

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

NESTOR ZILIOOTTO SALAMON

Representação sonora da complexidade de algoritmos numéricos

Porto Alegre, Rio Grande do Sul, Brasil

9 de julho de 2012

NESTOR ZILIOOTTO SALAMON

*Representação sonora da complexidade de
algoritmos numéricos*

Relatório final da disciplina de Trabalho de
Conclusão de Curso II

Orientador:

Prof^a Beatriz Regina Tavares Franciosi

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL

Porto Alegre, Rio Grande do Sul, Brasil

9 de julho de 2012

Agradecimentos

Não somente através da construção teórica da solução do problema e o desenvolvimento de competências e habilidades referentes à computação, este trabalho foi possível devido ao apoio permanente de uma equipe de colaboradores especialistas: o orientador do trabalho que conduziu os estudos necessários sobre métodos numéricos e facilitou o contato com o autor do *NewtonFractal*; Artur Freitas que prestou pontual ajuda na compreensão e extensão do *software NewtonFractal* por ele desenvolvido; o professor João Batista que agregou, ao trabalho, seus conhecimentos sobre o desenvolvimento de algoritmos e ideias para tornar interessante e instigante a visualização de sequências de aproximações das raízes; o professor Tiago que trouxe sólido conhecimento musical, diferencial sem o qual este trabalho seria apenas mais um trabalho de codificação de algoritmos numéricos e Leonardo Heredia pelas dicas sobre estruturação do texto e modelagem UML.

Também agradeço a todos amigos e familiares que incentivaram, criticaram, duvidaram ou elogiaram este trabalho. Sem a ajuda de todos, as noites em claro não teriam trazido resultados.

Resumo

O estudo da complexidade de algoritmos acontece desde os primórdios da computação. Na década de 70, era necessário fazer o programa rodar em tempo hábil com alguns *kilobytes* de memória. Hoje em dia, a capacidade de processamento vem aumentando significativamente, de forma que os problemas mais simples são resolvidos e outros mais complexos surgem. Em se tratando de problemas matemáticos, muitas vezes a capacidade de representação numérica do computador não é suficiente para o armazenamento dos números obtidos no processo de solução. Assim, arredondamentos dos valores são necessários. Estes arredondamentos podem se acumular durante o processo de resolução e gerar uma grande diferença na resposta final ou, ainda, fazer com que o algoritmo não encontre uma resposta.

Com intuito de verificar como o comportamento do algoritmo, durante a resolução destes problemas matemáticos, relaciona-se com sua complexidade, é usado o método numérico de Newton para gerar um fractal, mostrando as bacias de convergência para as raízes de uma equação. Com esta imagem fractal, uma associação entre as suas características é feita com os conceitos da música, de forma a obter uma relação sonora do comportamento do algoritmo durante a resolução do problema. Este som resulta em uma representação de como os erros de arredondamento influenciam na convergência e, conseqüentemente, no tempo de execução do algoritmo e/ou consumo de recursos da máquina, ou seja, em uma representação da complexidade do algoritmo.

Palavras-chave: Métodos Numéricos. Método Newton. Fractais. Som. Música. Complexidade de algoritmos. Estabilidade do processo de solução.

Abstract

The study of computational complexity of algorithms has been following the computing since its beginning: in the 70s, it was necessary to make the program run in certain time with few kilobytes of memory. Today, the processing capacity has increased significantly, so the simpler problems are solved faster and more complex ones appear. Dealing with mathematical problems, so many times the capacity of the computer numerical representation is not enough to store the numbers obtained in the solution process. Therefore, values rounding are necessary. These roundings can be accumulated during the resolution process generating a big difference in the final answer, or even make the algorithm does not find an answer.

With the objective of seeing how the algorithm's behavior solving these mathematical problems is related to its complexity, the Newton's numerical method is used to generate a fractal showing the convergence basins to the roots of an equation. With this fractal image, an association between its characteristics is made with the concepts of music, in order to obtain a sound relationship of the algorithm's behavior while solving the problem. This sound results in a representation of how rounding errors influence in the convergency and, consequently, in the run time of the algorithm and/or consumption of machine resources or, in other words, a representation of the algorithm complexity.

Keywords: Numeric Methods. Newton's Method. Fractals. Sound. Music. Complexity of algorithms. Stability of the solution process.

Lista de Figuras

- 2.1 Atrator de Lorenz mostrando, através de um fractal, o comportamento de um sistema caótico. p. 17
- 2.2 Um raio é um belo exemplo de um fractal na natureza, pois sua trajetória não pode ser representada por uma reta da geometria euclidiana. p. 18
- 2.3 A folha de uma samambaia é um exemplo de fractal autossemelhante p. 18
- 2.4 O fractal de funções iteradas gerado pelo método numérico de Newton para o cálculo das raízes do polinômio $f(x) = x^3 - 1$ p. 19
- 2.5 O tom fundamental e os seis sobretons de uma corda vibrante: o estudo de Pitágoras. p. 20
- 2.6 A duração das notas em uma pauta. p. 23
- 2.7 O espectro das frequências que alguns instrumentos musicais e a voz conseguem atingir. p. 24
- 2.8 A relação harmoniosa das notas representada nas teclas do piano. p. 26
- 2.9 Forma de ondas de um *pink noise* mais agradável ao ouvido humano. p. 27
- 2.10 Forma de ondas de um *white noise*. Combinação de sons de todas frequências; nosso cérebro não é capaz de distingui-las. p. 28
- 2.11 A arquitetura da *MIDI audio* para sequenciamento e sintetização (reprodução) dos sons. p. 29
- 2.12 A relação das notas MIDI's com as notas de um piano acústico p. 30
- 3.1 A interface do programa desenvolvido por Artur Freitas p. 32
- 3.2 Um fractal gerado pelo *Newton Fractal* para o cálculo das raízes do polinômio $f(z) = z^5 + z^2 + z + 1$ p. 32
- 3.3 A interface do *Photosounder* quando carregada uma imagem do conjunto de Mandelbrot. p. 34

3.4	<i>AudioPaint</i> com um conjunto de Mandelbrot para ser utilizado na geração do som.	p. 35
3.5	A interface do <i>Mandelbrot Composer</i>	p. 36
4.1	Mapeamento de um ponto na tela (500 x 500) para um ponto no plano (2 x 2)	p. 40
4.2	Mapeamento das notas musicais no plano de números complexos.	p. 41
4.3	Mapeamento das notas musicais no plano com a opção <i>Allow Chords</i> ativa. .	p. 42
4.4	A hipotenusa AB representando a distância entre os pontos A e B	p. 43
4.5	A opção para utilizar o <i>FractalNoise</i> no <i>NewtonFractal</i> configurado para resolução de $f(z) = z^6 - 1$	p. 44
4.6	A tela do <i>FractalNoise</i> quando carregado o fractal correspondente à equação $f(z) = z^6 - 1$	p. 45
4.7	A tela do <i>FractalNoise</i> mostrando a divisão dos quadrantes com os quatro pontos pré-estabelecidos para análise, na área de desenho do fractal correspondente à equação $f(z) = z^3 - 2$	p. 48
4.8	O diagrama de casos de uso do <i>FractalNoise</i>	p. 50
4.9	As classes do <i>NewtonFractal</i> alteradas para utilização no <i>FractalNoise</i>	p. 51
4.10	As classes criadas para implementação do <i>FractalNoise</i>	p. 52
4.11	Tela do <i>FractalNoise</i> analisando por uma <i>Analyse 1</i> o fractal $f(z) = z^6 - 1$. .	p. 54
4.12	Tela do <i>FractalNoise</i> analisando o fractal da equação $f(z) = z^6 + z^3 - 1$. . .	p. 55
5.1	A tela do <i>FractalNoise</i> configurado para resolução de $f(z) = z^3 - 2$ nos quatro pontos iniciais destacados em regiões estáveis.	p. 57
5.2	O <i>FractalNoise</i> configurado para resolução de $f(z) = z^3 - 2$ nos quatro pontos iniciais destacados em regiões instáveis	p. 58
5.3	O <i>FractalNoise</i> configurado para resolução de $f(z) = z^3 - 2z^2 - 5z + 6$ através de uma análise <i>Analyse 1</i> : os pontos iniciais estão destacados.	p. 59
5.4	O <i>FractalNoise</i> configurado para resolução de $f(z) = z^6 + z^4 + z^2 - 1$ com uma análise <i>Solve Selected</i> para os pontos destacados.	p. 60
5.5	O <i>FractalNoise</i> configurado para resolução de $f(z) = z^6 + z^4 + 2z^2 - 1$ através de uma <i>Analyse 2</i>	p. 61

A.1	O prisma da experiência de Newton reproduzido: as cores da luz branca são refratadas pelo prisma.	p. 69
A.2	A divisão do espectro das cores visíveis de acordo com o comprimento da onda.	p. 70
A.3	As cores representadas nas teclas de um piano.	p. 71
B.1	A interface do protótipo quando carregada uma imagem do conjunto de Mandelbrot.	p. 73
B.2	As principais classes do protótipo implementado.	p. 74
C.1	Diagrama de atividades representando o fluxo da análise <i>Solve Step</i>	p. 76
C.2	Diagrama de atividades representando o fluxo da análise <i>Solve Selected</i>	p. 77
C.3	Diagrama de atividades representando o fluxo da análise <i>Solve Together</i>	p. 78
C.4	Diagrama de atividades representando o fluxo da <i>Analyse 1</i>	p. 79
C.5	Diagrama de atividades representando o fluxo da <i>Analyse 2</i>	p. 80
D.1	O mapa mental utilizado para escolha da área e orientação do estudo.	p. 82

Lista de Tabelas

2.1	As frequências (em <i>hertz</i>) das notas em cada oitava.	p. 23
2.2	Os números MIDI das notas em cada oitava.	p. 29
4.1	A bateria de teste utilizada durante o trabalho	p. 44
A.1	A variação (Δ) do comprimento de onda (λ) e a frequência (f) das cores no espectro de luz visível	p. 70
A.2	As notas musicais e suas cores.	p. 71

Sumário

1	Introdução	p. 11
1.1	Objetivos	p. 12
1.1.1	Objetivo geral	p. 12
1.1.2	Objetivos específicos	p. 12
2	Fundamentação teórica	p. 14
2.1	Algoritmos numéricos	p. 14
2.1.1	Método de Newton	p. 15
2.2	Teoria do caos	p. 16
2.3	Geometria fractal	p. 16
2.4	Som	p. 20
2.5	Música	p. 21
2.5.1	Notas musicais e suas frequências	p. 22
2.5.2	Periodicidade e a estabilidade da música	p. 24
2.5.3	Música fractal	p. 26
2.6	Padrão MIDI	p. 28
3	Prospecção da infraestrutura de <i>software</i>	p. 31
3.1	Geração de fractais	p. 31
3.2	Conversão de imagem em música	p. 33
3.2.1	Photosounder	p. 33
3.2.2	AudioPaint	p. 34

3.2.3	Mandelbrot Composer	p. 35
3.3	Considerações sobre as ferramentas analisadas	p. 36
4	Representação sonora do método de Newton	p. 38
4.1	Mapeamento tela x plano	p. 39
4.2	Mapeamento ponto do plano cartesiano x nota musical	p. 39
4.3	Cálculo do tempo	p. 42
4.4	Descrição do sistema	p. 43
4.4.1	Análises dinâmicas	p. 45
4.4.2	Análises estáticas	p. 47
4.4.3	Funcionalidades extras	p. 49
4.5	Diagrama de casos de uso	p. 50
4.6	Diagrama de classes	p. 51
4.7	Diagrama de atividades	p. 53
4.8	Testes realizados	p. 53
5	Análise dos resultados obtidos	p. 56
6	Conclusão	p. 63
	Referências Bibliográficas	p. 66
	Apêndice A – Cores e sons	p. 69
	Apêndice B – Protótipo da análise baseada em cores	p. 72
	Apêndice C – Diagramas de Atividades	p. 75
	Apêndice D – Mapa mental	p. 81

1 *Introdução*

Este trabalho visa analisar a possibilidade de estudar a instabilidade do algoritmo numérico através da estética da música, mais especificamente através da representação sonora de sequências numéricas, resultante do mapeamento de fractais que representam, de forma gráfica, o comportamento do algoritmo numérico.

A palavra *computador* vem do latim *computare* que significa *calcular*. Logo, computador é sinônimo de cálculos. Por ser uma máquina, o computador utiliza precisão finita e, portanto, realiza cálculos com valores aproximados.

O grau de incerteza do valor aproximado é denominado erro de arredondamento. O erro de arredondamento, quando associado à natureza do problema, determina a instabilidade do problema matemático. Assim, pequenas alterações nos dados de entrada produzem grandes alterações nos resultados obtidos. Um exemplo ilustrativo de instabilidade consiste em colocar um lápis comum, de base plana, em pé. Se a base for colocada para baixo, a tarefa de deixá-lo em pé será facilitada. Por outro lado, ao colocar a ponta cônica para baixo, a tarefa se torna quase impossível. Esta segunda maneira constitui um problema instável (CLÁUDIO; MARINS, 2000), uma vez que qualquer movimento, ao tentar deixar o lápis em pé, influenciará de forma aguçada no seu comportamento. Problemas como avaliação polinomial, resolução de sistemas de equações e avaliação de funções se incluem nesta classe de problemas matemáticos e, quando resolvidos em um computador, estão sujeitos ao erro de arredondamento.

Além do erro de arredondamento, no universo computacional, também é importante a complexidade do algoritmo, que é analisada pelo comportamento assintótico da função que descreve a relação entre tamanho da entrada (n) e tempo de execução ($t(n)$), sendo uma abordagem válida para a pesquisa científica. Expandindo a análise do problema para a pesquisa sistêmica, tem-se a seguinte questão: se a complexidade do algoritmo é medida em função de sua entrada, para resolução de dois sistemas de equações lineares com três equações de mesmo grau, por exemplo,

por que um deles pode demorar mais para executar se ambos possuem a mesma complexidade? A resposta está na instabilidade do problema.

Assim, analisar a complexidade do algoritmo numérico consiste não só em prever o esforço computacional, mas também sua instabilidade. Uma forma de realizar esta análise é associar a Teoria da Complexidade de Algoritmos à Teoria do Caos, uma vez que uma das características de um sistema caótico é representar a sensibilidade às condições iniciais. Compreendendo o comportamento caótico, muitas vezes é possível entender o comportamento da resolução do problema como um todo (CINTRA, 2005).

A representação gráfica da Teoria do Caos se dá através de fractais: *"O mundo que nos cerca é caótico, mas podemos tentar limitá-lo no computador. A geometria fractal é uma imagem muito versátil que nos ajuda a lidar com os fenômenos caóticos e imprevisíveis"* (MANDELBROT, 1982). A visualização da ordem em meio ao caos nem sempre deixa evidentes as relações que se estabelecem entre os elementos. Então, parece natural procurar recursos para potencializar esta visualização. Será que a música não pode ser um destes recursos? Como a estética das ciências pode ser expressa através das artes?

"Muda o mundo, mudam as formas de expressão."
(autor desconhecido)

1.1 Objetivos

1.1.1 Objetivo geral

O objetivo deste Trabalho de Conclusão é estudar a possibilidade de representação da instabilidade de um algoritmo numérico, que influencia na sua complexidade, através do som. Confirmando a possibilidade, é estudada a utilização da representação fractal na geração do som através de um *software* cuja implementação faz parte deste trabalho.

1.1.2 Objetivos específicos

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

- compreender a implementação algorítmica do método de Newton;

- analisar implementações existentes deste método;
- investigar bibliotecas e *softwares* existentes para a representação gráfica de fractais;
- obter imagens fractais para uma bateria de testes sobre resolução de equações pelo método de Newton;
- investigar bibliotecas e *softwares* existentes para a conversão de imagem fractal em música fractal;
- gerar os sons correspondentes aos padrões fractais identificados na bateria de testes;
- relacionar os padrões da música fractal com os padrões da representação gráfica do fractal;
- analisar possibilidades de aplicação dos resultados no ensino de métodos numéricos.

2 *Fundamentação teórica*

Neste Capítulo são apresentados os conceitos teóricos que forneceram suporte para os estudos no decorrer do trabalho.

2.1 Algoritmos numéricos

Algoritmos numéricos são sequências de instruções ordenadas de maneira a dar, em seu decurso, a solução para um problema voltado ao processamento numérico (CLÁUDIO; MARINS, 2000), ou seja, operações aritméticas são o núcleo do algoritmo que objetiva encontrar um ou mais resultados numéricos. Um algoritmo numérico codifica a solução de uma classe de problemas matemáticos através de métodos numéricos, que têm, como condição de parada, exatidão ou número de iterações. Quando a condição de parada atendida for exatidão, então é obtida a resposta do problema.

Ao transformar o algoritmo numérico em *software*, a manipulação de valores está associada ao erro de arredondamento que ocorre quando o valor não é representado exatamente por um número de máquina. Dependendo da natureza do problema e da forma como se estrutura o algoritmo, este erro pode se propagar e degradar irremediavelmente o resultado calculado.

O Sistema de Ponto Flutuante define os números que a máquina é capaz de representar. Este sistema é definido por $F = F(b, n, e_1, e_2)$, onde b é a base do sistema de enumeração ($b \geq 2$ e $b \in \mathbb{N}$), n é a precisão de máquina, e_1 é o menor expoente e e_2 é o maior expoente da configuração.

Para averiguar se certo número é representado como número de máquina, é preciso conhecer a configuração de F . O número a ser representado deve estar contido no intervalo de números de máquina, ou seja, no intervalo $[-M, M]$, onde $M = 0, [b - 1][b - 1] \dots [b - 1] * b^{e_2}$ e admitir

representação exata. Do contrário, o número será arredondado, podendo este arredondamento ser:

- por truncamento - todos os dígitos que excedem a precisão (n) são desconsiderados;
- por excesso - todos os dígitos que excedem a precisão são desconsiderados, porém soma-se 1 ao último dígito que está na precisão;
- científico - todos os dígitos que excedem a precisão são desconsiderados. Se o primeiro dígito que estourar a precisão for maior ou igual à metade da base do sistema ($d_{n+1} \geq b/2$), o último dígito é acrescido de uma unidade ($d_n + 1$).

Uma vez aplicado um arredondamento, a diferença entre o valor exato e o valor obtido é propagada, através dos cálculos, influenciando o resultado final. A reescrita do algoritmo pode tornar menor esta diferença ou até mesmo anulá-la, quando o problema é estável, ou seja, as alterações na entrada não influenciam fortemente a diferença do resultado. Contudo, isto não se aplica aos problemas instáveis onde, por melhor que seja a implementação do algoritmo, o resultado está sujeito a um grande erro de arredondamento.

2.1.1 Método de Newton

Um algoritmo numérico bastante utilizado é o do método de Newton para resolução de equações. O método permite o cálculo das raízes através de iterações, buscando aproximações para as raízes da equação. Com um valor inicial, utiliza-se a função $f(x)$ e sua derivada $f'(x)$ para calcular o valor da próxima iteração, de acordo com fórmula:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \text{ onde } f'(x_n) \neq 0. \quad (2.1)$$

Com o valor da próxima iteração, a fórmula é aplicada novamente e as iterações continuarão até que o valor seja suficientemente próximo da raiz, ou seja, que a quantidade de dígitos representativos seja aceitável (FREITAS, 2011) (VIANA, 2001).

Novamente, a "quantidade de dígitos necessária na representação de um número" é encontrada. Aqui se pode ter, além do arredondamento no resultado da operação de divisão na fórmula do método, os dígitos representados arredondados atingindo precariamente a exatidão - quando calculadas raízes muito próximas, por exemplo. O método de Newton apresenta sensibilidade aos dados de entrada: um valor inicial bastante próximo pode demorar muito mais ou até mesmo não convergir para a raiz.

2.2 Teoria do caos

Caos, segundo a mitologia grega, é o lugar onde as formas se formam. De fato, ele se apresenta como um dos elementos da tríade *Eros-Caos-Gaia*, onde *Eros* (éter, estado gasoso) representa o céu, *Gaia* (matéria, estado sólido) representa a terra e *Caos*, o espaço entre *Eros* e *Gaia*, onde as formas se formam. Trazendo para as áreas da matemática e computação, a Teoria do Caos estuda o comportamento de sistemas dinâmicos que são altamente sensíveis à condição inicial, ou seja, pequenas alterações na entrada de dados tornam o sistema caótico, instável e não previsível, sendo somente resolvido por algoritmos instáveis.

Comportamentos caóticos também podem ser observados em diversas outras áreas do conhecimento, como na medicina - nos estudos de epilepsia, tentando prever ocorrências aleatórias de um ataque epilético dadas as condições iniciais do paciente - uma vez que o organismo humano se comporta como um sistema complexo não-linear (GODOY, 2005), ou na meteorologia - um sistema caótico ao natural. Outro exemplo deste comportamento é o popularmente conhecido *Efeito Borboleta* (que deu origem, em 2004, ao filme homônimo), surgido em 1972 a partir do artigo publicado por Edward Lorenz, intitulado "*Predictability: Does the Flap of a Butterfly's Wings in Brazil set off a Tornado in Texas?*", onde o simples bater de asas de uma borboleta é considerado a condição inicial de um sistema que desencadeia uma série de eventos. Contudo, a trajetória do sistema poderia ser influenciada caso esta borboleta não batesse suas asas (LORENZ, 1972).

Para visualizar o comportamento caótico de um sistema, é utilizada a geometria fractal, como mostrado na Figura 2.1, onde se observa como o estado do sistema evolui - no tempo - durante o processo de resolução (mantendo, ou não, seu comportamento padrão).

2.3 Geometria fractal

Geometria fractal é o ramo da matemática que descreve o comportamento dos fractais. Fractais, por sua vez, são objetos da geometria não-euclidiana divisíveis em partes semelhantes ao objeto original. O conceito de fractal provém do latim *fractus* (divisível, quebrável) e foi utilizado pela primeira vez em 1975 por Benoit Mandelbrot. Tais objetos de formas estranhas foram considerados "monstros matemáticos" (FILLIPIN, 2009) por fugirem à razão da geometria euclidiana e, em casos extremos, ditos "lixo da ciência" antes de Mandelbrot provar o valor científico destes objetos.

