + 1$

Agora, inserindo o termo z^2 em f_2, obtém-se $f_3(z) = z^4 + z^2 + 1$ e, nesse caso, parece que "perturbamos" o gráfico (Figura 46).

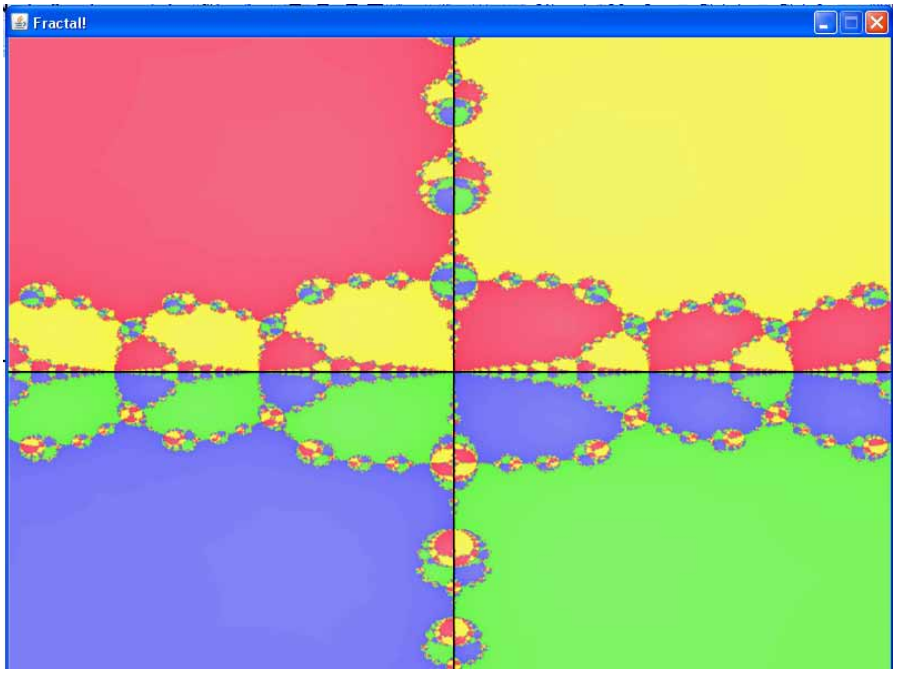

Figura 46: Newton Fractal do polinômio $f_3(z) = z^4 + z^2 + 1$

As imagens fractais pareciam ser bem estruturadas, até realiza a exploração visual do polinômio $f_4(z) = z^4 - z^2 + 5$ que possui um fractal com visual distorcido (Figura 47).