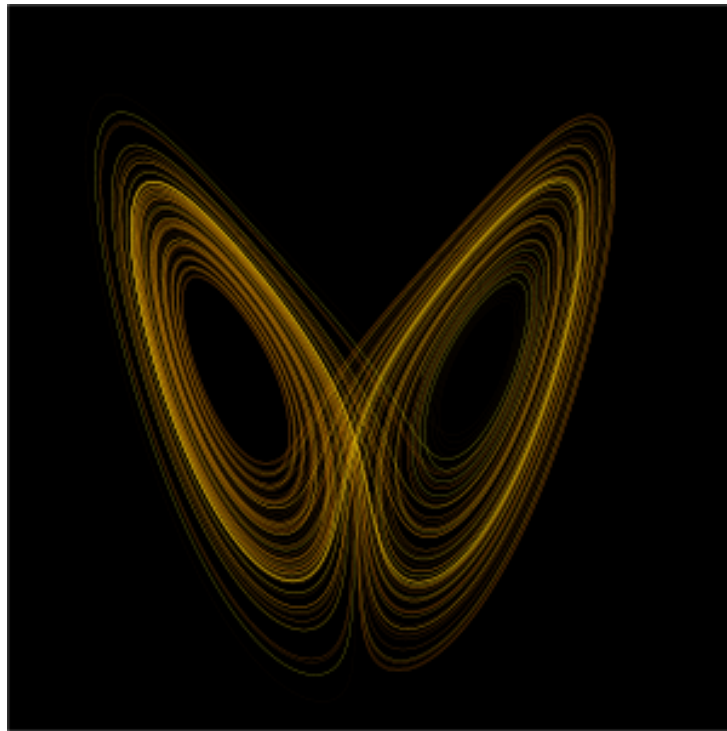


Figura 2.1: Atrator de Lorenz mostrando, através de um fractal, o comportamento de um sistema caótico.

Fonte: http://en.wikipedia.org/wiki/File:Lorenz_attractor_yb.svg

A necessidade do uso dos fractais, para representação do mundo em que se vive, surgiu da célebre frase de Mandelbrot: *"Nuvens não são esferas, montanhas não são cones, continentes não são círculos, o som do latido não é contínuo e nem o raio viaja em linha reta"*(MANDELBROT, 1982), ou seja, a geometria euclidiana não é capaz de representar tais ocorrências da natureza.

De acordo com (NUNES, 2006), as seguintes características podem ser encontradas em um fractal:

- autossemelhança - apresenta sempre o mesmo aspecto visual - total (exato) ou estatisticamente aproximado - em qualquer escala, como o exemplo da Figura 2.3;
- escala - ampliação ou redução do fractal. Intrinsecamente ligada à característica de autossemelhança, a escala da representação do fractal será semelhante a seu nível anterior ou posterior;
- complexidade infinita - os fractais, por apresentarem autossemelhança nas escalas, são de complexidade infinita uma vez que, quando gerados por funções iteradas, podem ser gerados infinitamente;



Figura 2.2: Um raio é um belo exemplo de um fractal na natureza, pois sua trajetória não pode ser representada por uma reta da geometria euclidiana.

Fonte: <http://www.sobralportaldenoticias.com/v1/wp-content/uploads/2011/08/Raio-011.jpg>

- dimensão - representa o nível de irregularidade ou a complexidade de um objeto fractal (BACKES; BRUNO, 2005). Suas aplicações são diversas como, por exemplo, a medição da costa de um país (MANDELROT, 1967).

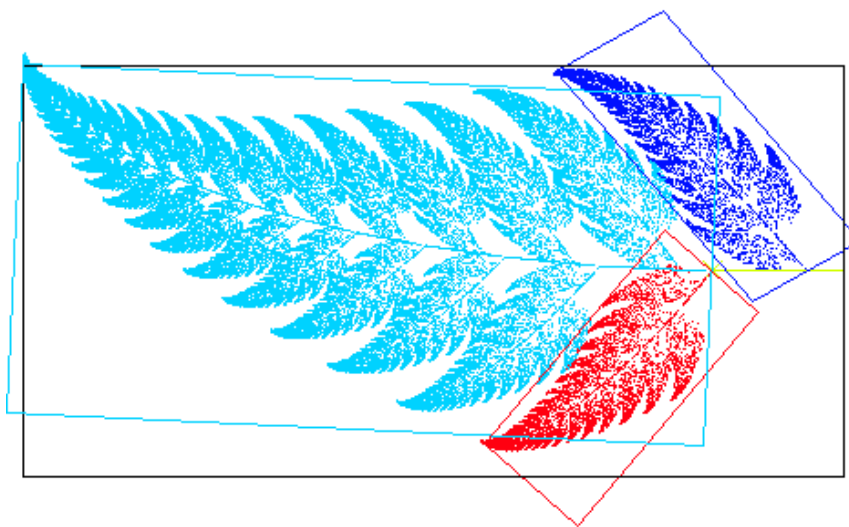


Figura 2.3: A folha de uma samambaia é um exemplo de fractal autossemelhante

Fonte: http://pt.wikipedia.org/wiki/Ficheiro:Fractal_fern_explained.png

Ao aprofundar os estudos nesta área, além das características observadas, os fractais são classificados de acordo com a maneira como são gerados (FERNANDES, 2007):

- fractais de funções iteradas - gerados passo a passo, por regras pré-fixadas de substituição geométrica, em sistemas de funções iteradas;
- fractais de fuga no tempo - possuem uma relação de recorrência de cada um dos seus pontos no plano complexo (espaço);
- fractais aleatórios - gerados por processos não-determinísticos, têm similaridade estatística.

Métodos numéricos iterativos podem gerar um fractal, mostrando o comportamento, durante o processo de solução, como ilustrado pela Figura 2.4, onde as cores estão associadas ao número de iterações do processo de solução.

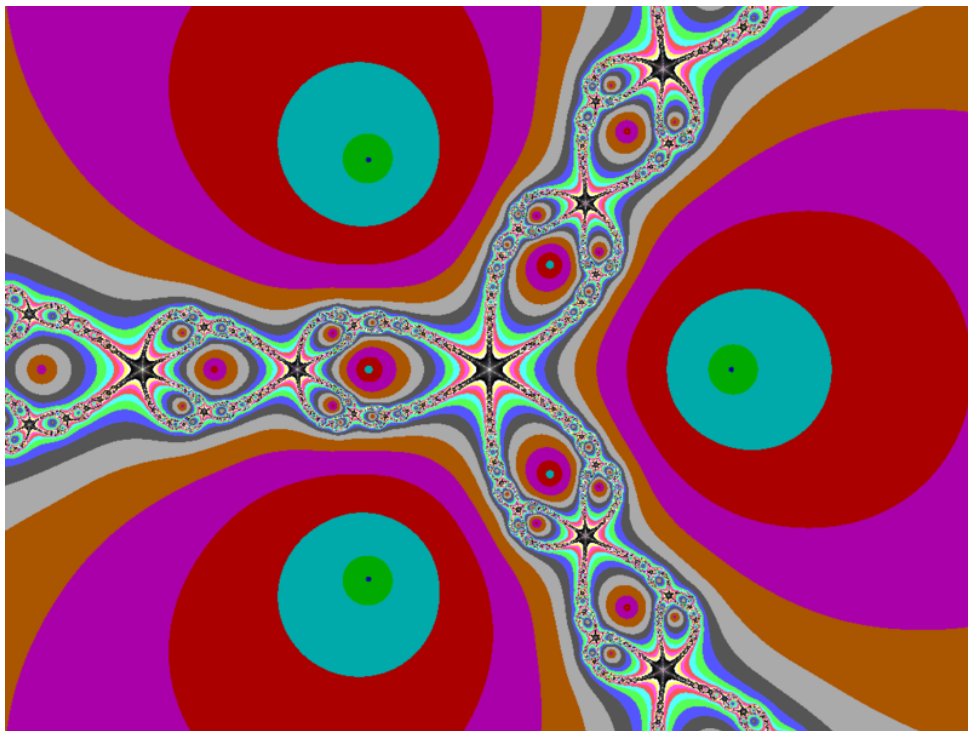


Figura 2.4: O fractal de funções iteradas gerado pelo método numérico de Newton para o cálculo das raízes do polinômio $f(x) = x^3 - 1$

Fonte: <http://en.wikipedia.org/wiki/File:FRACT008.png>

A partir do padrão fractal gerado pelo método numérico, é estabelecida a relação com a Teoria do Caos, mostrando a complexidade e a instabilidade do problema. A partir desta relação,

o passo seguinte é a conexão do conceito de caos ao conceito de ordem (seu complemento). A ordem pode ser associada à harmonia, que direciona à outra área de conhecimento: o som.

2.4 Som

Antigamente, a acústica - ramo da ciência que estuda as ondas mecânicas que geram os sons - já era bastante estudada e aplicada. Marco Polião, engenheiro militar de Júlio César, fez vários projetos acústicos para os teatros romanos, onde chegou a utilizar vasos de bronze como ressonadores. Pitágoras procurou saber por que alguns intervalos de uma corda vibrante em seu monocórdio pareciam mais bonitos que outros e, no século VI a.C., observou que a altura do som produzido por uma corda vibrando varia de acordo com seu comprimento (MINGATOS, 2006).

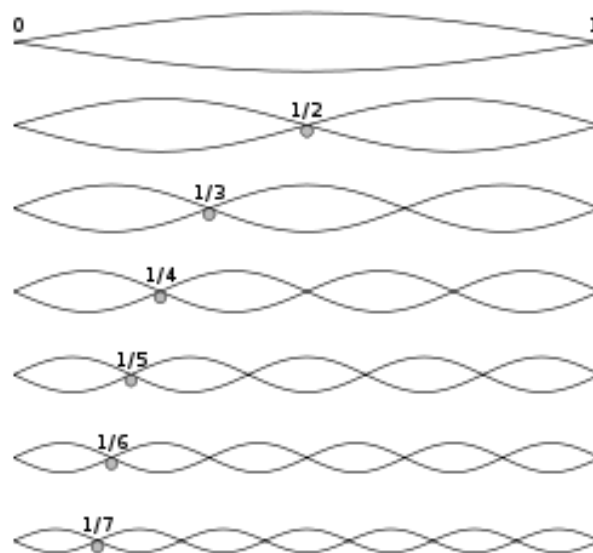


Figura 2.5: O tom fundamental e os seis sobretons de uma corda vibrante: o estudo de Pitágoras.

Fonte: http://en.wikipedia.org/wiki/File:Harmonic_partials_on_strings.svg

Hoje em dia, sons e ruídos estão demasiadamente impregnados no cotidiano de forma que, na maioria das vezes, passam despercebidos, devido ao turbilhão de sons escutados indiscriminada e simultaneamente.(MORAES, 2000).

Com a junção destes sons e ruídos contidos nas ondas sonoras, chega-se à música que, por sua vez, é uma ordem que se constrói de sons (enquanto recorrência periódica, produção

de constância) e o ruído (enquanto perturbação relativa da estabilidade, superposição de pulsos complexos, irracionais, defasados) (WISNIK, 1989). Segundo (CARDOSO, 2009), o som produtor da música pode ser definido por seis parâmetros:

- altura - definido na acústica pela frequência, é a nota; define se um som é grave ou agudo;
- duração - tempo cronológico da acústica; é a duração da emissão do som;
- intensidade - medida pela amplitude, representa a força imprimida ao tocar um som;
- timbre - material do objeto sonoro representado pelas vozes ou instrumentos;
- articulação - na acústica, conhecido como ataque; é o modo de produção do som;
- andamento - a velocidade de execução de um som.

2.5 Música

A palavra música é de origem grega e significa "força das musas". As musas ensinavam aos seres humanos a verdade dos deuses através de diferentes formas de arte, principalmente pelo canto. Até alguns séculos atrás, a música tinha função utilitária: rituais, comunicações, trabalho, lazer e atividades socio-artísticas. Hoje é a arte da inteligência humana trabalhar com sons (CARDOSO, 2009).

Atualmente, ao falar em música, existem lacunas devido à variação nas abordagens em diferentes culturas. Citando (WISNIK, 1989): *"Enquanto experiência do mundo em seu caráter intrinsecamente ondulatório, o som projeta o limiar do sentido na medida de sua estabilidade e instabilidade relativas. Esse sentido é vazado de historicidade - não há nenhuma medida absoluta para o grau de estabilidade e instabilidade do som, que é sempre produção e interpretação das culturas: certos intervalos, certos ritmos, certos timbres adotados aqui podem ser recusados ali ou, proibidos antes, podem ser fundamentais depois"* ou ainda *"O modo de conceber as escalas musicais é decisivo em cada cultura pra fazer a relação entre som e ruído e, com o tempo, tornou mais improvável a medição do limiar diferencial entre ordem e não ordem na música."*, pode-se perceber essa variação da abordagem cultural. Desta maneira, foi assumido o padrão da cultura ocidental, no que trata de geração musical, quando necessário para definir estabilidade.

Neste trabalho, a música relaciona som e ruído com fractais: a chamada música fractal. Tal tipo de música é obtida diferentemente das formas tradicionais onde um conjunto de notas musicais é repetido para dar o ritmo. Na música fractal, a próxima nota musical é obtida durante o processo de geração da música.

2.5.1 Notas musicais e suas frequências

Para representar a linguagem falada, são utilizadas as letras do alfabeto. Para representar a música, as notas musicais. As sete notas musicais conhecidas hoje - *Dó, Ré, Mi, Fá, Sol, Lá* e *Si* - surgiram na Idade Média, através do sistema chamado solmização, proposto pelo monge beneditino Guido de Arezzo, com o objetivo de facilitar o aprendizado e diminuir os erros de interpretação de uma peça musical. Para tanto, Guido retirou as primeiras sílabas do hino cantado em louvor a São João Batista, onde cada frase era cantada em um grau acima na escala musical (BARTZ, 2011):

*Ut queant laxis,
Resonare fibris,
Mira gestorum,
Famuli tuorum,
Solve polluti,
Labbii reatum.
Sancte Iohannes*

Mais tarde, Giovanni Doni, músico italiano, propôs a substituição de *Ut* por *Dó*. Com o acréscimo de outra nota *Dó* ao final destas sete, obtém-se a escala diatônica típica da música ocidental: cinco intervalos de um tom (*Dó/Ré, Ré/Mi, Fá/Sol, Sol/Lá* e *Lá/Si*) e dois intervalos de semitom (*Mi/Fá* e *Si/Dó*, visíveis no teclado do piano acústico pela ausência das teclas pretas entre essas notas) (WISNIK, 1989). Cada nota musical está associada a uma duração e uma frequência, podendo ser representada em uma pauta musical, como a da Figura 2.6: cinco linhas horizontais paralelas, com quatro espaços (contados de baixo para cima) entre elas, onde cada representação mostra a fração do tempo que a nota deverá ser executada em função do ritmo.

A frequência das notas foi demonstrada por Pitágoras, assim como apresentado na Figura 2.5: a corda, em partes menores, vibra com uma frequência maior. Com o avanço dos estudos,



Figura 2.6: A duração das notas em uma pauta.

Fonte: <http://www.profcardy.com/cardicas/musical.php>

sabe-se que a frequência deve estar compreendida entre 20 *hertz* e 20 *kilohertz* para ser audível ao ouvido humano, representando, respectivamente, os sons mais graves e mais agudos. Pode, ainda, variar de acordo com a oitava em que é tocada.

A oitava, por sua vez, é o intervalo entre uma nota musical e outra com o dobro de sua frequência, ou seja, a próxima ocorrência desta nota/frequência. A oitava central foi conveniada como a que contém a nota *Lá* na frequência de 440 *hertz* e é usada para afinação dos instrumentos musicais. Tomando, por exemplo, a nota *Dó* da escala diatônica na oitava central e a sua próxima ocorrência - o próximo *Dó* uma oitava acima - tem-se, respectivamente, as frequências 261,62 *hertz* e 523,25 *hertz*. A Tabela 2.1 mostra a frequência das notas musicais nas onze oitavas que compreendem a faixa audível. Um exemplo é apresentado na Figura 2.7, mostrando o intervalo das frequências que alguns instrumentos musicais e a voz conseguem atingir.

Nota	<i>Dó</i>	<i>Ré</i>	<i>Mi</i>	<i>Fá</i>	<i>Sol</i>	<i>Lá</i>	<i>Si</i>
Oitava -1	16,35	18,35	20,60	21,82	24,49	27,50	30,86
Oitava 0	32,70	36,70	41,20	43,65	48,99	55,00	61,73
Oitava 1	65,40	73,41	82,40	87,30	97,99	110,00	123,47
Oitava 2	130,81	146,83	164,81	174,61	195,99	220,00	246,94
Oitava 3	261,62	293,66	329,62	349,22	391,99	440,00	493,88
Oitava 4	523,25	587,32	659,25	698,45	783,98	880,00	987,76
Oitava 5	1046,50	1174,65	1318,51	1396,91	1567,98	1760,00	1975,53
Oitava 6	2093,00	2349,31	2637,02	2793,82	3135,96	3520,00	3951,06
Oitava 7	4186,00	4698,63	5274,04	5587,65	6271,92	7040,00	7902,13
Oitava 8	8372,01	9397,27	10548,08	11175,08	12543,85	14080,00	15804,26
Oitava 9	16744,03	17739,68	21096,16	22350,60	25087,71	28160,00	31608,52

Tabela 2.1: As frequências (em *hertz*) das notas em cada oitava.

Como a música é gerada por uma quantidade finita de notas, é comum que haja repetição ou uma diferença significativa entre a nota tocada e sua adjacente na pauta, causando variação na periodicidade ou na estabilidade da composição.

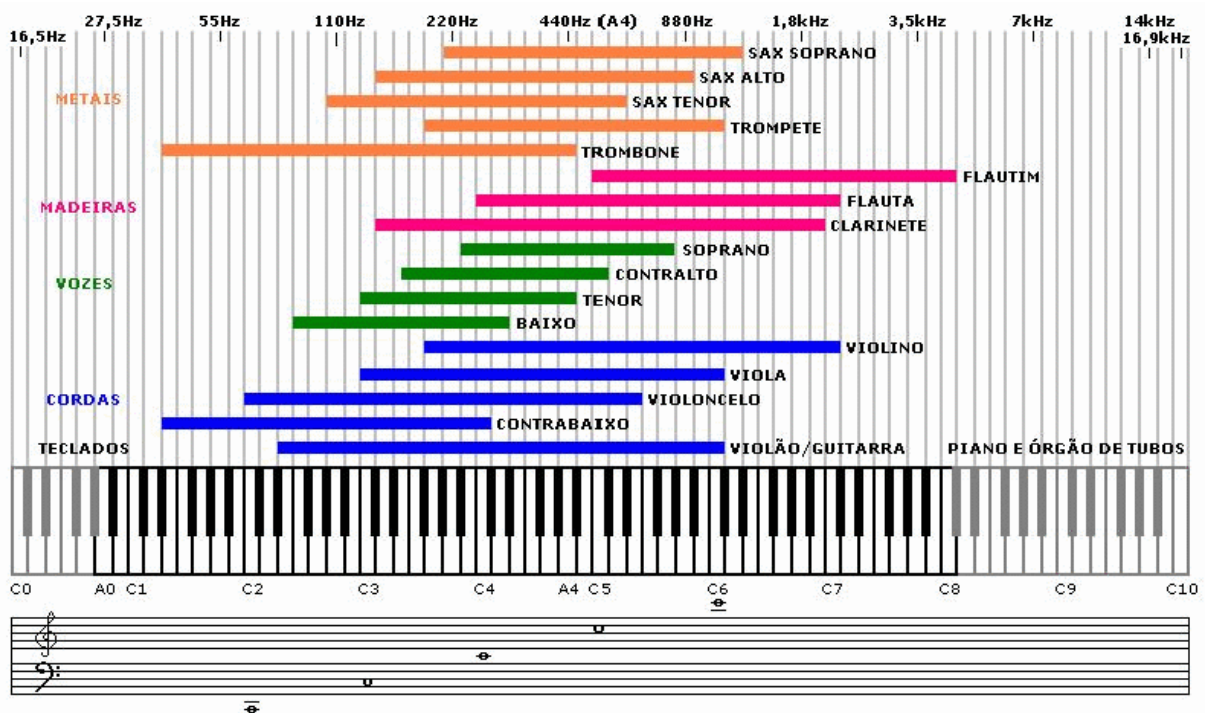


Figura 2.7: O espectro das frequências que alguns instrumentos musicais e a voz conseguem atingir.

Fonte: http://www.audionasigrejas.org/Apostila/Apostila_html_1fc97d78.gif

2.5.2 Periodicidade e a estabilidade da música

A natureza oferece dois grandes modos de experiências da onda complexa que faz o som: frequências regulares, constantes, estáveis, como aquelas que produzem o som afinado, com altura definida, e frequências irregulares, inconstantes, instáveis, como aquelas que produzem barulhos, manchas, rabiscos sonoros, ruídos (WISNIK, 1989).

"Se você tem um barulho percutido qualquer e ele começa a se repetir e a mostrar uma certa periodicidade, abre-se um horizonte de expectativa e virtualidade de uma ordem subjacente ao pulso sonoro em suas regularidade e irregularidades. Do mesmo modo, se você está falando e de repente produz e sustenta um som de altura definida, remete a fala para um outro lugar, o paradigma das alturas contínuas, não codificado pela língua, com toda a estranheza que isso implica (e pode-se saltar então do patamar da fala para o do canto, ou habilitar o espaço intercalar entre ambos)". Com estas palavras (WISNIK, 1989), constata-se que o som encontrado na natureza é dotado de periodicidade.

Outra característica do som também é exemplificada pelo mesmo autor, através da escala pentatônica - escala interrompida na quinta nota, estruturada por *Fá, Sol, Lá, Dó e Ré*, comum na música oriental - mostrando a instabilidade da música com a seguinte figuração: *"Segundo um tratado cerimonial clássico, a nota kong (fá) representa o príncipe; chang (sol) os ministros; kio (lá) o povo; tché (dó) os negócios e yu (ré) os objetos. A escala corresponde ao jogo - estável e instável - da ordem social, cujo equilíbrio ela reproduz (metaforicamente) e contribui para manter (metonimicamente). O microcosmo sonoro, imbricado com a ordem social e política, compromete também o equilíbrio entre o céu (correspondente aos 'sons claros e distintos', agudos) e a terra (correspondente aos 'sons fortes e poderosos', graves): equilíbrio mantido pela coluna simbólica das relações harmônicas. Está visto aí por que as relações escalares nas tradições modais mais remotas não devem mudar, sob a pena de infligir graves danos à ordem das coisas. Cada uma das notas (ou cada parte integrante da realidade natural e social) deve contribuir para o bom funcionamento (perpetuamente mutável) de um todo imutável. 'Se kong é perturbado, o som é desordenado; é que o príncipe é arrogante. Se chang é perturbado, o som é pesado; é que os ministros se perverteram. Se kio é perturbado, o som é doloroso; é que os negócios estão difíceis. Se yu é perturbado, o som é ansioso; é que as fortunas estão esgotadas. Se os cinco sons estão perturbados, as categorias interferem umas sobre as outras; e é o que se chama de insolência. Se assim for, a queda do reino intervirá em menos de um dia.'"*

Um som com periodicidade indefinida - sem padrão, instável, é um som "estranho" ao ouvido humano. Tal exemplo pode ser visto na apresentação *"The World's Ugliest Music"* de Scott Rickard (RICKARD, 2011). Segundo Scott *"A música mais feia do mundo é totalmente ausente de repetições, é algo que somente um matemático poderia escrever"*. Esta música é composta de sons cujas notas são resultantes de processos matemáticos que garantem a não periodicidade (ou não repetição). *"Até uma música produzida aleatoriamente poderá ter repetições, esta é totalmente livre"*, afirma Scott.

Já a estabilidade do som está intrinsecamente ligada à harmonia da música: é necessário que as notas não fujam à regularidade da relação entre si. Uma música deverá ter harmonia com as notas tocadas simultaneamente (diferente do que é encontrado na música de Scott), o que fará dela algo interessante de se ouvir.

Os fractais também são dotados de padrões e instabilidades em maior ou menor grau, dependendo do caos do sistema que representam. A relação entre sons e fractais produz a música fractal.

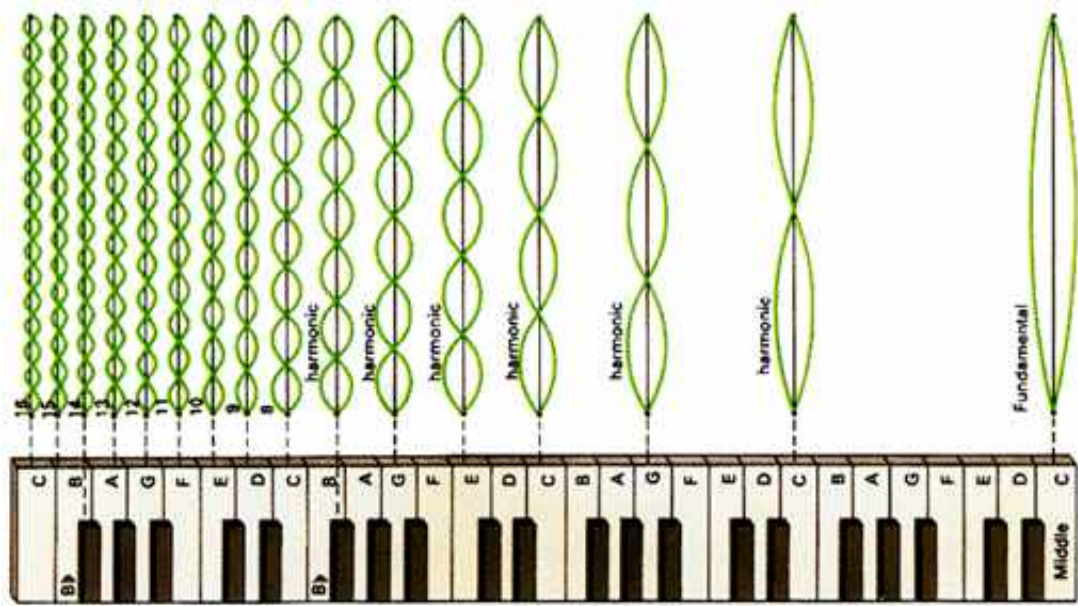


Figura 2.8: A relação harmoniosa das notas representada nas teclas do piano.

Fonte: <http://culturadigital.br/dacio/files/2011/05/musica81-300x169.jpg>

2.5.3 Música fractal

Apesar de muitos não considerarem a música fractal como uma verdadeira forma de música por não ser gerada a partir de um processo criativo e inspirado, ela se apresenta de forma útil em contextos matemáticos ou teóricos, podendo também ser empregada na geração de outros sons, se devidamente trabalhada. Este tipo de música é assim chamado não por ser um gênero musical e sim pela forma como é criado. Gerada a partir de uma projeção em um espaço musical do comportamento de um fractal, a música fractal seguirá - na maioria dos casos - o padrão do fractal. Se o fractal for autossemelhante e sua escala não variar, é provável que a música também seja (ORTIZ, 2000).

Na prática, pode-se gerar música fractal a partir de:

- nota musical - com esta nota (que pode ser um número convertido em nota ou de acordo com a especificação do *software*), aplica-se um processo iterativo e gera-se a nota seguinte. Em seguida, a mesma fórmula é aplicada gerando a terceira nota e assim sucessivamente como, por exemplo, nos métodos *L-Systems* (ZAMMOURI; AYEB, 2007) - a cada iteração é produzida a substituição do símbolo inicial pelo resultado de suas regras de produção (VILELA, -);

- imagem fractal - dada uma imagem de um fractal, esta é processada e o som correspondente é gerado. O processamento pode ser realizado, no mínimo, de duas maneiras: na primeira, em imagens coloridas, a imagem é percorrida linha por linha tomando-se a cor de cada pixel (onde cada cor representa uma nota); na segunda, a localização de um pixel na imagem (pixel de uma borda, por exemplo) é o gerador da nota.

A música fractal produzida pelo homem é feita por meio de ajustes e união de diversos fractais, visando aproximar um *pink noise* (um som mais agradável ao ouvido). Este som está compreendido entre o *white noise* (caos completo, desordenado) e o *brown noise* (demasiado maçante) (HAZARD; KIMPORT; JOHNSON, 1998 - 1999).

No entanto, a música fractal "crua", gerada somente por um fractal, podendo representar um sistema caótico, por exemplo, pode não soar tão agradável ao ouvido humano, aproximando-se de um *white noise*. Ao identificar o ponto em que uma música fractal se aproxima de um *pink noise* ou de um *white noise*, pode-se verificar se o fractal que a gerou é a representação de um algoritmo cujo comportamento está mais próximo da estabilidade ou da instabilidade.

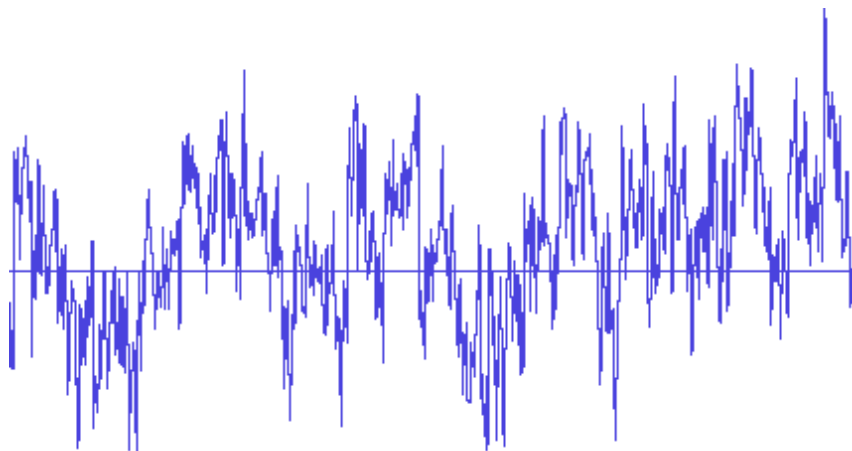


Figura 2.9: Forma de ondas de um *pink noise* mais agradável ao ouvido humano.

Fonte: <http://www.tursiops.cc/fm/pink.gif>

Para ouvir a música fractal associada a um algoritmo numérico, a mesma deve ser gerada em um computador e sintetizada através da placa de som ou do emulador de som do Sistema Operacional. Para tanto, o *software* gerador deverá utilizar um padrão compatível com os sintetizadores, como o padrão MIDI.

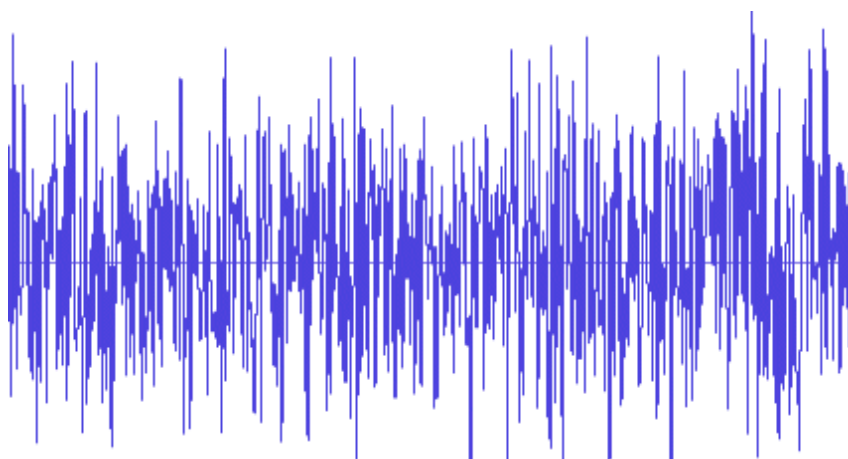


Figura 2.10: Forma de ondas de um *white noise*. Combinação de sons de todas frequências; nosso cérebro não é capaz de distingui-las.

Fonte: <http://www.tursiops.cc/fm/white.gif>

2.6 Padrão MIDI

O desenvolvimento do padrão MIDI - *Musical Instrument Digital Interface* - ocorreu devido à necessidade de se criar um padrão de comunicação entre instrumentos musicais e computadores, através de uma linguagem comum (SILVA, 2007a). Com este padrão é possível que uma composição musical seja executada em qualquer dispositivo que o reconheça, além de possibilitar o armazenamento destas composições em arquivos com extensão *.mid* - uma forma de partitura digitalizada - diferente dos arquivos de áudio digital, como o *.mp3*. Em um computador, uma mensagem MIDI é recebida (enviada por um controlador - um teclado ou um *software*, por exemplo) e sequenciada. Posteriormente, a placa de som a executa ou o Sistema Operacional emula este som. É como se o controlador dissesse: "*toque a nota Fá durante 4 tempos a um andamento de 120 BPM, com volume máximo, utilizando timbre de flauta transversal*" (OLIVEIRA, 2009).

A linguagem de programação Java oferece uma interface multiplataforma baseada em *software* que possibilita trabalhar com o padrão MIDI: a *JavaSound* e sua arquitetura *MIDI audio* (BECKER, 2009). Com esta interface é possível criar MIDIs e sequenciá-las salvando em arquivos para serem sintetizados. A arquitetura da *MIDI audio* é demonstrada na Figura 2.11.

Através desta interface, tem-se a possibilidade de tocar 128 notas, começando pelo *Dó* na oitava número -1 (diferentemente de um piano acústico que inicia pelo *Lá* na oitava 0). Cada nota corresponde a um número no sistema decimal, num intervalo de 0 a 127. O *Lá*

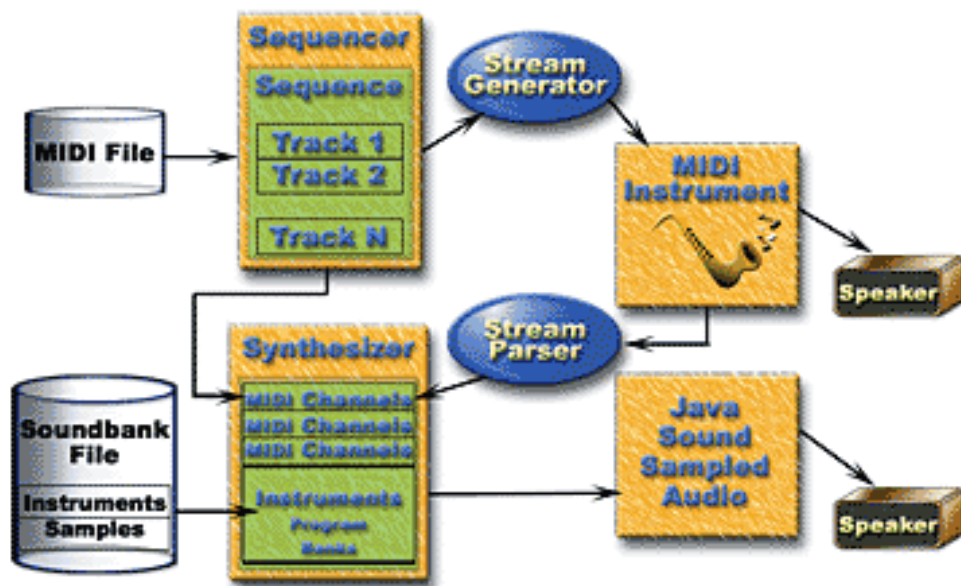


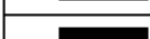
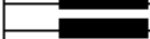
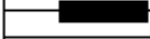
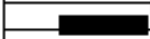
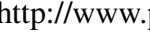
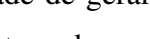
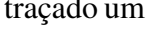
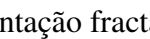
Figura 2.11: A arquitetura da *MIDI audio* para sequenciamento e sintetização (reprodução) dos sons.

Fonte: <http://www.ibm.com/developerworks/library/it/it-0801art38/art38fig1.gif>

central (440 *hertz*) corresponde à nota de número 69. A Tabela 2.2 mostra os números destas notas (STASKEVICH, 2011). Vale ressaltar, a fim de conformidade, que algumas literaturas apresentam indexações diferentes para as oitavas - começando, por exemplo, no índice 0 - e, portanto, foi assumida a indexação já utilizada na Tabela 2.1. Já a Figura 2.12 mostra os números das notas MIDI's em um piano acústico.

	<i>Dó</i>	<i>Dó♯</i>	<i>Ré</i>	<i>Ré♯</i>	<i>Mi</i>	<i>Fá</i>	<i>Fá♯</i>	<i>Sol</i>	<i>Sol♯</i>	<i>Lá</i>	<i>Lá♯</i>	<i>Si</i>
Oitava -1	0	1	2	3	4	5	6	7	8	9	10	11
Oitava 0	12	13	14	15	16	17	18	19	20	21	22	23
Oitava 1	24	25	26	27	28	29	30	31	32	33	34	35
Oitava 2	36	37	38	39	40	41	42	43	44	45	46	47
Oitava 3	48	49	50	51	52	53	54	55	56	57	58	59
Oitava 4	60	61	62	63	64	65	66	67	68	69	70	71
Oitava 5	72	73	74	75	76	77	78	79	80	81	82	83
Oitava 6	84	85	86	87	88	89	90	91	92	93	94	95
Oitava 7	96	97	98	99	100	101	102	103	104	105	106	107
Oitava 8	108	109	110	111	112	113	114	115	116	117	118	119
Oitava 9	120	121	122	123	124	125	126	127				

Tabela 2.2: Os números MIDI's das notas em cada oitava.

MIDI number	Note name	Keyboard	Frequency Hz	Period ms
21	A0		27.500	36.36
23	B0		30.868	32.40
24	C1		32.703	30.58
26	D1		36.708	27.24
28	E1		41.203	24.27
29	F1		43.654	22.91
31	G1		48.999	20.41
33	A1		55.000	18.18
35	B1		61.735	16.20
36	C2		65.406	15.29
38	D2		73.416	13.62
40	E2		82.407	12.13
41	F2		87.307	11.45
43	G2		97.999	10.20
45	A2		110.00	9.091
47	B2		123.47	8.099
48	C3		130.81	7.645
50	D3		146.83	6.811
52	E3		164.81	6.068
53	F3		174.61	5.727
55	G3		196.00	5.102
57	A3		220.00	4.545
59	B3		246.94	4.050
60	C4		261.63	3.822
62	D4		293.67	3.405
64	E4		329.63	3.034
65	F4		349.23	2.863
67	G4		392.00	2.551
69	A4		440	2.273
71	B4		493.88	2.025
72	C5		523.25	1.910
74	D5		587.33	1.703
76	E5		659.26	1.517
77	F5		698.46	1.432
79	G5		783.99	1.276
81	A5		880.00	1.136
83	B5		987.77	1.012
84	C6		1046.5	0.9556
86	D6		1174.7	0.8513
88	E6		1318.5	0.7584
89	F6		1396.9	0.7159
91	G6		1568.0	0.6378
93	A6		1760.0	0.5682
95	B6		1975.5	0.5062
96	C7		2093.0	0.4778
98	D7		2349.3	0.4257
99	E7		2637.0	0.3792
101	F7		2793.0	0.3580
103	G7		3136.0	0.3189
105	A7		3520.0	0.2841
107	B7		3951.1	0.2531
108	C8		4186.0	0.2389

www.phys.unsw.edu.au/music

Figura 2.12: A relação das notas MIDI's com as notas de um piano acústico

Fonte: <http://www.phys.unsw.edu.au/jw/graphics/notes.GIF>

Unindo a possibilidade de gerar e sequenciar sons em um computador às demais teorias vistas até agora, já se tem traçado um possível caminho do algoritmo numérico à geração sonora, passando por sua representação fractal.

3 *Prospecção da infraestrutura de software*

Neste Capítulo é apresentado um levantamento de ferramentas utilizadas na geração dos fractais produzidos a partir de sequências iterativas, bem como na conversão destes fractais em representações sonoras.

3.1 Geração de fractais

Para gerar os fractais correspondentes ao processo de resolução de equações através do método de Newton, é utilizado o *software Newton Fractal* desenvolvido por Artur Luiz Silva Freitas no Trabalho de Conclusão "*Um estudo do método de Newton fractal para resolução de equações*" (FREITAS, 2011). O trabalho teve por objetivo investigar a possibilidade de utilizar a exploração visual na identificação da estabilidade de métodos numéricos. O autor discorre sobre a análise do método de Newton gerando imagens cujas formas gráficas fractais identificam regiões de atratores ou bacias de convergência.

No *Newton Fractal* são informados os coeficientes de uma equação e suas raízes são calculadas no *Octave*¹. É possível definir uma cor para cada raiz. Pode-se configurar os limites do plano em que o fractal será gerado, o número máximo de iterações até a convergência e as dimensões da tela que mostrará o plano. A interface do *Newton Fractal* pode ser visualizada na Figura 3.2.

Ao configurar o *software* para resolver o polinômio $f(z) = z^5 + z^2 + z + 1$, obtém-se o fractal mostrado na Figura 3.2.

¹<http://www.gnu.org/software/octave/>

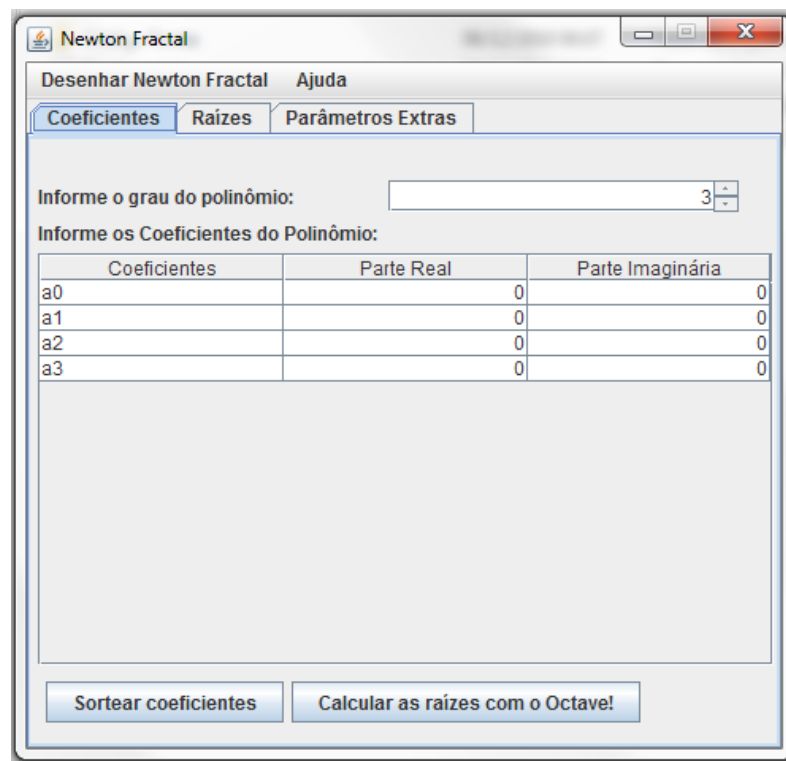


Figura 3.1: A interface do programa desenvolvido por Artur Freitas

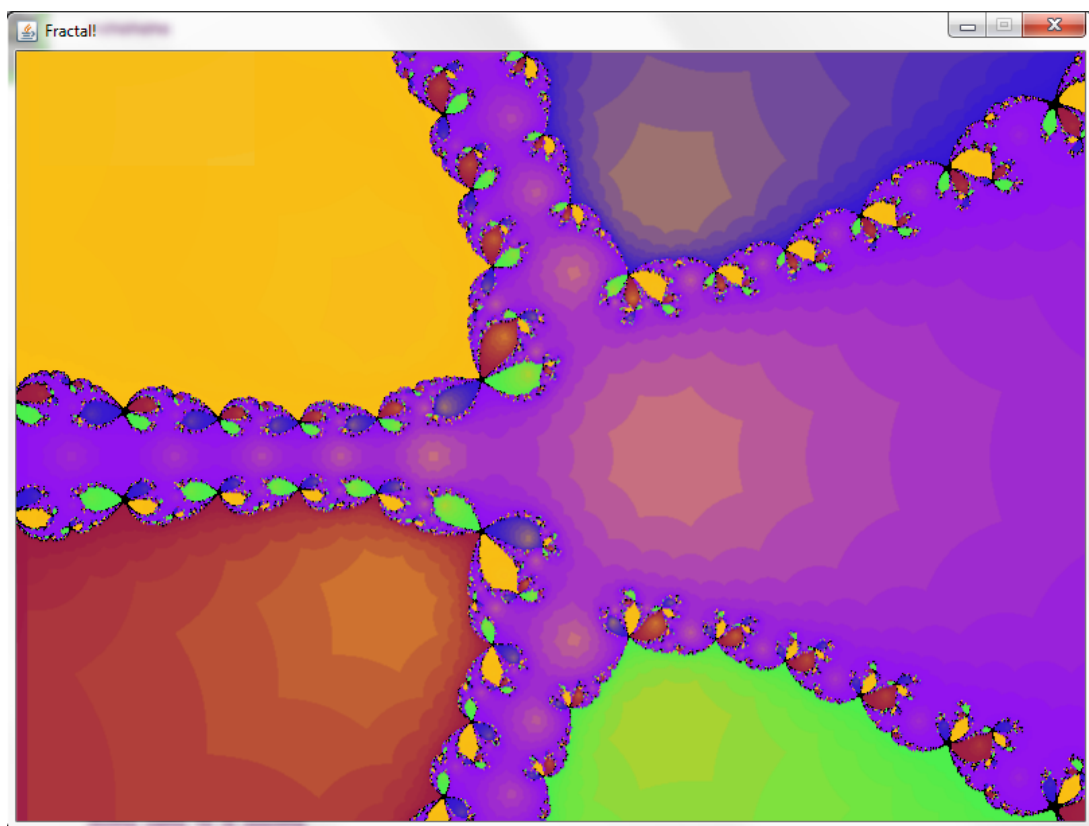


Figura 3.2: Um fractal gerado pelo *Newton Fractal* para o cálculo das raízes do polinômio $f(z) = z^5 + z^2 + z + 1$.