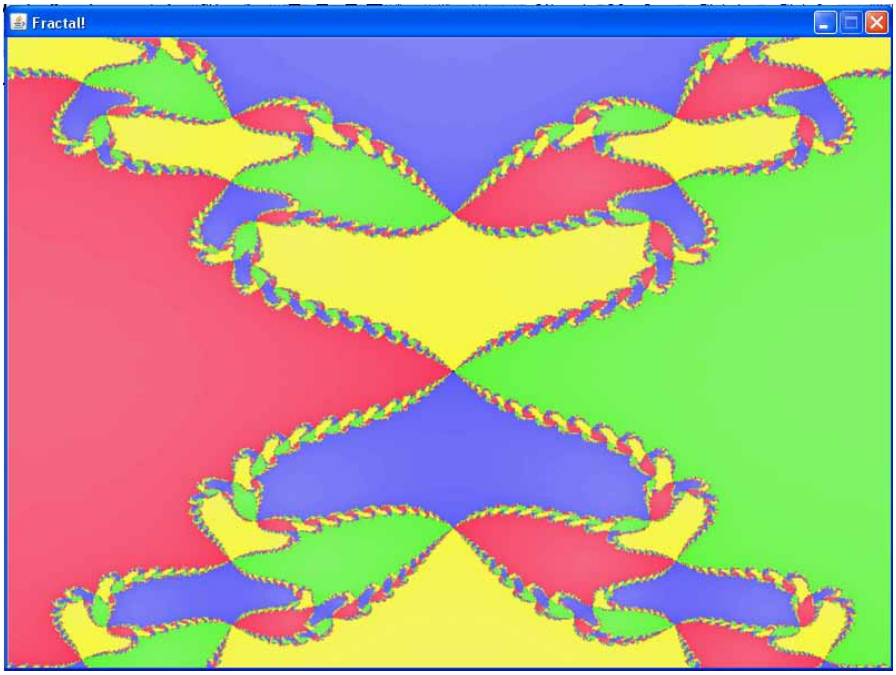

Figura 47: Newton Fractal do polinômio $f_4(z) = z^4 - z^2 + 5$

Na imagem a seguir (Figura 48) é importante observar a simetria entre as duas imagens da esquerda, pois aparentam ser o mesmo gráfico com uma transformação de rotação. O polinômio associado a imagem de cima da esquerda é $f_5(z) = -z^4 + z^2 + 1$ e o polinômio imediatamente abaixo é $f_6(z) = z^4 + z^2 - 1$. Já as duas imagens da direita possuem coeficientes complexos e parece que o gráfico de uma é igual ao da outra de cabeça para baixo. O polinômio associado à imagem de cima da direita é $f_7(z) = z^4 + iz^3 + z^2 + iz + 1$ e o polinômio da imagem de baixo é $f_8(z) = z^4 - iz^3 + z^2 - iz + 1$.

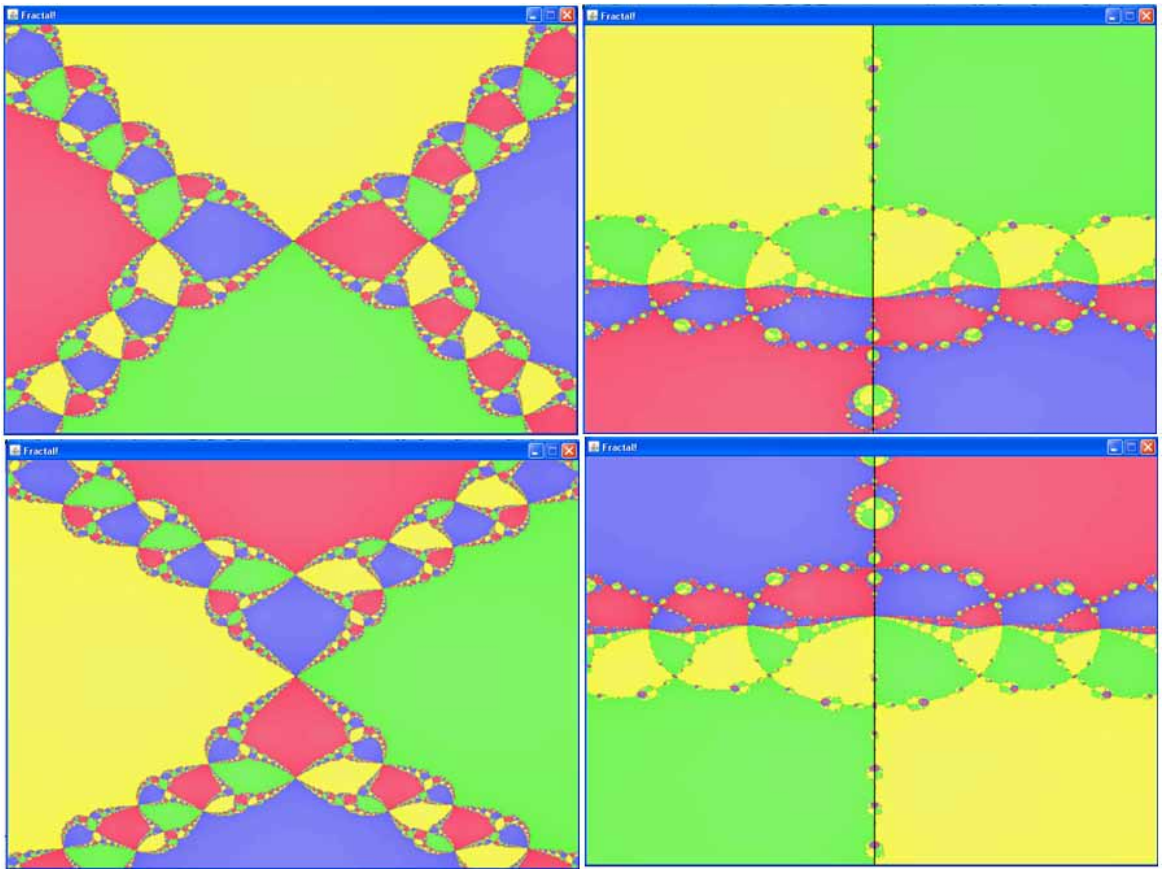

Figura 48: Fractais de quatro polinômios diferentes que aparentam possuir semelhança e simetria