O *software Newton Fractal*, auxiliado pelo *Octave*, fornece a representação gráfica da sequência de valores gerada no processo de cálculo das raízes do polinômio, ou seja, gera o fractal necessário para iniciar a conversão em música. Além disto, o *software* é de código aberto², o que permite que seja verificada a maneira como o fractal é gerado, facilitando a manipulação de dados e a inclusão de novos módulos ao *software*.

3.2 Conversão de imagem em música

Os *softwares* apresentados a seguir possibilitam a leitura de uma imagem e sua transformação em música. Em uma primeira análise, o fractal gerado pelo *Newton Fractal* pode ser usado como entrada nestes *softwares*. Três *softwares* foram analisados, sendo que cada um tem uma abordagem diferente na geração da música.

3.2.1 Photosounder

"*Todo som que você já ouviu pode ser representado como uma imagem e todos os sons possíveis podem ser feitos a partir de uma imagem*". Essa é a proposta do *Photosounder*³. O programa foi utilizado nos testes para leitura da imagem de um fractal e a transformação desta em som. Na Figura 3.3, pode-se visualizar uma imagem do conjunto de Mandelbrot carregada no *software*.

Os testes foram efetuados com diferentes imagens de fractais, porém o *software* não se mostrou adequado à necessidade do trabalho: a geração de música fractal. O *Photosounder* gera sons com base na escala vertical da imagem (*pitch*), na frequência horizontal (tempo), na intensidade da imagem (a própria intensidade do som gerado) e as cores são descartadas. O *software* também sintetiza o som pela modulação de cada linha da imagem com um *bandpassed noise* que corresponde à frequência que seus pixels representam. Essa abordagem não permite fazer relação das características matemáticas do fractal com a música que será gerada, uma vez que a análise é feita pela sintetização da imagem e não pela semântica do conteúdo desta.

²<http://code.google.com/p/newton-fractal/>

³<http://photosounder.com/>

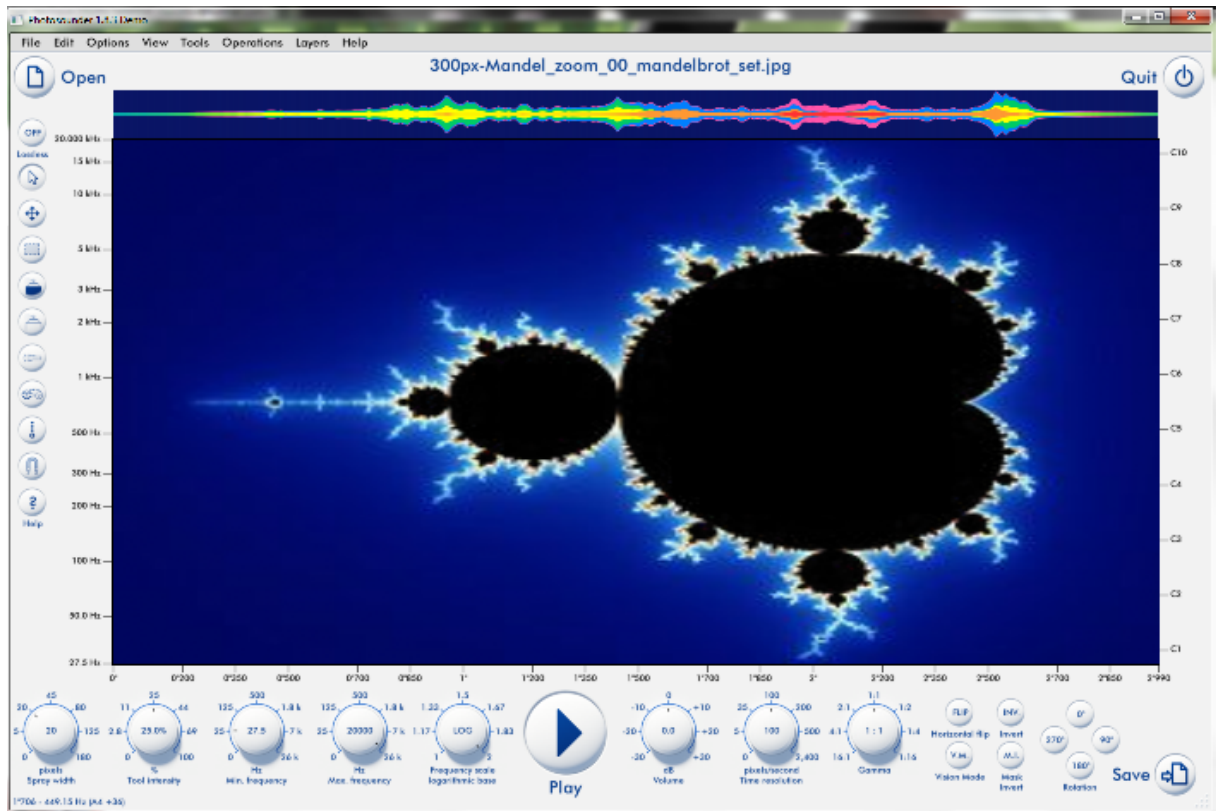


Figura 3.3: A interface do *Photosounder* quando carregada uma imagem do conjunto de Mandelbrot.

3.2.2 AudioPaint

O *AudioPaint*⁴ é outro *software* capaz de gerar sons a partir de uma imagem. Esta imagem é processada como uma grade de frequência por tempo. Cada linha da figura age como um oscilador, alimentando a geração do som. A posição vertical do pixel determina a frequência da oscilação enquanto a horizontal corresponde ao tempo do deslocamento desta. As cores vermelho e verde são levadas em consideração para controlar o volume dos canais direito e esquerdo de som, respectivamente. A cor azul não é utilizada. A interface do *AudioPaint* é mostrada na Figura 3.4.

Novamente, a ferramenta não interpreta a imagem de maneira relevante ao trabalho. As cores, em sua totalidade, não representam exclusivamente o som gerado, de modo que foi descartada a abordagem por cores neste *software*.

⁴<http://www.nicolasfournel.com/audiopaint.htm>

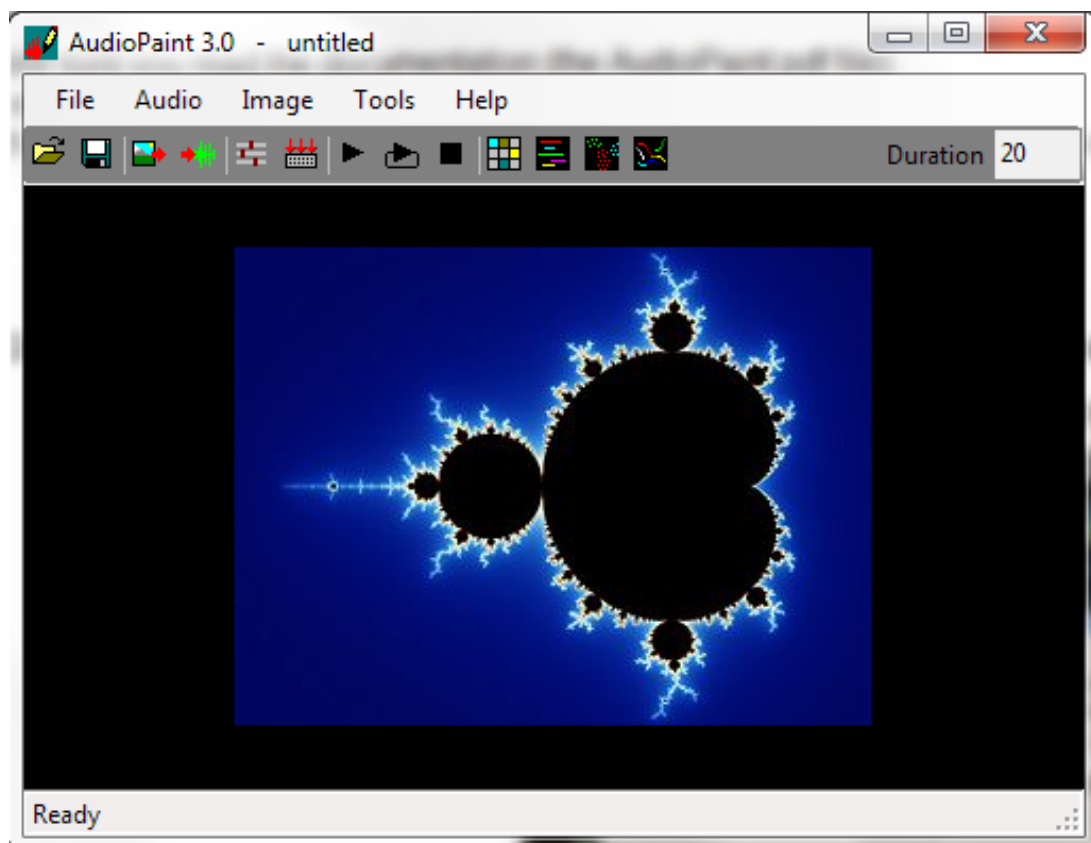


Figura 3.4: *AudioPaint* com um conjunto de Mandelbrot para ser utilizado na geração do som.

3.2.3 Mandelbrot Composer

O terceiro *software* testado destaca-se dos demais por gerar um som mais próximo de uma música da forma que se conhece. A abordagem do *Mandelbrot Composer*⁵ é diferente: o som é gerado pela borda do fractal; cores são descartadas; a parte destacada do fractal, mostrado na Figura 3.5, é a borda que será utilizada para a composição do som, através da curvatura (grau) dos ângulos.

Apesar da abordagem da leitura pela borda ser outra análise interessante, esta não se mostra eficiente na captura de importantes características dos fractais como, por exemplo, a convergência para a raiz, que estaria além das bordas do fractal. Além disso, este *software* é específico para o conjunto de Mandelbrot, não permitindo a carga de outra imagem para geração do som.

⁵<http://www.asahi-net.or.jp/~HB9T-KTD/music/English/Soft/fractal.html>

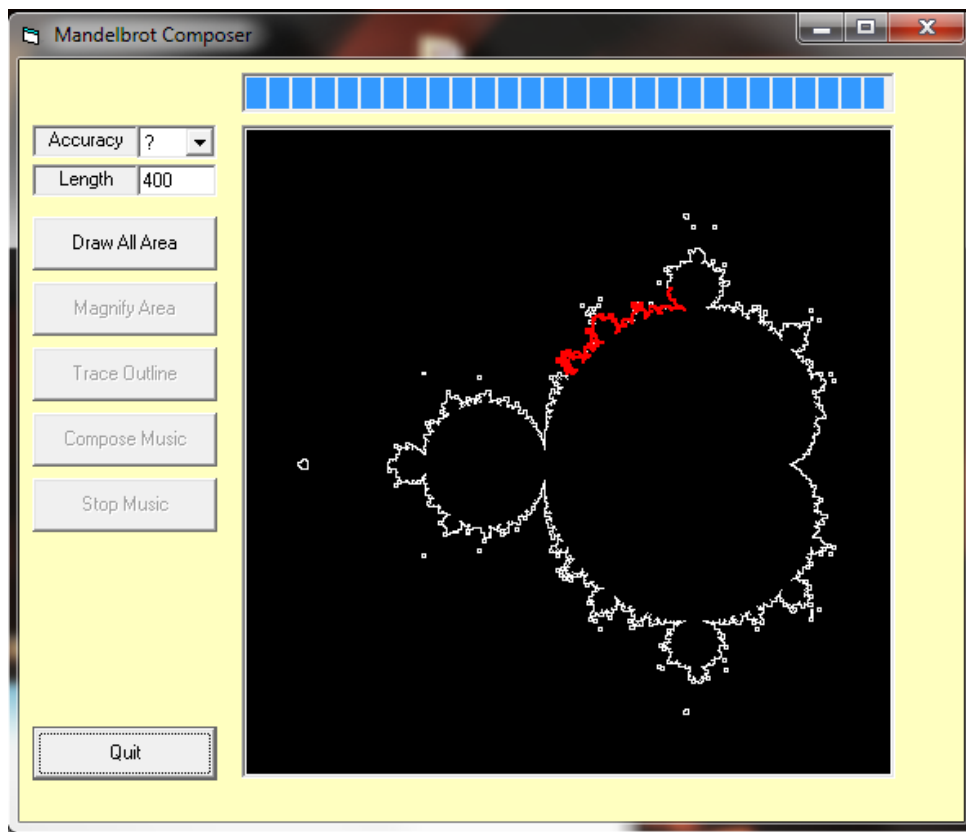


Figura 3.5: A interface do *Mandelbrot Composer*.

3.3 Considerações sobre as ferramentas analisadas

Além das características específicas de cada *software*, observa-se em comum entre o *Photosounder* e o *AudioPaint* a característica de que estes geram sons em todas as frequências, sendo ou não notas musicais. Desta maneira, não é possível a obtenção de uma música fractal propriamente dita e sim de uma música como combinação de sons e ruídos, não de uma forma que possa interessar neste trabalho. Outro fator relevante é que os *softwares* analisados são de código fechado, ou seja, não são possíveis modificações ou adaptações e, no caso do *Photosounder*, a versão completa do *software* é paga.

Todavia, os *softwares* analisados deixaram a lacuna para o estudo da interpretação do fractal através de suas cores.

O estudo da relação entre cores e música foi feito, juntamente com um protótipo para validar - de maneira trivial - esta abordagem. O estudo e a implementação do protótipo constam nos Apêndices A e B deste trabalho. Apesar de interessantes, os resultados obtidos a partir da utilização do protótipo não foram úteis pois um mesmo fractal poderia ser gerado com cores

diferentes, de modo a obter músicas diferentes, descaracterizando a função geradora de notas musicais.

Para gerar algo que melhor represente as características matemáticas do fractal com relação ao comportamento do algoritmo numérico, seguiu-se em busca de outra abordagem. O código do *software Newton Fractal* foi estudado e confirmou-se a viabilidade da sua extensão: o código fonte foi adaptado para disponibilizar a interface necessária e foi criado um módulo de geração sonora da complexidade do algoritmo.

4 *Representação sonora do método de Newton*

O objetivo deste Capítulo é apresentar o desenvolvimento do *software* que foi implementado para a geração da representação sonora da complexidade do algoritmo numérico. Para tanto, são apresentados os mapeamentos definidos, a descrição das funcionalidades do *software*, os diagramas de casos de uso, de classes e de atividades, bem como os resultados dos testes executados.

Uma vez constatada a inviabilidade de uso dos *softwares* analisados para geração de música fractal, optou-se por implementar um *software* que buscasse contemplar as características matemáticas e possibilite a análise da complexidade do algoritmo numérico através do som.

Antes de descrever a implementação, é importante considerar os seguintes fatores: a representação sonora, neste trabalho, não é focada em fazer uma música "agradável" ou "desagradável", independente da cultura, e sim analisar a diferença entre os sons e ruídos gerados por problemas instáveis ou estáveis e descobrir um padrão no som resultante. Além disso, o *software* desenvolvido teve sua trajetória modificada, não refletindo totalmente o que foi definido na disciplina de TCI porque foi encontrada uma abordagem que representa melhor as características buscadas nos fractais. Antes, a imagem fractal propriamente dita seria interpretada, trabalhando com matemática e computação gráfica, exclusivamente pela associação entre cores e sons; agora, os resultados dos cálculos, durante a execução do algoritmo, são mapeados para interpretação do fractal.

O *software* desenvolvido neste Trabalho de Conclusão é denominado *FractalNoise* e consiste em um módulo adicionado ao *NewtonFractal* de Artur Freitas (FREITAS, 2011).

A estratégia adotada, para o desenvolvimento da solução, foi assim definida: gerar a representação fractal da solução da equação utilizando o *Newton Fractal*; carregar a representação no *software* desenvolvido; gerar a música fractal. Para isto, o *FractalNoise* estará intrinsecamente ligado ao *NewtonFractal*, permitindo que os valores digitados na configuração do segundo sejam recuperados pelo primeiro. Assim, tem-se as funcionalidades implementadas disponíveis, para a análise sonora do algoritmo, logo após gerado o fractal correspondente. Como resultado, a música fractal gerada deverá expressar informações sobre a estabilidade do algoritmo.

Para compreender melhor as funcionalidades implementadas, são descritos os mapeamentos utilizados no desenvolvimento.

4.1 Mapeamento tela x plano

As dimensões da tela e limites do plano são configuráveis no *NewtonFractal* e recebidos pelo *FractalNoise*. No *FractalNoise* é possível clicar em qualquer ponto da área onde o fractal está desenhado e o ponto de tela é convertido para um número complexo.

A área da tela é definida pelo quadrado cuja diagonal é o segmento de reta de extremos $(0, 0)$ e $(500, 500)$, por exemplo, assim como apresentado na Figura 4.1. Nesta área é possível obter a correspondência entre um ponto na tela e um ponto no plano cartesiano cuja origem $(0,0)$ está localizada no centro da tela, conforme mostrado na mesma Figura 4.1 para um plano 2×2 .

4.2 Mapeamento ponto do plano cartesiano x nota musical

Após gerado o ponto no plano, este é associado a uma nota musical. A mesma associação é feita para os valores da sequência iterativa obtida através da aplicação do método de Newton. Esta associação é mostrada a seguir.

Como as doze notas musicais (inclusos semi-tons) são repetidas em cada oitava, sendo que estas notas podem ou não ser audíveis ao ouvido humano de acordo com suas frequências, optou-se por utilizar as notas de um piano acústico - como o da Figura 2.12 - onde todas as notas são audíveis, iniciando pelo *Lá* na oitava -1 até o *Dó* da oitava de número 7. Para padronizar a quantidade de notas no mapeamento, foram utilizadas somente oitenta e quatro das oitenta e



Figura 4.1: Mapeamento de um ponto na tela (500 x 500) para um ponto no plano (2 x 2)

oito notas do piano acústico, ou seja, um total de notas divisível pela quantidade de notas em cada oitava, resultando na utilização de sete oitavas.

Como o plano é dividido em quatro quadrantes, o mapeamento foi feito para cada um deles de forma espelhada. O mapeamento implementado funciona da seguinte forma: as oitavas são representadas no eixo vertical (Y), enquanto as doze notas são representadas no eixo horizontal (X). Assim, a nota em uma posição X qualquer terá uma nota com o dobro da frequência (próxima oitava) ao ser acrescida de uma unidade no eixo Y. De forma análoga, um *Dó* qualquer no eixo Y terá um *Ré* ao seu lado se deslocada uma unidade no eixo X.

Assim, considerando a Tabela 2.2 onde constam os *MIDI numbers* das notas de cada oitava, a Figura 2.12 que ilustra a relação das notas *MIDIs* com as notas de um piano acústico e o mapeamento ponto cartesiano x nota musical, tem-se a primeira nota com o *MIDI number* 21 no ponto (0,0) do plano cartesiano. A nota no ponto máximo (x, y) será o *MIDI number* 104. A Figura 4.2 mostra a distribuição das notas no plano.

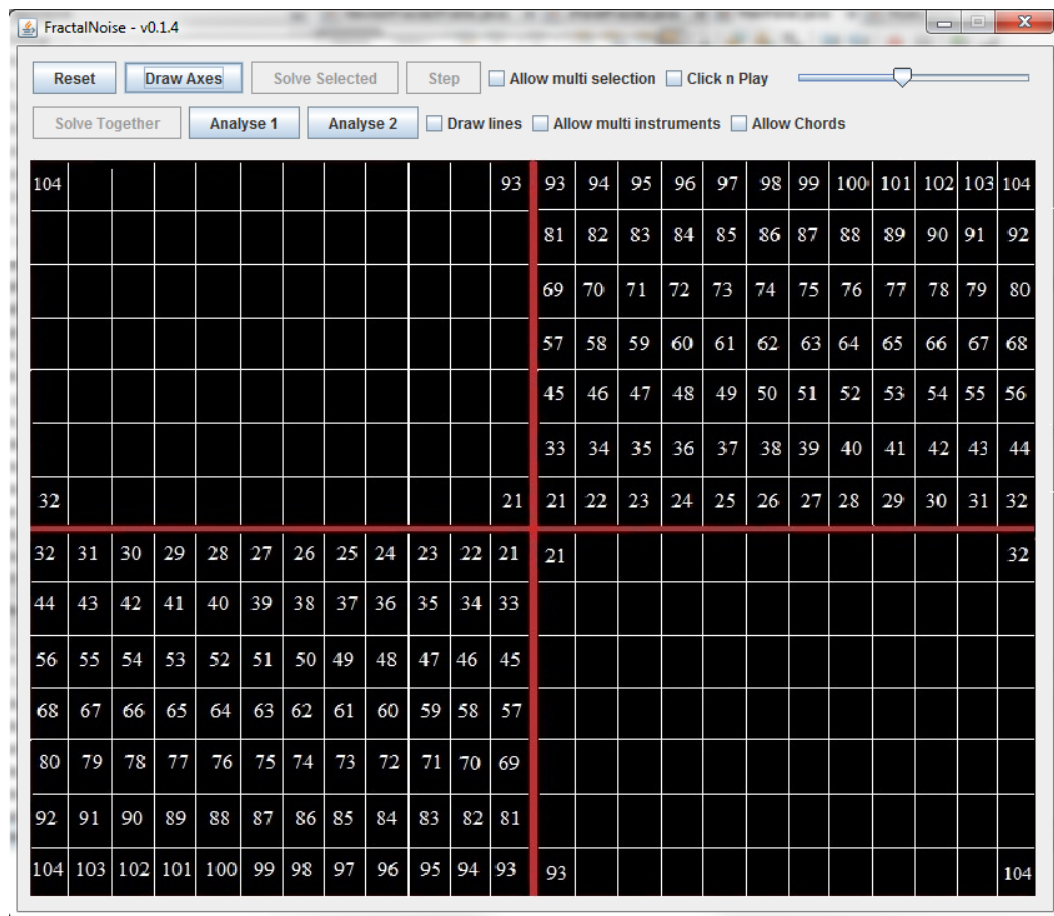


Figura 4.2: Mapeamento das notas musicais no plano de números complexos.

Além deste mapeamento, há uma extensão quando ativada a opção *Allow Chords*, modificando a nota inicial de cada quadrante. A funcionalidade estritamente de cunho musical deixa a música mais harmônica através do uso de acordes: a combinação de três ou mais sons simultâneos diferentes (SANTOS, 2009).

A implementação consiste em "afinar" os quadrantes através de um acorde maior. Tomando a primeira nota do quadrante superior direito como tônica (inicial), é aplicado um deslocamento de quatro semitons (intervalos) para gerar nota inicial do segundo quadrante (superior esquerdo, no sentido anti-horário). Para gerar a primeira nota do terceiro quadrante (inferior esquerdo), um deslocamento de três semitons é aplicado com base na primeira do segundo quadrante. Por fim, no quarto quadrante (inferior direito), um deslocamento de quatro semitons é aplicado à inicial do terceiro quadrante.

Como resultado, não são produzidos sons simultâneos na mesma posição do plano em diferentes quadrantes. Por exemplo, não irá ocorrer uma repetição de quatro notas *Dó*, e sim as

notas *Dó*, *Mi*, *Sol* e *Si*. O mapeamento do plano, nesta funcionalidade, é apresentado na Figura 4.3.

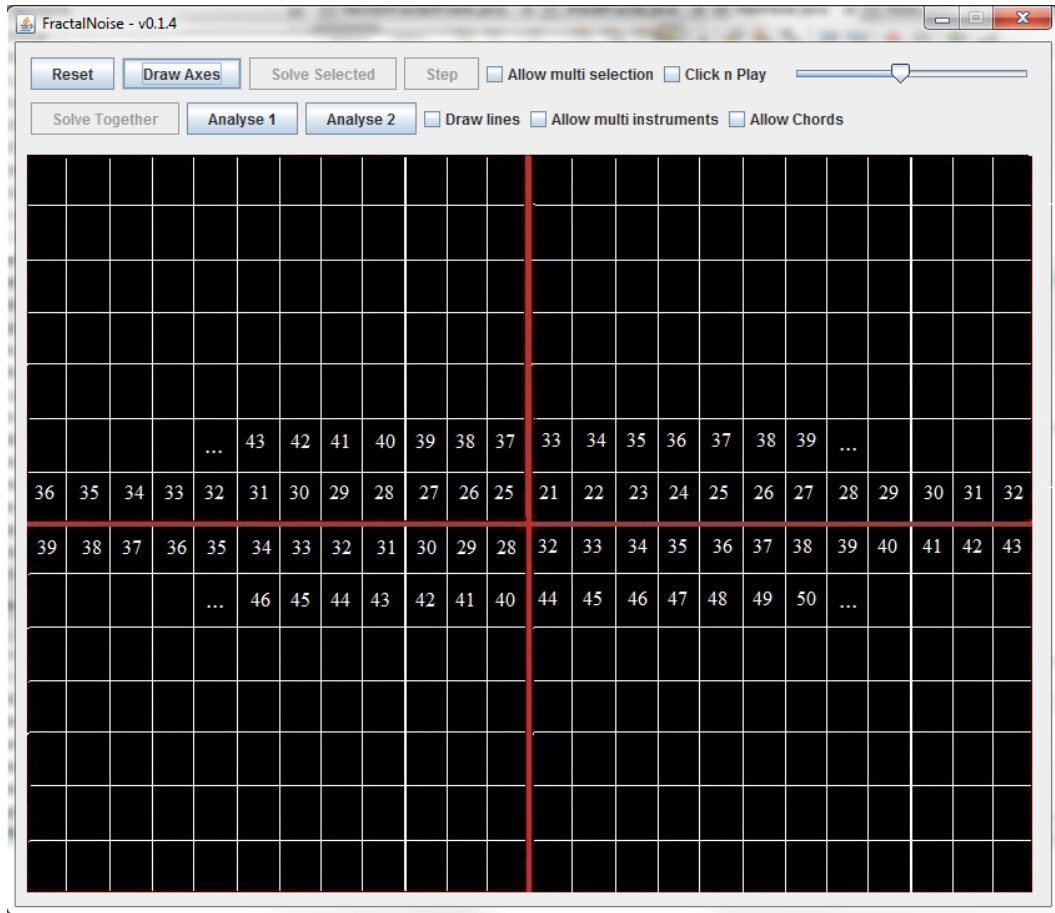


Figura 4.3: Mapeamento das notas musicais no plano com a opção *Allow Chords* ativa.

4.3 Cálculo do tempo

Como os mapeamentos são definidos pelo plano e as notas são pontos neste plano, foi considerada a distância entre dois pontos como a definição do tempo entre as notas musicais. Em outras palavras, o tempo de execução de uma nota corresponderá à sua distância no plano até o ponto calculado pela próxima iteração do método de Newton. O cálculo utilizado para obtenção da distância entre dois pontos no plano é mostrado na Figura 4.4 e definido por:

$$d_{AB} = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2} \quad (4.1)$$

Uma vez definidos os mapeamentos, é possível avançar à descrição das funcionalidades utilizadas nas análises desenvolvidas no *FractalNoise*. As opções adicionais que podem ser

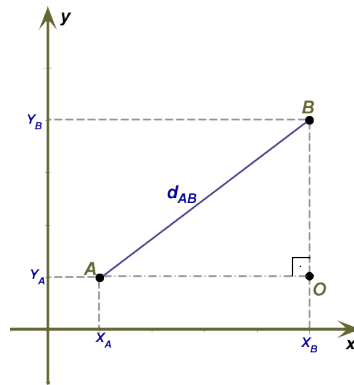


Figura 4.4: A hipotenusa AB representando a distância entre os pontos A e B

Fonte: <http://www.brasilescola.com/upload/conteudo/images/distanc1.png>

utilizadas para customizar estas funcionalidades são descritas ao final desta Seção.

4.4 Descrição do sistema

O *software FractalNoise* tem por objetivo gerar sons que possibilitem avaliar a estabilidade do algoritmo numérico. Para isto, é informada uma equação e são configurados os parâmetros no *NewtonFractal*. A seguir é desenhado o fractal que corresponde à resolução desta equação através do *NewtonFractal* e gerado o som correspondente pelo *FractalNoise*. A opção para utilizar o *FractalNoise* e a sua interface são mostradas, respectivamente, na Figura 4.5 e Figura 4.6.

As equações utilizadas durante o desenvolvimento do *FractalNoise* correspondem à bateria de testes apresentada na Tabela 4.1.

Para execução do *NewtonFractal* e do *FractalNoise*, é necessária a seguinte infraestrutura de *hardware* e *software*:

- processador *quad core* de 1.5 GHz;
- memória RAM de 1GB;
- placa de vídeo de 512MB;
- interpretador Java JRE 1.6+;
- ambiente de computação matemática *Octave*;

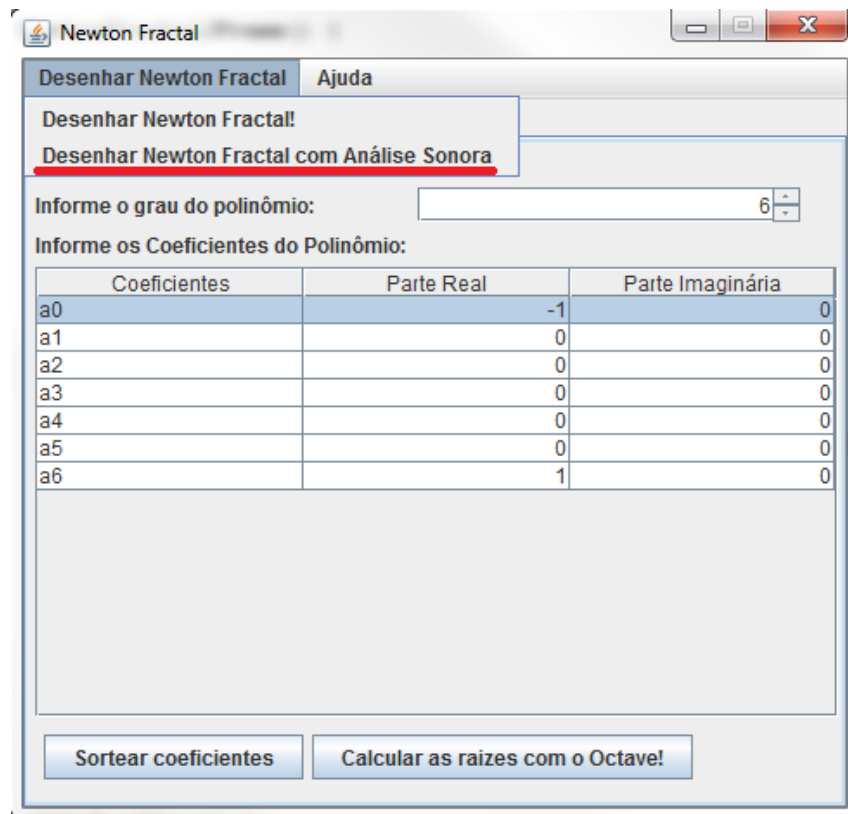


Figura 4.5: A opção para utilizar o *FractalNoise* no *NewtonFractal* configurado para resolução de $f(z) = z^6 - 1$.

Problema	Equação	Raízes
Raiz real simples Raiz complexa par conjugado	$f(z) = z^3 - 2$	$r_1 = 1,25992$ $r_2 = -0,629961 - 1,09112i$ $r_3 = -0,629961 + 1,09112i$
Raiz real simples Raiz complexa par conjugado	$f(z) = z^6 + z^4 + z^2 - 1$	$r_1 = 0,737353$ $r_2 \approx -0,737365$ $r_3, r_4, r_5, r_6 \in C$
Raiz real simples Raiz complexa par conjugado	$f(z) = z^6 + z^3 - 1$	$r_1 = -0,85205078$ $r_2 = 1,1739501953$ $r_3, r_4, r_5, r_6 \in C$
Raiz real simples Raiz complexa par conjugado	$f(z) = z^6 + z^2 - 1$	$r_1 = -0,82603113$ $r_2 = 0,82603670$ $r_3, r_4, r_5, r_6 \in C$
Raiz real simples	$f(z) = (x-1)(x+2)(x-3)$	$r_1 = 1$ $r_2 = -2$ $r_3 = 3$
Raiz real simples Raiz complexa par conjugado	$f(z) = z^6 + z^4 + 2z^2 - 1$	$r_1 = 0,627$ $r_2 = -0,627$ $r_3, r_4, r_5, r_6 \in C$
Raiz real simples Raiz complexa par conjugado	$f(z) = z^6 - 1$	$r_1 = 1$ $r_2 = -1$ $r_3, r_4, r_5, r_6 \in C$

Tabela 4.1: A bateria de teste utilizada durante o trabalho

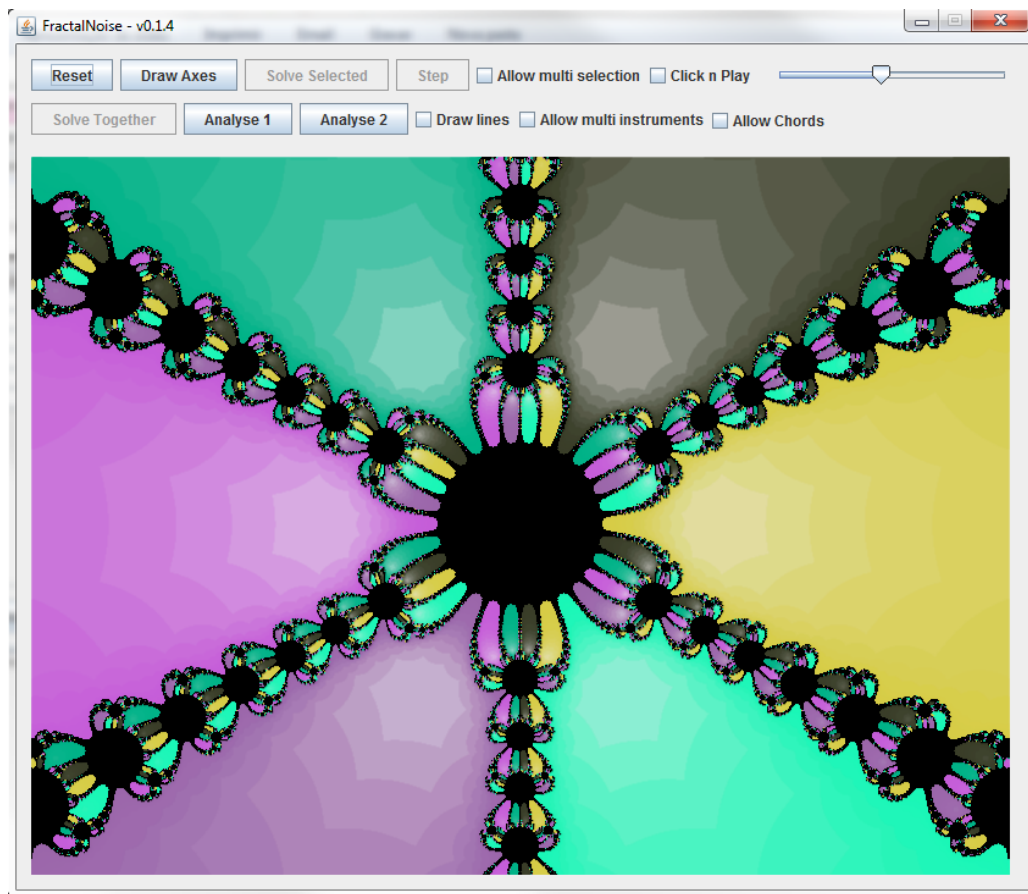


Figura 4.6: A tela do *FractalNoise* quando carregado o fractal correspondente à equação $f(z) = z^6 - 1$.

As funcionalidades do *FractalNoise* são descritas a seguir, sendo cada item uma funcionalidade do *software*. As funcionalidades podem ser combinadas, por exemplo, para a análise de uma sequência específica ou de várias sequências simultâneas.

4.4.1 Análises dinâmicas

Para iniciar uma análise, o *FractalNoise* apresenta o desenho do fractal e possibilita que um ponto seja escolhido. Uma vez selecionado, o ponto é convertido para o ponto correspondente no plano de números complexos e considerado como ponto inicial. O *FractalNoise*, então, possibilita as seguintes análises:

- *Solve Step* - o ponto selecionado é usado como valor inicial do método de Newton. Ao clicar nesta opção, uma iteração é executada através do método, conforme mostrado na Seção 2.1.1. O ponto calculado é desenhado na tela. O som correspondente ao ponto é gerado pelo mapeamento do ponto do plano para a nota musical e tocado por um tempo padrão proveniente da duração natural da nota no *MIDI System*. Este procedimento pode

se repetir até que a diferença entre os dois valores adjacentes seja menor ou igual à 10^{-5} . Apesar de este valor ser configurável no *NewtonFractal*, o método de Newton pode calcular valores repetidos e, por isso, é permitido que sejam executadas iterações até que o desenho dos pontos calculados se sobreponha, de forma a não ser mais possível a diferenciação visual. Uma vez atingido o critério de parada, o botão é desabilitado e a análise concluída.

- *Solve Selected* - de forma análoga ao *Solve Step*, deve-se selecionar um ponto para iniciar a análise. Uma vez selecionado, ao clicar nesta opção, a resolução pelo método de Newton é iniciada pelo ponto escolhido e segue automaticamente até que a diferença entre os dois valores calculados seja menor ou igual à 10^{-5} . Cada iteração calculada desenha o ponto resultante no fractal e reproduz a nota correspondente. Como esta é uma análise contínua, a distância entre os pontos define o tempo durante o qual a nota é tocada.

É possível, também, a marcação de mais de um ponto inicial para as análises. Assim, habilitam-se as seguintes opções:

- *Solve Selected* - de forma semelhante à análise executada para um único ponto através da opção *Solve Selected*, esta opção possibilita a seleção de mais de um ponto. A diferença é que os pontos selecionados são analisados em sequência, com um intervalo de dois segundos entre cada análise. Em outras palavras, é criada uma fila de pontos iniciais que serão analisados. O método de Newton é aplicado ao primeiro ponto até que o critério de parada exatidão seja atingido. O intervalo é aguardado e o processo se repete para o próximo ponto selecionado.
- *Solve Together* - após selecionados os pontos, todos são armazenados em uma matriz, um por linha, na primeira coluna. Cada ponto é utilizado na geração da sequência de valores até que a condição de parada seja satisfeita, ou seja, exatidão menor ou igual à 10^{-5} . Os valores obtidos são colocados na linha correspondente ao ponto inicial. Ao final, tem-se uma matriz cujas linhas representam os pontos iniciais e as sequências associadas geradas pelo método de Newton. A seguir, a análise dos pontos se inicia através da leitura coluna por coluna desta matriz, ou seja, o primeiro ciclo analisará os pontos iniciais escolhidos pelo usuário, na ordem de clicagem; o segundo ciclo analisará a primeira iteração de cada um desses pontos; o terceiro ciclo, a segunda iteração e, assim, sucessivamente. Com uma visão de mais alto nível, esta análise intercala a resolução de uma iteração para cada um dos pontos iniciais selecionados. O processo de geração das notas mantém o

mesmo mapeamento das demais análises, bem como o tempo entre os pontos que permanecem calculados pela distância entre os pontos de cada iteração (da sua iteração, o ponto adjacente lateral na mesma linha da matriz).

Como estas análises são customizáveis, é importante ressaltar que, nas áreas de cor preta no fractal, por definição do *NewtonFractal*, estão aqueles pontos que não geram sequências convergentes ou geram sequências que não observam a condição de parada exatidão quando assumidos como valor inicial. Tais pontos, portanto, não refletirão as características buscadas no fractal, ou seja, não tornam possível o acompanhamento da sequência iterativa através de suas representações sonora e gráfica.

As análises dinâmicas permitem capturar propriedades em partes específicas dos fractais. Pode-se, por exemplo, estudar o comportamento de uma área próxima a uma bacia de convergência do fractal ou de uma área onde o fractal é menos uniforme. Estas análises são usadas em abordagens mais específicas e mostram características detalhadas para cada região do fractal.

4.4.2 Análises estáticas

As abordagens estáticas permitem trabalhar com um conceito mais voltado à música que representa o fractal, ou seja, o mesmo fractal sempre resultará na mesma música, uma vez que a análise parte de pontos pré-determinados.

Foram implementadas duas formas de análises estáticas que diferem, somente, no método de geração do som.

Os pontos iniciais são pré-estabelecidos de acordo com a divisão da área de desenho do fractal em quatro quadrantes. Cada quadrante recebe um ponto em seu centro, conforme Figura 4.7. A distribuição dos pontos ocorre em sentido anti-horário, iniciando pelo quadrante superior direito. Esta divisão dos quadrantes nada mais é que os eixos X e Y do plano, cada um representando, respectivamente, a parte real e imaginária dos números complexos que representam o fractal.

- *Analyse 1* - a primeira análise estática faz uso da funcionalidade *Solve Together*. A execução ocorre da mesma forma que a funcionalidade dinâmica, porém com os pontos iniciais pré-estabelecidos.