No plano cartesiano o eixo horizontal é o eixo das abscissas e o eixo vertical é o eixo das ordenadas. Quando associamos uma função ao plano cartesiano, obtemos pontos discretos ou contínuos no plano que representam os valores da função. Por outro lado, no plano complexo, o eixo horizontal representa a parte real de um número complexo e o eixo vertical representa sua parte imaginária. Para associar uma função no plano complexo usamos cores, mas o significado desses "formatos" ainda está em aberto. A pergunta "Dado um z0 qualquer para qual raiz ele converge?" é respondida na imagem do Newton Fractal. Mas outra pergunta aparece: assim como existe explicação sobre as formas dos gráficos no plano cartesiano, como retas, parábolas, e etc, qual será a explicação para o **formato** desses gráficos no plano complexo?

6. CONSIDERAÇÕES FINAIS

Uma utilidade para o Newton Fractal é facilitar a visualização do comportamento do algoritmo numérico do método de Newton para o cálculo de raízes. Além de possibilitar um maior entendimento do método de Newton, ele pode ser usado no ensino de métodos numéricos. Dessa forma, é possível despertar a curiosidade e chamar a atenção dos alunos com uma imagem original e colorida gerada simplesmente por uma equação matemática.

Além disso, o Newton Fractal pode ser classificado como arte fractal, onde funções matemáticas transformam o resultado do cálculo em mídias que podem ser imagens, animações ou música. Podem ser estudadas melhores formas de visualizar a imagem fractal, usando transparência, brilho, cores, sombras, bordas, entre outros.

Um estudo sistemático poderia ser realizado, onde fosse preparado um conjunto de funções de testes. Neste caso, as imagens poderiam ser comparadas na tentativa de encontrar alguma similaridade ou relação entre os gráficos de diferentes funções. Além disso, a precisão da máquina e a forma de avaliação polinomial poderiam ser alteradas para tentar encontrar imagens diferentes usando a mesma equação.

A escolha do *Octave* para a computação matemática foi fundamentada no fato de que o *Octave* é gratuito ao contrário do *MatLab* e o *Mathematica*. Além disso, o *Octave* é *open source* e possui a mesma sintaxe do *MatLab*. O *Maxima* é uma opção que também é gratuita e *open source*, que poderia ser semelhantemente usada para computação matemática. Uma diferença entre ambos é que o *Maxima* foi planejado para computação simbólica, enquanto que o *Octave* foi projetado inicialmente para computação numérica. Não implementamos a integração do Java com o *Maxima*, mas acreditamos que seja viável também.

Uma versão inicial do software desenvolvido computava a aritmética usando a classe *BigDecimal* da API do Java. Com o *BigDecimal*, era possível fazer cálculos usando uma precisão arbitrária, no entanto, o desempenho não ficou satisfatório. Simplesmente trocando o *BigDecimal* pelo tipo de dado *float*, obtivemos uma aceleração considerável na computação do desenho fractal.

Outra otimização possível, que seria semelhante a programação dinâmica, funcionaria da seguinte forma: para cada ponto seria calculado a sequência do

método de Newton e todos os pontos do desenho pertencentes a sequência receberiam a cor correspondente a raiz que o último ponto da sequência convergiu. Dessa forma, espera-se diminuir o esforço computacional usado para calcular a convergência de valores que já foram calculados anteriormente. Caso essa versão fosse implementada, poderiam ser feitas comparações entre os desenhos da versão original com a versão otimizada para identificar se haveriam diferenças.

7. REFERÊNCIAS

[1] MANDELBROT, Benoit. **The fractal geometry of nature.** New York: W. H. Freeman, 1982.

[2] WEISSTEIN, Eric W. **Fractal.** From MathWorld – A Wolfram Web Resource. Disponível em <http://mathworld.wolfram.com/Fractal.html>. Acesso em 23 de março de 2010.

[3] WEISSTEIN, Eric W. **Newton's Method.** From MathWorld – A Wolfram Web Resource. Disponível em <http://mathworld.wolfram.com/NewtonsMethod.html>. Acesso em 22 de março de 2010.

[4] CLÁUDIO, Dalcídio Moraes; KOLBERG, Mariana Luderitz; BOCIAN, Daniel. **Ferramenta para avaliação gráfica de polinômios.** Faculdade de Informática, Pontifícia Universidade Católica do Rio Grande do Sul, 2001. Disponível em <http://www.inf.pucrs.br/~dalcidio/disciplinas/software_cientifico/Manual_Polinomios.pdf>. Acesso em 26 de março de 2010.