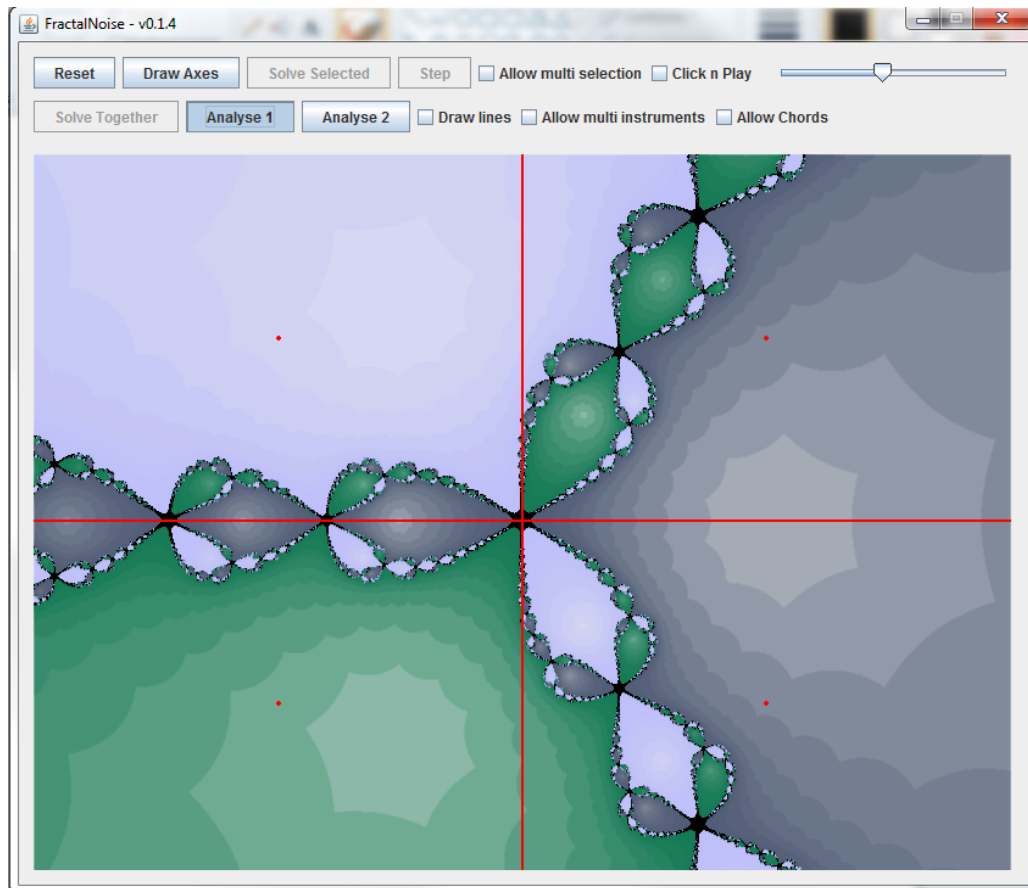


Figura 4.7: A tela do *FractalNoise* mostrando a divisão dos quadrantes com os quatro pontos pré-estabelecidos para análise, na área de desenho do fractal correspondente à equação $f(z) = z^3 - 2$.

- *Analyse 2* - a segunda análise estática difere totalmente das demais no quesito geração de música fractal: a análise necessita que notas sejam tocadas simultaneamente. Para tanto, os pontos iniciais pré-estabelecidos são resolvidos da mesma forma que na funcionalidade *Analyse 1*. Uma vez geradas todas as sequências associadas aos pontos iniciais, a interpretação da matriz resultante é iniciada. No tempo inicial (zero), todas as notas correspondentes aos pontos iniciais são tocadas pelo tempo (distância) até o próximo ponto. Na segunda iteração, cada nota será iniciada após o término da nota anterior, ou seja, se a nota da primeira iteração do ponto um durar cinco segundos, a nota da segunda iteração, deste ponto, iniciará no tempo cinco; em paralelo, se a nota da primeira iteração do ponto dois durar três segundos, a segunda iniciará no tempo três e assim para todos os pontos até o final da execução, sobrepondo os sons. Para gerar tal som com notas simultâneas, foi necessária a exportação de um arquivo *.midi*. Este arquivo é sequenciado durante a resolução do método e, ao final, é carregado e reproduzido pela *JavaSound*.

Em ambas análises estáticas também é possível o caso de um ponto que não converge - ponto preto no fractal - ser definido como ponto inicial. Como estas análises têm os pontos iniciais pré-definidos, caso um destes seja um local de não convergência, há a possibilidade de deslocamento dos pontos iniciais para que a análise seja executada. O deslocamento em 10^{-1} - valor médio estimado para sair da região de não convergência - ocorre na diagonal, no sentido exterior de cada quadrante, ou seja, $+10^{-1}$ para o eixo positivo e -10^{-1} para o eixo de sinal negativo, até que todos pontos iniciais converjam.

Tanto as análises dinâmicas quanto as estáticas permitem analisar, através do som, o comportamento da sequência de valores gerada pelo método de Newton. Com estas duas formas de análise, é possível capturar as propriedades matemáticas necessárias para responder questões como, por exemplo, o tempo de convergência até a raiz ou se a sequência gerada para um ponto inicial imediatamente ao lado pode ter um comportamento drasticamente diferente.

4.4.3 Funcionalidades extras

Como complemento das análises sonoras, aproveitando o fato do *NewtonFractal* ser de código aberto, foram implementadas funcionalidades que permitem o acompanhamento da análise de forma visual, executadas sobre o fractal gerado:

- *Draw Axes* - permite que os eixos X e Y, representando o plano de números complexos no qual o fractal está desenhado, sejam destacados, tal como na Figura 4.7.
- *Draw Lines* - funcionalidade que pode ser ativada ou desativada, ao iniciar uma análise. Permite que sejam traçados vetores entre os pontos obtidos a cada iteração, possibilitando o acompanhamento visual do caminho percorrido durante a resolução do método - do ponto inicial até a raiz.
- *Reset* - para eliminar a necessidade de nova geração do fractal, permite que, após uma análise, a tela interativa seja limpa e o fractal redesenhado para novas análises.

Já, para customizar a parte sonora, foram implementados os seguintes facilitadores:

- *Velocity* - permite o controle da velocidade de execução de uma análise. O tempo entre as notas pode ser alterado, *a priori*, pelo fator selecionado no *slider*. O mesmo fator será aplicado do início ao fim da análise, mantendo a conformidade entre os tempos.

- *Click n Play* - permite a escolha do ponto inicial da análise através do som: ao clicar em um ponto qualquer no fractal, o som correspondente àquele ponto é tocado com seu tempo natural. Caso esteja de acordo com o som buscado, a análise pode ser iniciada. Do contrário, novos sons podem ser procurados selecionando outros pontos até que o som desejado seja encontrado.
- *Multi Instruments* - funcionalidade experimental que permite que a nota tocada utilize um instrumento diferente para cada ponto inicial selecionado. Fazendo o uso da *Solve Together*, por exemplo, ter-se-ia a intercalação de instrumentos que são definidos de forma estática de acordo com o número sequencial do ponto inicial, sendo este número usado como chave na busca do instrumento no sintetizador da *JavaSound* (e.g ponto inicial um, instrumento um; ponto dois, instrumento dois).
- *Allow Chords* - Funcionalidade voltada à parte musical: permite que seja feito o uso de acordes na geração do som para obter um som menos dissonante.

4.5 Diagrama de casos de uso

O diagrama de casos de uso da Figura 4.8 mostra as interações do usuário com o *Fractal-Noise* através das funcionalidades já descritas.

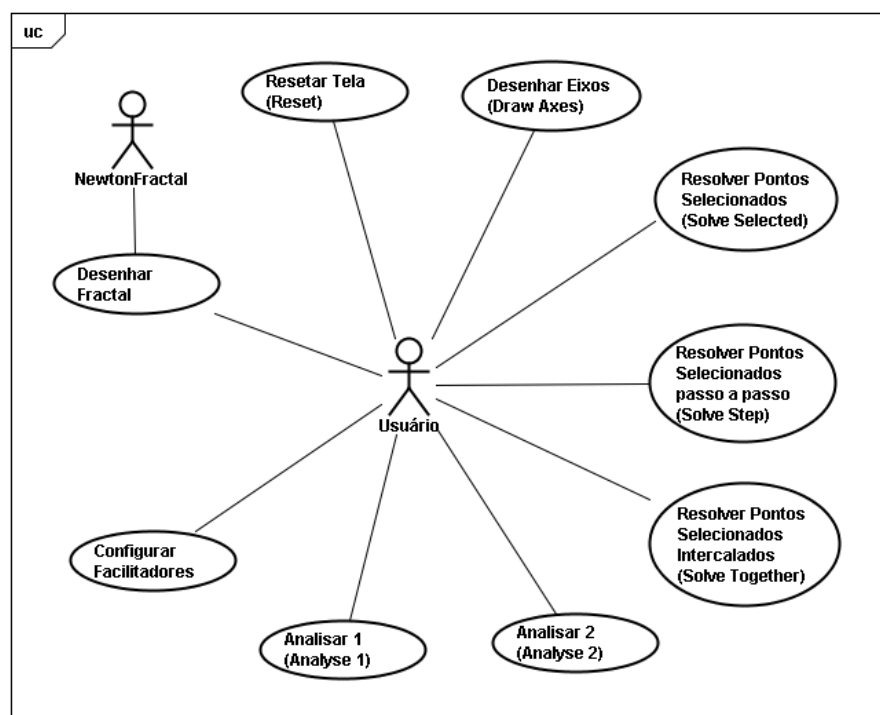


Figura 4.8: O diagrama de casos de uso do *FractalNoise*.

4.6 Diagrama de classes

Para utilizar o código do *NewtonFractal*, foram necessárias modificações nas classes que interagem com o *FractalNoise*. Tais classes modificadas do *NewtonFractal* são apresentadas na Figura 4.9.

A classe modificada *JPanelFractal* foi acrescida dos métodos responsáveis pelo desenho dos pontos, vetores, linhas e eixos. Métodos para recuperação de informações - como a quantidade máxima de iterações - também foram adicionados.

Na classe *NewtonSolver*, foi criado o método responsável pela resolução de uma única iteração do método de Newton, baseado na implementação já existente, porém passo a passo.

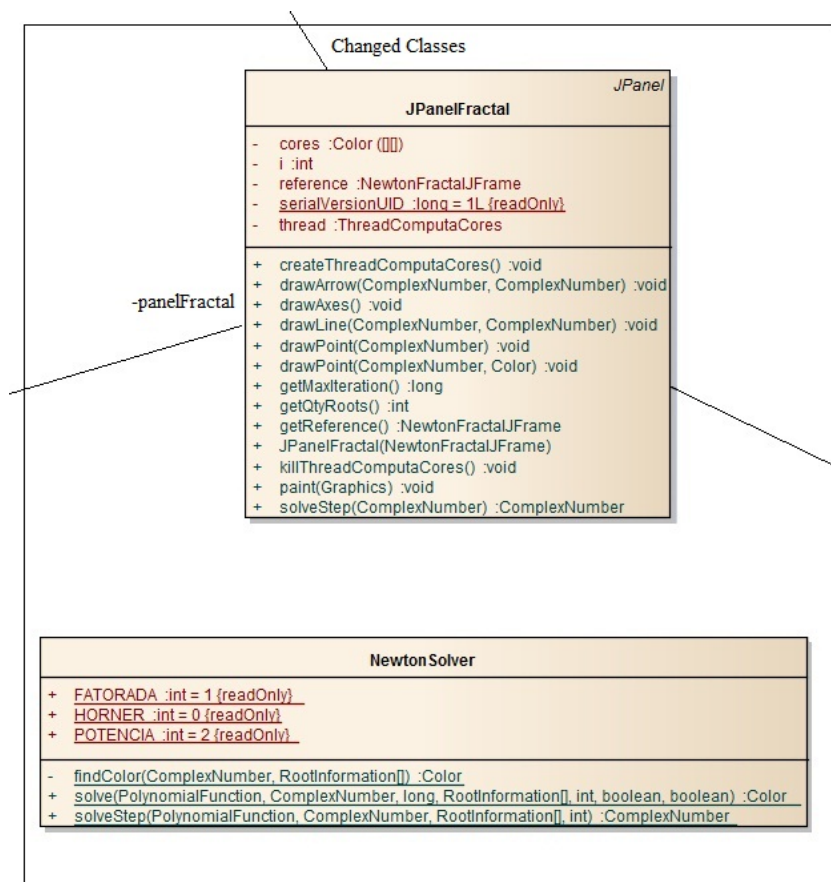


Figura 4.9: As classes do *NewtonFractal* alteradas para utilização no *FractalNoise*

Já, na implementação do *FractalNoise*, foram criadas as classes *MainPanel*, *Music*, *MIDI-Sequencer* e *MIDIPlayer*, que são mostradas na Figura 4.10.

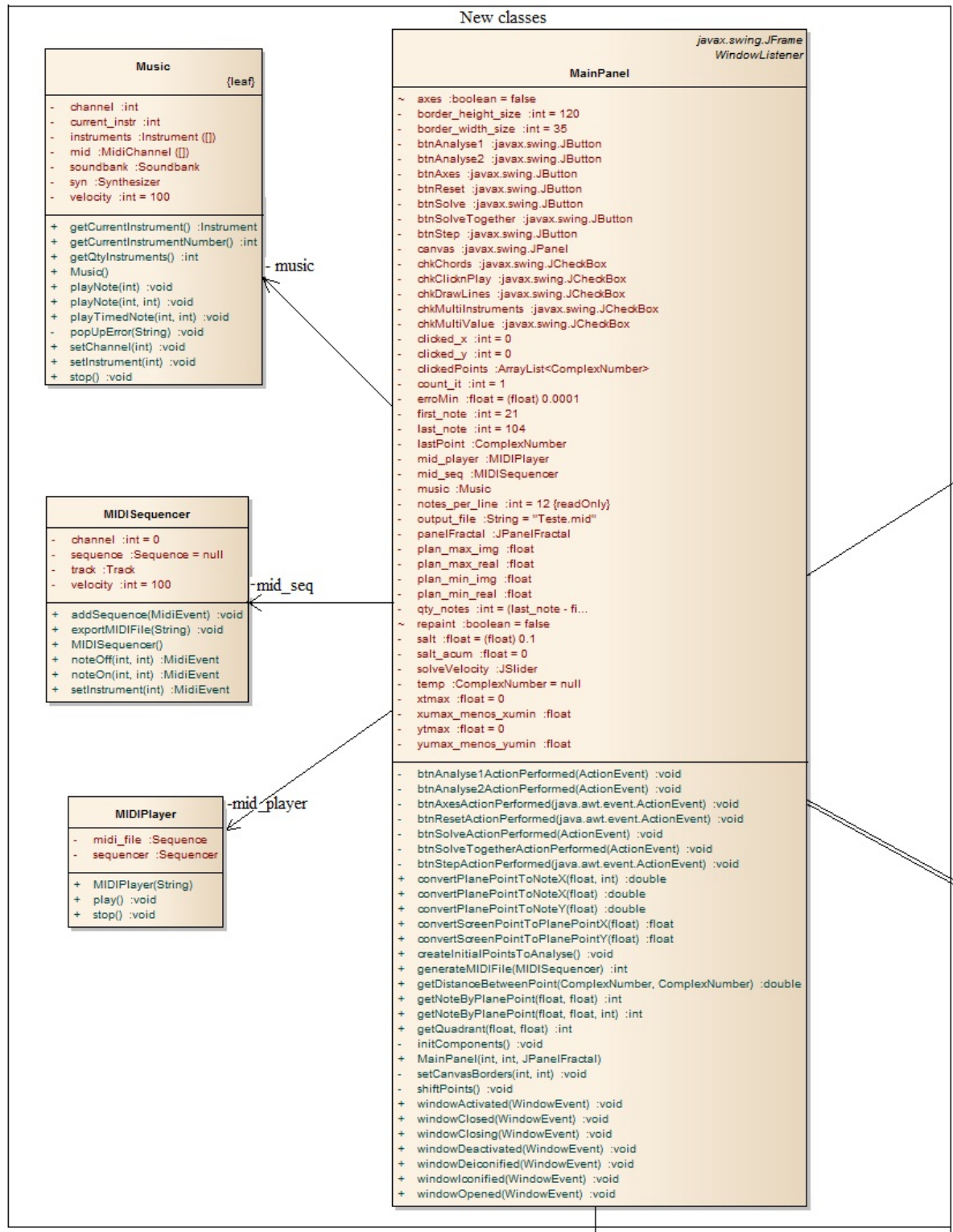


Figura 4.10: As classes criadas para implementação do *FractalNoise*

A classe *Music* é responsável pela execução das notas musicais em tempo real. Nela é possível tocar a nota pelo tempo determinado e manipular o instrumento corrente. É também responsável pela manipulação do *MIDISystem* e do sintetizador da *JavaSound*.

A classe *MIDISequencer* é encarregada da geração do arquivo *.midi*. Aqui são sequenciados os eventos recebidos - sejam estes sobre notas ou instrumentos - e o arquivo é exportado.

A classe *MIDIPlayer* é responsável pela reprodução de arquivos *.midi* e faz a carga do sequenciador que interpretará e reproduzirá tais arquivos.

Já a *MainPanel* implementa a interface gráfica do *FractalNoise*. A classe recebe uma instância do fractal desenhado pelo *NewtonFractal* e manipula os eventos executados solicitando as análises ou configurando facilitadores. Além disso, é responsável pelos cálculos que mapeiam pontos de tela em pontos de plano e pontos de plano em notas musicais, pelo cálculo da distância entre os pontos e pela atribuição dos pontos iniciais para as análises estáticas.

4.7 Diagrama de atividades

A fim de facilitar o entendimento do fluxo de execução do *FractalNoise*, foram elaborados os diagramas de atividades correspondentes a cada análise. Tais diagramas podem ser visualizados no Apêndice C, dispostos da seguinte forma: as análises dinâmicas *Solve Step*, *Solve Selected* e *Solve Together* representadas, respectivamente, pelos diagramas das Figuras C.1, C.2 e C.3. Já as análises estáticas *Analyse 1* e *Analyse 2*, são representadas pelos diagramas das Figuras C.4 e C.5, respectivamente.

4.8 Testes realizados

A seguir são apresentados os testes realizados durante a implementação do *FractalNoise*. São mostrados os fractais utilizados nas análises e suas equações correspondentes. Um *link* para o som resultante também é informado. Este *link* direciona para o *website SoundCloud*¹, escolhido como repositório dos sons gerados por ser uma ferramenta gratuita e acessível publicamente, na qual é possível ouvir o som sem baixá-lo no computador.

É importante ressaltar que, nos testes, foram utilizadas somente análises estáticas, pois estas compreendem as análises dinâmicas e garantem que o teste possa ser igualmente reproduzido, uma vez que os pontos iniciais são fixos.

¹<http://soundcloud.com/>

A Figura 4.11 representa uma análise *Analyse 1*- com instrumento único (Piano), velocidade padrão e desenho de vetores habilitado - do fractal correspondente à equação $f(z) = z^6 - 1$. O som resultante está disponível no *link* <http://soundcloud.com/nestorsalamon/fractal-music-z6-1>.

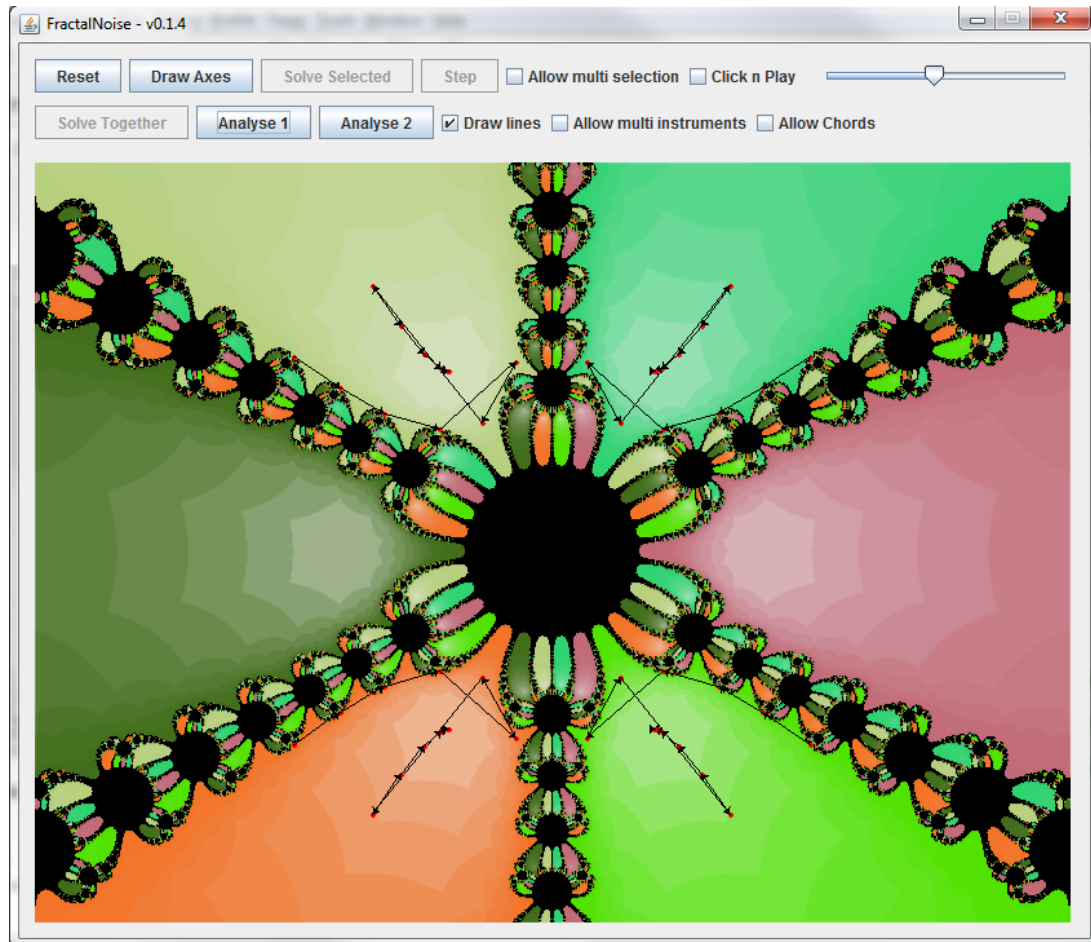


Figura 4.11: Tela do *FractalNoise* analisando por uma *Analyse 1* o fractal $f(z) = z^6 - 1$.

A Figura 4.12 representa uma análise *Analyse 1* - com múltiplos instrumentos (Piano, Electric Piano 2, Music Box e Dulcimer), velocidade padrão, desenho de vetores habilitado, deslocada inicialmente em 2×10^{-1} (uma vez que os pontos iniciais não convergem) - do fractal correspondente à equação $f(z) = z^6 + z^3 - 1$. O som resultante encontra-se no *link* <http://soundcloud.com/nestorsalamon/z6-z3-1-multi>.