[5] CARREIRA, Ana Sofia Nunes; ANDRADE, Carlos António Dias. **Geometria a várias dimensões – Fractais?** Faculdade de Ciências, Universidade de Lisboa, 1999. Disponível em <http://www.educ.fc.ul.pt/icm/icm99/icm43/fractais.htm>. Acesso em 25 de março de 2010.

[6] MCCLURE, Mark. **Fractals from Newton's Method.** Asheville, departamento de matemática da Universidade da Carolina do Norte. Disponível em <http://facstaff.unca.edu/mcmcclur/mathematicaGraphics/Newton/>. Acesso em 25 de março de 2010.

[7] ALVES, Célia Maria Filipe Santos Jordão. **Fractais: conceitos básicos, representações gráficas e aplicações ao ensino não universitário.** Tese de mestrado, departamento de matemática, universidade de Lisboa, 2007.

[8] CLÁUDIO, Dalcídio Moraes; MARINS, Jussara Maria. **Cálculo Numérico Computacional – Teoria e Prática.** São Paulo: Atlas, 2000.

[9] WEISSTEIN, Eric W. **Complex Number.** From MathWorld – A Wolfram Web Resource. Disponível em <http://mathworld.wolfram.com/ComplexNumber.html>. Acesso em 31 de março de 2010.

[10] COURANT, Richard; ROBBINS, Herbert; STEWART, Ian. **What is mathematics? An elementary approach to ideas and methods.** Oxford: Oxford University Press US, 1996.

[11] TATHAM, Simon. **Fractals derived from Newton-Raphson iteration.** Disponível em <http://www.chiark.greenend.org.uk/~sgtatham/newton/>. Última modificação em 4 de abril de 2007. Acesso em 5 de abril de 2010.

[12] APACHE, Commons Math. **Numerical Analysis.** Disponível em <http://commons.apache.org/math/userguide/analysis.html>. Acesso em 13 de junho de 2010.

[13] ORACLE, The Java™ Tutorials. **What Applets Can and Cannot Do**. Disponível em <http://download.oracle.com/javase/tutorial/deployment/applet/security.html>. Acesso em 8 de novembro de 2010.

[14] LONG, P.J.G. **Introduction to Octave.** Department of Engineering, University of Cambridge, 2005.

8. BIBLIOGRAFIA

FALCONER, Kenneth. **Fractal Geometry: Mathematical Foundations and Applications.** Chicester: John Wiley & Sons, 1990.

BARNSLEY, Michael**. Fractals everywhere.** London: Academic Press Professional, 1993.

COSTA, Luciano da Fontoura; BIANCHI, Andrea Gomes Campos. **A outra dimensão da dimensão fractal.** Grupo de pesquisa em Visão Cibernética, Instituto de Física de São Carlos, Universidade de São Paulo. Disponível em <http://www.bioinfo.usp.br/Ciencia_Hoje_Jun02.pdf>. Acesso em 24 de março de 2010.

MANDELBROT, Benoit**. How long is the coast of Britain? Statistical self-similarity and fractional dimension.** Departamento de matemática, Universidade Yale, Estados Unidos. Versão online do artigo disponível em <http://www.math.yale.edu/~bbm3/web_pdfs/howLongIsTheCoastOfBritain.pdf>. Acesso em 25 de março de 2010.

WEISSTEIN, Eric W. **Julia Set.** From MathWorld – A Wolfram Web Resource. Disponível em <http://mathworld.wolfram.com/JuliaSet.html>. Acesso em 23 de março de 2010.

ADDISON, Paul. **Fractals and chaos – An illustrated course.** Institute of Physics Publish. London: CRC Press, 1997.

JOYCE, David E. **Newton Basins.** Clark University. Disponível em <http://aleph0.clarku.edu/~djoyce/newton/newton.html> Última modificação em agosto de 1994. Acesso em 12 de junho de 2010.

KORMANN, Dmitry. **Fractals, Chains and the Golden Ratio: Musical Applications.** Disponível em <http://bowerbird-studios.com/aicaramba/page2.html>. Acesso em 20 de maio de 2010.

FRAME, Michael; MANDELBROT, Benoit; NEGER, Nial. **Fractal Geometry.** Disponível em < http://classes.yale.edu/fractals/>. Yale University, última modificação em 23 de junho de 2010. Acesso em 24 de junho de 2010.

RICHLING, Mitch. **Newton's Method.** Disponível em <http://www.mitchr.me/SS/newton/>. Última modificação em 2009. Acesso em 18 de maio de 2010.