A Figura 4.12 também representa uma análise *Analyse 2*- com instrumento único (Piano), desenho de vetores habilitado, deslocada inicialmente em 2×10^{-1} - do fractal correspondente à equação $f(z) = z^6 + z^3 - 1$. A velocidade do som gerado foi reduzida em cinco vezes. O som encontra-se no *link* <http://soundcloud.com/nestorsalamon/z6-z3-1-sol2>.

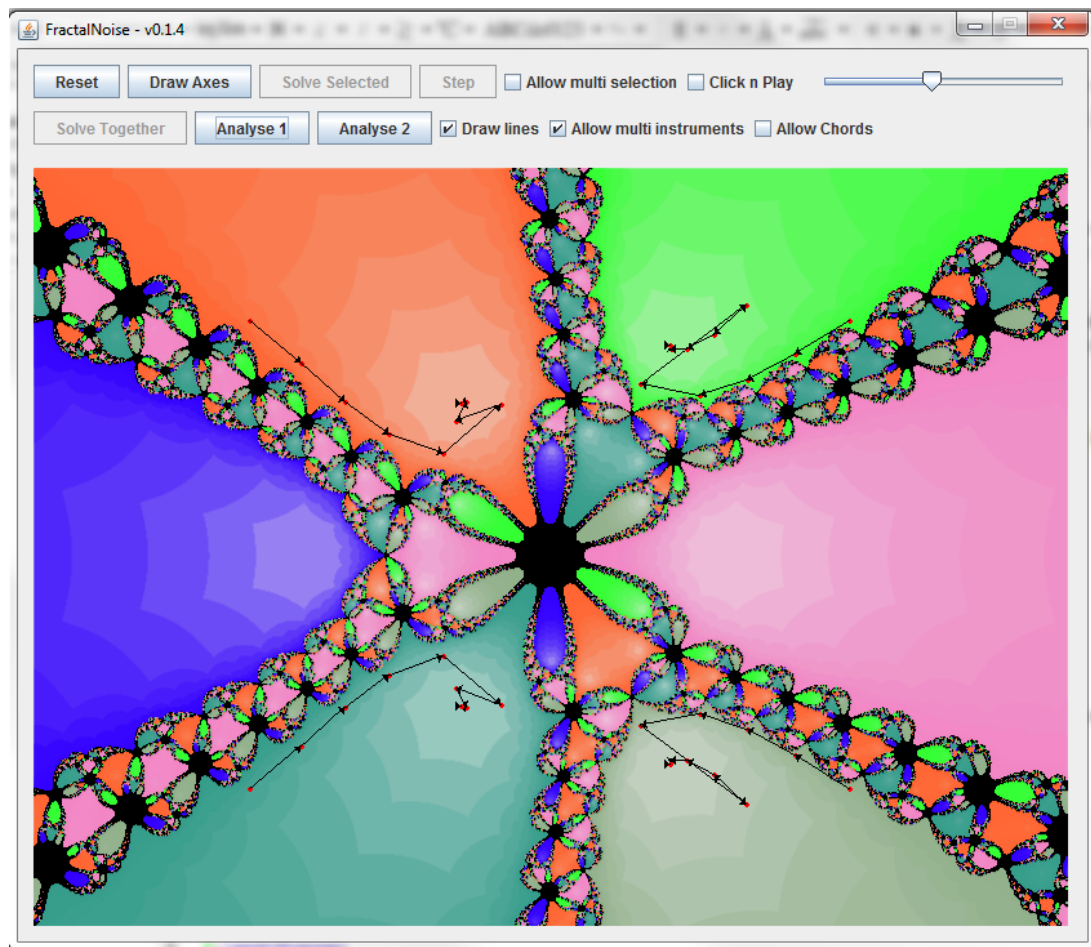


Figura 4.12: Tela do *FractalNoise* analisando o fractal da equação $f(z) = z^6 + z^3 - 1$.

5 *Análise dos resultados obtidos*

Neste Capítulo são mostrados os resultados obtidos com a utilização do *FractalNoise* nas análises da bateria de teste apresentada na Tabela 4.1, onde constam exemplos de problemas estáveis e instáveis. As informações utilizadas durante estas análises são disponibilizadas juntamente com a imagem obtida e o *link* para o som resultante.

Inicia-se pela análise do fractal correspondente à equação $f(z) = z^3 - 2$. A equação possui uma raiz real e um par conjugado de complexas. O fractal associado apresenta regiões de estabilidade mais frequentes que regiões instáveis. A análise dinâmica pelo *FractalNoise* - com múltipla seleção, velocidade padrão, desenho de vetores e eixos habilitados - é apresentada na Figura 5.1 onde quatro pontos foram selecionados em uma região visivelmente estável. O som que corresponde à análise, obtido através do *Solve Together*, está disponível no *link* <http://soundcloud.com/nestorsalamon/z3-2-res-1>.

Analisando o som gerado, é perceptível que os pontos iniciais estão contidos em uma área de maior estabilidade, já que o som resultante apresenta um comportamento constante com uma diminuição gradativa relativamente estável entre a duração das notas, confirmando o que já era sabido sobre a estabilidade da região analisada. Também é possível perceber que é obtida, no início, uma sequência de notas bem espaçadas as quais parecem, se projetadas em maior escala, estarem se aproximando de um *pink noise*, um som mais calmo e agradável. Ao se aproximar da raiz, o som se torna mais rápido, com várias frequências sendo tocadas em um menor espaço de tempo que quando projetado em maior escala, parece estar se aproximando de um *white noise*, ou seja, ficando dissonante e difícil de se ouvir distinções.

Contudo, o que acontece ao selecionar quatro pontos em regiões mais instáveis do fractal? Foram selecionados quatro pontos em regiões instáveis distintas do mesmo fractal, conforme mostrado na Figura 5.2. O som obtido nesta análise se encontra em <http://soundcloud.com/nestorsalamon/z3-2-res-2>. Aqui é perceptível um som mais espaçado, em partes

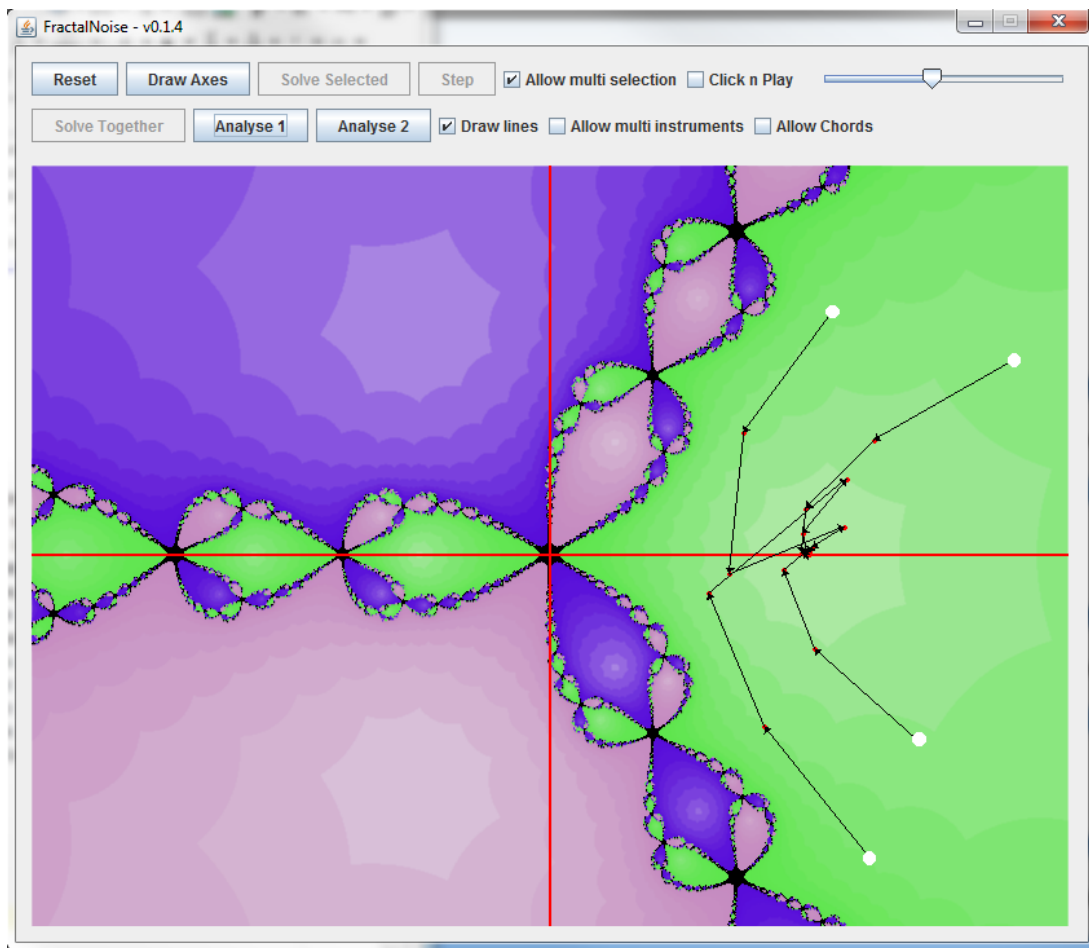


Figura 5.1: A tela do *FractalNoise* configurado para resolução de $f(z) = z^3 - 2$ nos quatro pontos iniciais destacados em regiões estáveis.

"agradável" de se ouvir como um *pink noise*. Há partes demoradas e "entediadas" tendendo a um *brown noise* e, somente ao final, sinalizando uma aproximação a um som mais "agitado", induzindo a um *white noise*.

Tomando agora o fractal da Figura 5.3 correspondente à equação $f(z) = z^3 - 2z^2 - 5z + 6$. Utilizando uma análise *Analyse 1*- com desenho de vetores e eixos habilitados, acordes desabilitados e velocidade aumentada em três vezes - gerou-se o seguinte som: <http://soundcloud.com/nestorsalamon/z3-2z2-5z-6>. É perceptível aqui um som terminando em notas com frequências bastante baixas, o que induz à conclusão de que as raízes estão próximas ao eixo horizontal, já que os sons, pelo mapeamento especificado, tendem a ficar mais agudos conforme se aproximam dos limites do plano.

Analisando especificamente o comportamento da mesma resolução, pode-se perceber uma variação de duração nas notas correspondentes ao segundo e terceiro ponto resolvido. Na se-

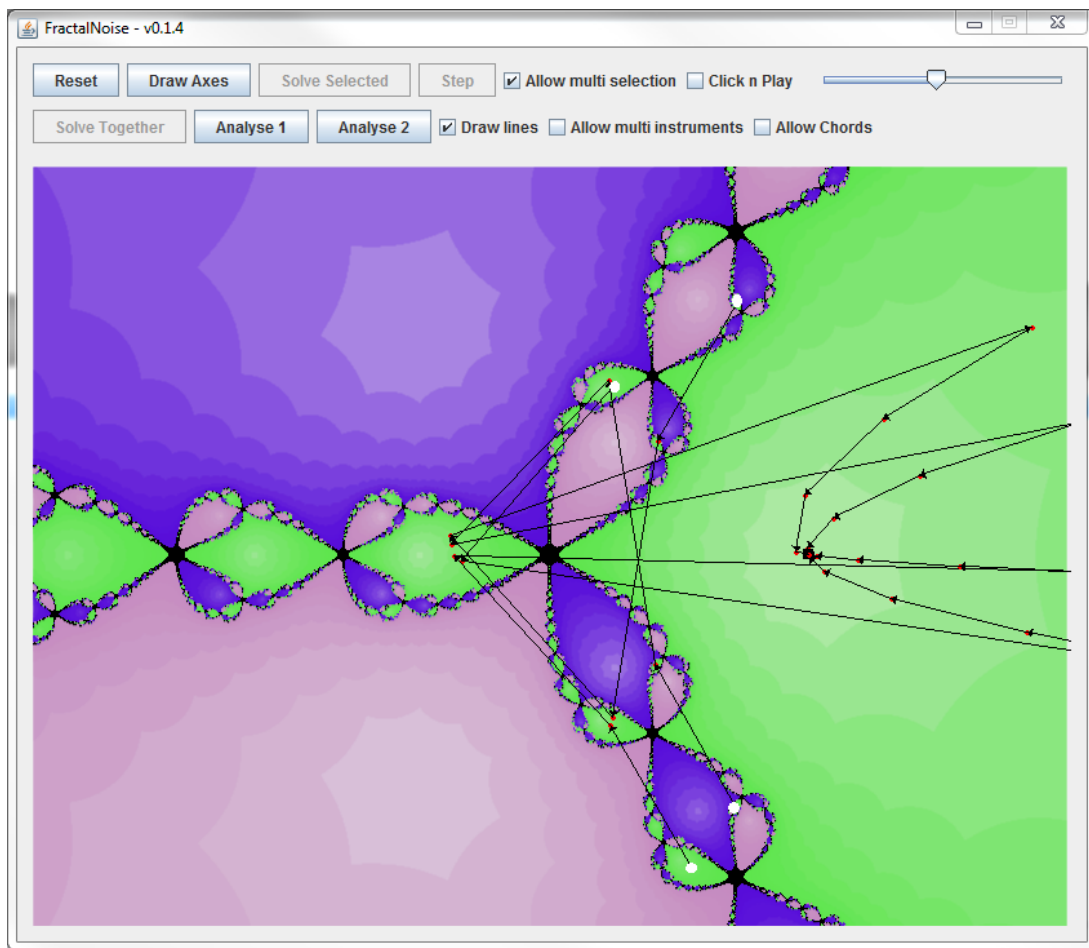


Figura 5.2: O *FractalNoise* configurado para resolução de $f(z) = z^3 - 2$ nos quatro pontos iniciais destacados em regiões instáveis.

gunda iteração destes pontos, tem-se dois sons graves e mais longos que os anteriores; na próxima iteração, é reproduzido um som mais agudo; na quarta iteração dos pontos dois e três, ambos os sons são de baixa frequência novamente, vindo a manter o padrão grave até suas respectivas raízes. Constata-se um desvio no comportamento do algoritmo para estes dois pontos iniciais, onde conclui-se que, nestes pontos, a resolução é ligeiramente instável. Ouvindo novamente o som e focando nos pontos de número um e quatro, percebe-se um comportamento padrão, uma vez que o espaçamento e a frequência das notas diminuem gradativa e constantemente, indicando uma resolução mais estável.

Visto que a análise se tornou complexa mostrando comportamentos distintos para cada par de pontos, utilizaram-se duas análises dinâmicas, uma para cada par de pontos: dois e três e um e quatro. A velocidade foi diminuída quatro vezes em relação à análise anterior. Os sons resultantes encontram-se, respectivamente, nos *links* <http://soundcloud.com/nestorsalamon/z3-2z2-5z-6-res-3-1> e <http://soundcloud.com/nestorsalamon/z3-2z2-5z-6-res-3-2>

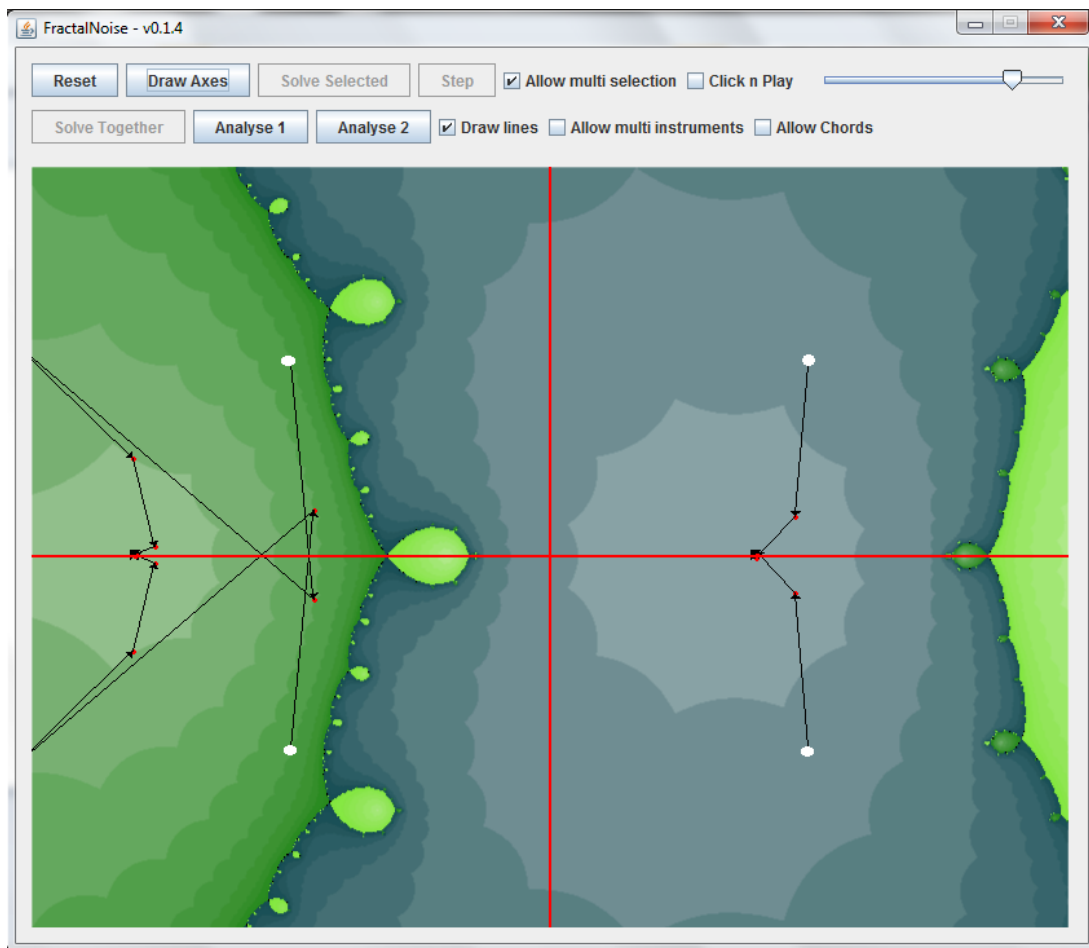


Figura 5.3: O *FractalNoise* configurado para resolução de $f(z) = z^3 - 2z^2 - 5z + 6$ através de uma análise *Analyse 1*: os pontos iniciais estão destacados.

e possibilitam a confirmação do que já fora explanado, porém de maneira mais clara. Tal caso também permite verificar a adaptabilidade do *FractalNoise*, para melhor análise dos resultados, de acordo com a necessidade específica.

Já observando o fractal correspondente à equação $f(z) = z^6 + z^4 + z^2 - 1$, através de uma análise com a funcionalidade *Solve Selected* - velocidade padrão, desenho de vetores e eixos habilitados, foram selecionados dois pontos próximos no fractal, conforme mostrado na Figura 5.4. O ponto de número um está localizado rapidamente acima do ponto de número dois. O som da resolução para os dois pontos encontra-se no *link* <http://soundcloud.com/nestorsalamon/z6-z4-z2-1-custom> - o ponto um e, após o intervalo de dois segundos, o ponto dois. Pode-se perceber, através do som, que o ponto um tem sua resolução muito mais rápida que o ponto dois, pois o som gerado tende a ser mais "agitado" logo após a segunda nota. Já o ponto dois tem sua primeira nota demorada e, em seguida, uma sequência de notas constantes até chegar em sua sequência mais "agitada". Aqui se torna perceptível que, um pequeno deslocamento no

ponto inicial, pode causar uma diferença significativa no tempo de execução do algoritmo e, então, em sua complexidade.

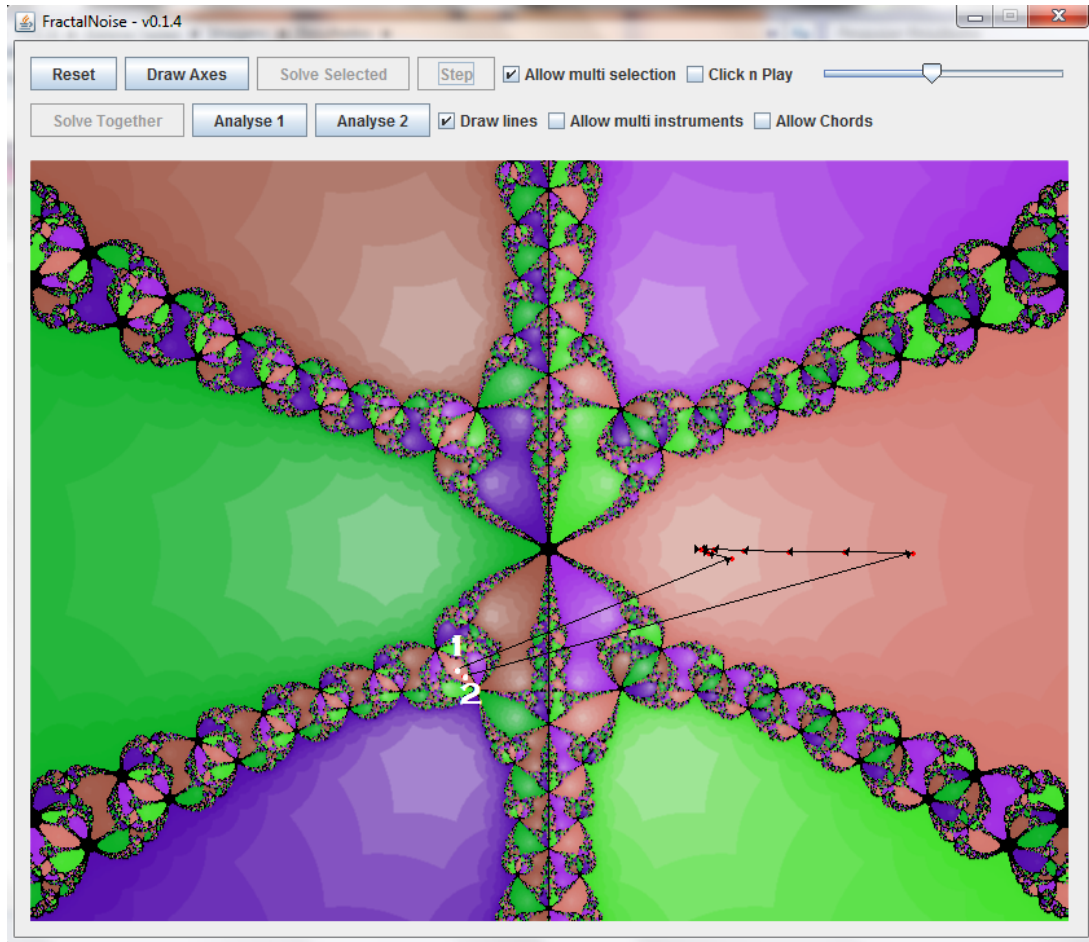


Figura 5.4: O *FractalNoise* configurado para resolução de $f(z) = z^6 + z^4 + z^2 - 1$ com uma análise *Solve Selected* para os pontos destacados.

Apesar das diferenças nestes sons, o limiar entre os mais agradáveis e mais agitados, da maneira em que se apresentam na conversão dos algoritmos, é demasiado estrito, tornando difícil a percepção exata de uma diferença. Por outro lado, aliando ao tempo (espaçamento) entre as notas, tornam-se mais perceptíveis os padrões.

Uma vez que o objetivo do método numérico é encontrar a raiz, pode-se deduzir pelas análises que, quando um som está mais "agitado", com mais frequências tocadas em um menor espaço de tempo, tem-se o melhor caso: a sequência de valores está convergindo rapidamente para a raiz. Já em casos onde o som se torna cansativo, demorado ou apresenta uma grande variação e inconstância em seu padrão, ou seja, intercala entre notas demoradas e rápidas, há o pior caso: a sequência está demorando para convergir, aproximando-se e afastando-se da raiz.

Com as análises *Analyse 1* e *Analyse 2*, é possível detectar, ainda, a simetria dos fractais através do som pela repetição das notas: com quatro pontos seleccionados, tem-se a repetição de quatro notas iguais a cada iteração para fractais simétricos horizontal e verticalmente. Já para fractais com somente uma simetria, seja esta horizontal ou vertical, há dois caminhos repetindo as mesmas notas em suas iterações. Um exemplo desta simetria horizontal e vertical é o fractal da equação $f(z) = z^6 + z^4 + 2z^2 - 1$, simétrico nos pares conjugados de raízes complexas. O fractal é mostrado na Figura 5.5, com desenho de vetores e eixos habilitados. O som disponível no *link* <http://soundcloud.com/nestorsalamon/z6-z4-2z2-1-sol2> mostra a repetição das quatro notas no decorrer da resolução, em uma *Analyse 2* com velocidade diminuída em cinco vezes e acordes desabilitados. Com o mesmo fractal, obtém-se um resultado bastante interessante com relação à simetria se forem habilitados acordes: repetindo a mesma análise com esta opção habilitada, obteve-se o som mais harmônico, disponível no *link* <http://soundcloud.com/nestorsalamon/z6-z4-2z2-1-sol2-chords>

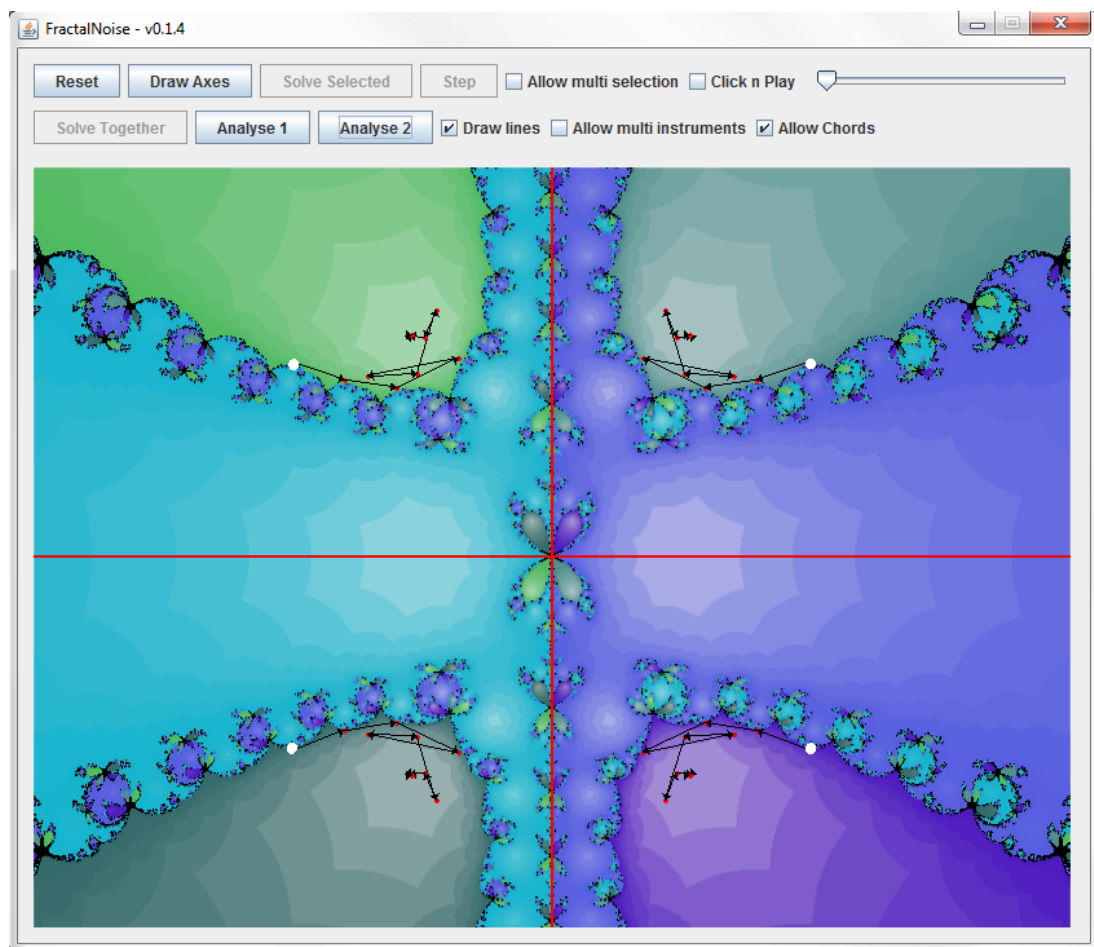


Figura 5.5: O *FractalNoise* configurado para resolução de $f(z) = z^6 + z^4 + 2z^2 - 1$ através de uma *Analyse 2*.

Outra característica interessante é que, tendo conhecimento da sonoridade das notas musicais, pode-se obter informações sobre a trajetória definida pela sequência iterativa através do som resultante a cada nota, pois estas têm suas posições específicas em cada quadrante no plano. Sabendo-se o quadrante inicial, levando em conta a duração da nota, o caminho pode ser desenhado e colorido, fornecendo uma ideia aproximada do desenho do fractal. Assim, análises que começam e terminam na mesma nota - com acordes desabilitados - convergem para a mesma raiz, para um par de raiz simétricas ou para raízes demasiado próximas.

Com um enfoque somente na representação sonora, é possível analisar o som do fractal da equação $f(z) = z^6 + z^2 - 1$ - que resulta em um fractal simétrico, através de uma análise *Analyse 2*, com acordes habilitados, velocidade diminuída em cinco vezes e múltiplos instrumentos. O som está disponível no *link* <http://soundcloud.com/nestorsalamon/z6-z2-1-sol2-multi-chords>. Tal análise permite a apreciação de um som diferente, que faz relação entre quatro instrumentos musicais (Piano, Electric Piano 2, Music Box e Dulcimer), através de acordes, mostrando que é possível a obtenção de resultados interessantes aplicando o *FractalNoise* em pesquisas voltadas à composição musical.

As análises dinâmicas permitem capturar propriedades específicas de um fractal enquanto artefato matemático e as análises estáticas possibilitam identificar propriedades que relacionam dois ou mais fractais. Além destas análises, os facilitadores permitem a customização de detalhes para acompanhamento visual ou de melhorias na geração sonora. Tudo isso fará com que seja gerada, de maneira a capturar o comportamento do método de Newton para a determinada entrada, a representação sonora da complexidade do algoritmo numérico.

6 *Conclusão*

Matemática e música estão interligadas, seja através de um conjunto de sons que formam acordes obtidos por operações matemáticas para geração de estruturas musicais ou através da razão entre as alturas das notas para afinar instrumentos.

Computação é, por si, matemática. Então, por que não utilizar a música para realizar análises na computação? Será que é possível analisar a instabilidade da resolução computacional de problemas matemáticos através da estética da música?

Este Trabalho de Conclusão buscou apresentar resultados neste sentido, ou seja, utilizar a representação sonora para compreender a instabilidade do problema matemático. Para tanto foi projetado o *software FractalNoise*.

Tal *software* foi desenvolvido em um contexto que utiliza ambiente matemático do *Octave* e bibliotecas numéricas escritas em linguagem *Java*. A escolha do ambiente de desenvolvimento se justifica pelo fato do *Octave* ser o clone público do *Matlab* e, portanto, ser adequado para a computação científica sendo utilizado sem maiores problemas de licença de uso. Já a escolha da linguagem foi definida a fim da compatibilidade com o *software NewtonFractal*, *software* de código aberto que permitiu sua utilização e expansão também sem problemas de licença. Outro fator considerado no momento destas escolhas foi a possibilidade de produzir código livre de modo que possa ser utilizado no desenvolvimento de outros *softwares* de computação científica e sirva para a exploração de código-fonte.

O *FractalNoise* opera em conjunto com o *NewtonFractal*, que gera o fractal associado a uma equação polinomial, possibilitando a visualização das bacias de convergência das raízes a partir da aplicação do método de Newton. O *FractalNoise* extrai do *NewtonFractal* as sequências de aproximações das raízes e transforma estas sequências em som.

Para realizar as análises exploratórias, foi criada uma bateria de testes contendo diversas equações. As equações foram escolhidas de modo a possibilitar a geração de sons diferenciados para bacias de convergência (regiões de estabilidade) e simetrias.

O *FractalNoise* é utilizado no estudo sobre o delicado comportamento do algoritmo numérico, onde comparações entre os sons gerados podem definir áreas de maior ordem de convergência para uma determinada raiz. Além disso, o *software* pode ser utilizado para obter informações sobre o comportamento geral do sistema caótico - ou mesmo na composição musical - utilizando categorias de fractais para gerarem sons semelhantes que, mesclados, podem resultar em músicas com características dignas de serem estudadas.

Além de oferecer a interpretação sonora de sequência de valores, o som produzido pelo *FractalNoise* também pode ser utilizado como ferramenta de apoio no ensino da disciplina de Métodos Numéricos, onde a resolução de equações é uma das unidades de conteúdo.

Pelo registro da trajetória percorrida para a obtenção do *FractalNoise*, bem como pelos estudos que se fizeram necessários, é possível afirmar que o objetivo geral do Trabalho de Conclusão foi atingido, ou seja, a possibilidade de representação da instabilidade de um algoritmo numérico, através do som, foi confirmada. E, para tanto, foi necessário compreender modelos matemáticos, analisar implementações, pesquisar e elaborar modelos conceituais, engajando-se nas atividades propostas.

Outra conclusão importante foi a constatação, na prática, de que resolver problemas é uma ação transdisciplinar. Diferente da pesquisa disciplinar que diz respeito, no máximo, a um conjunto de disciplinas, a transdisciplinaridade se interessa pela dinâmica gerada pela reunião de conhecimentos de diversas áreas. Mas, para dar conta da transdisciplinaridade, foi necessário aprender a trabalhar em equipe e reconhecer que a reunião do conhecimento de diversas pessoas, quando interligada por um mesmo objetivo, pode realmente produzir soluções.

A incerteza de abrir brechas na segurança do conhecido para realizar descobertas é um dos legados deste trabalho. Não se sabia se o trabalho teria sucesso ou não, mas isto não foi motivo para desistir. De fato, isto foi um fator motivacional importante pelo desafio de aplicar conhecimento sobre resolução de problemas matemáticos com um "quê" de inovação.

Sugere-se a sequência deste trabalho, explorando músicas fractais para o estudo do som gerado matematicamente. Assim, questões como a geração de composições harmônicas com base no comportamento do algoritmo e os resultados da sobreposição de representações sonoras dos fractais, poderão ser aprofundadas a partir dos resultados obtidos neste trabalho.

Referências Bibliográficas

BACKES, A. R.; BRUNO, O. M. *Técnicas de Estimativa da Dimensão Fractal: Um estudo comparativo*. 2005. <http://www.dcc.ufla.br/infocomp/artigos/v4.3/art07.pdf>.

BARTZ, G. *Dó, ré, mi, fá, sol, lá, si - A origem do nome das notas musicais*. 2011. <http://www.estacaomusical.com.br/jornaldaestacao/curiosidades/53/do-re-mi-fa-sol-la-si>.

BECKER, D. *Understanding and using Java MIDI audio*. 2009. <http://www.ibm.com/developerworks/library/it/it-0801art38/>.

CARDOSO, M. *História da música ocidental*. 2009. <http://xa.yimg.com/kq/groups/24960419/1838510957/name/21438272-Historia-da-musica-Occidental.pdf>.

CASTRO, D. S. d. *Mais técnica que arte II*. 2011-03. <http://culturadigital.br/dacio/2011/03/02/mais-tecnica-que-arte-ii/>.

CASTRO, D. S. d. *Convertendo sons em cores*. 2011-06. <http://culturadigital.br/dacio/2011/06/04/convertendo-sons-em-cores/>.

CINTRA, L. A. *Análise Fractal de eventos de relâmpagos*. 2005. <http://mtc-m16.sid.inpe.br/col/sid.inpe.br/iris%401916/2005/09.30.13.51/doc/analise%20fractal.pdf>.

CLÁUDIO, D. M.; MARINS, J. M. *Cálculo Numérico Computacional, Teoria e Prática*. 3º. ed. [S.l.]: Editora Atlas, 2000.

FERNANDES, J. A. *Fractais: Uma nova visão da matemática*. 2007. http://www.diaadiaeducacao.pr.gov.br/diaadia/diadia/arquivos/File/conteudo/artigos_teses/MATEMATICA/MonografiaFractais.pdf.

FILLIPIN, G. G. *Estudo da geometria fractal e aplicações em sala de aula*. 2009. <http://www.unifra.br/cursos/matematica/downloads/TFG%20FINAL%20GABRIELA%20FILLIPIN%20C.pdf>.

FREITAS, A. L. S. *Um estudo do método de Newton fractal para resolução de equações*. 2011. <http://revistaseletronicas.pucrs.br/ojs/index.php/graduacao/article/viewFile/8807/6171>.

GODOY, M. F. d. *Teoria do caos aplicada à medicina*. 2005. <http://www.mfgodoy.med.br/caos.pdf>.

HAZARD, C.; KIMPORT, C.; JOHNSON, D. *Introduction to Fractal Music*. 1998 – 1999. <http://www.tursiops.cc/fm/>.

- KRAPAS, S. *Por que a luz é uma onda eletromagnética?: visita aos trabalhos de Maxwell*. 2009. <http://www.sbf1.sbfisica.org.br/eventos/snef/xviii/sys/resumos/T0897-1.pdf>.
- LORENZ, E. N. *Predictability: Does the Flap of a Butterfly's Wings in Brazil set off a Tornado in Texas?* 1972. http://eapsweb.mit.edu/research/Lorenz/Butterfly_1972.pdf.
- MANDELBROT, B. B. *How long is the coast of Britain?* 1967. http://users.math.yale.edu/~bbm3/web_pdfs/howLongIsTheCoastOfBritain.pdf.
- MANDELBROT, B. B. *The Fractal Geometry of Nature*. 1º. ed. [S.l.]: WH Freeman, 1982.
- MINGATOS, D. d. S. *Matemática e música a partir do estudo do monocórdio e de figuras musicais*. 2006. <http://www.ime.ufg.br/bienal/2006/poster/daniellemingatos.pdf>.
- MORAES, J. G. V. d. *História e música: uma canção popular e conhecimento histórico*. 2000. <http://www.scielo.br/pdf/rbh/v20n39/2987.pdf>.
- NUNES, R. S. R. *Geometria Fractal e Aplicações*. 2006. <http://www.fc.up.pt/pessoas/jfalves/Teses/Raquel.pdf>.
- OLIVEIRA, D. *MIDI, SAIBA O QUE É E COMO FUNCIONA*. 2009. <http://www.outrosventos.com.br/portal/tecnologia/computacao-musical/midi-saiba-o-que-e-e-como-funciona.html>.
- ORTIZ, J. A. P. *Música Fractal: el sonido del caos*. 2000. <http://www.dlsi.ua.es/~japerez/pub/pdf/mfsc2000.pdf>.
- POZZER, C. T. *Fundamentos de Cor*. 2008. http://www-usr.inf.ufsm.br/~pozzer/disciplinas/cg_2_cor.pdf.
- RICKARD, S. *The World's Ugliest Music*. 2011. <http://tedxtalks.ted.com/video/TEDxMIAMI-Scott-Rickard-The-Wor>.
- SANTOS, P. P. d. *Acorde*. 2009. <http://www.infoescola.com/musica/acorde/>.
- SILVA, C. C.; MARTINS, R. d. A. *A 'Nova Teoria sobre Luz e Cores' de Isaac Newton: uma Tradução Comentada*. 1996. <http://www.sbfisica.org.br/rbef/pdf/v18a33.pdf>.
- SILVA, C. C.; MARTINS, R. d. A. *A teoria das cores de Newton: Um exemplo do uso da história da ciência em sala de aula*. 2003. <http://www.scielo.br/pdf/ciedu/v9n1/05.pdf>.
- SILVA, M. A. B. d. *Padrão Midi*. 2007. <http://www.midiacom.uff.br/~debora/fsmm/trab-2007-1/midi.pdf>.
- SILVA, M. F. F. d. *Esclarecendo o significado de 'cor' em física*. 2007. <http://www.sbfisica.org.br/fne/Vol8/Num1/v08n01a06.pdf>.
- SILVEIRA, M. T. *RGB to Gray*. 2009. <http://marmsx.msxall.com/artigos/rgb2gray.pdf>.

SPRING, K. R.; DAVIDISON, M. W. *Sources of Visible Light*. 2003. <http://micro.magnet.fsu.edu/primer/lightandcolor/lightsourcesintro.html>.

STASKEVICH, J. *Midi note numbers*. 2011. <http://highlyliquid.com/hl2012/blog/912>.

VIANA, G. V. R. *Métodos Iterativos para resolução de sistemas de equações não lineares*. 2001. <http://www.flf.edu.br/revista-flf.edu/volume01/12.pdf>.

VILELA, L. A. R. *Utilização de L-Systems para Construção de Cenas Virtuais e Análise de Desempenho em Sistema de Realidade Virtual Distribuída*. —. <http://www.unimep.br/phpg/mostraacademica/anais/4mostra/pdfs/261.pdf>.

WISNIK, J. M. *O som e o sentido - Uma outra história das músicas*. [S.l.]: Companhia das Letras, 1989.

ZAMMOURI, I.; AYEB, B. *Fractal Shapes Description with Parametric L-system and Turtle Algebra*. 2007. <http://www.waset.org/journals/waset/v34/v34-19.pdf>.

APÊNDICE A – Cores e sons

Isaac Newton, em 1666, no escuro de seu quarto, fez um buraco na janela para deixar entrar certa quantidade de luz do sol. Sob esta luz, colocou um prisma de vidro para que a luz pudesse ser refratada na parede oposta (SILVA; MARTINS, 1996) (SILVA; MARTINS, 2003).

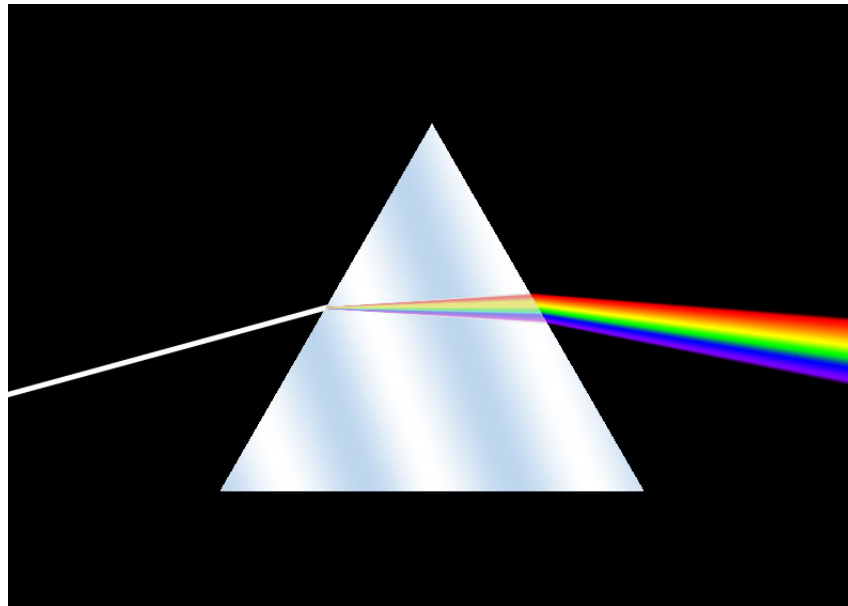


Figura A.1: O prisma da experiência de Newton reproduzido: as cores da luz branca são refratadas pelo prisma.

Fonte: http://pt.wikipedia.org/wiki/Ficheiro:Dispersion_prism.jpg

Mais tarde, em 1862, James Clerk Maxwell viria a dar o primeiro passo em direção à união do eletromagnetismo com a ótica inferindo que *"a luz consiste em ondulações transversais do mesmo meio que é a causa dos fenômenos elétricos e magnéticos"* (KRAPAS, 2009). Sendo a luz uma onda eletromagnética, possui comprimento de onda (CASTRO, 2011-03). Tomando por base a velocidade da luz ($3 \times 10^8 m/s$ no vácuo), o comprimento das ondas eletromagnéticas perceptíveis pelo olho humano (entre 380 e 740nm) (CASTRO, 2011-06) e os divisores mostrados no espectro da figura A.2, obtém-se a Tabela A.1, que mostra o comprimento de onda e a frequência de cada cor (SPRING; DAVIDISON, 2003)(POZZER, 2008)(SILVA, 2007b).

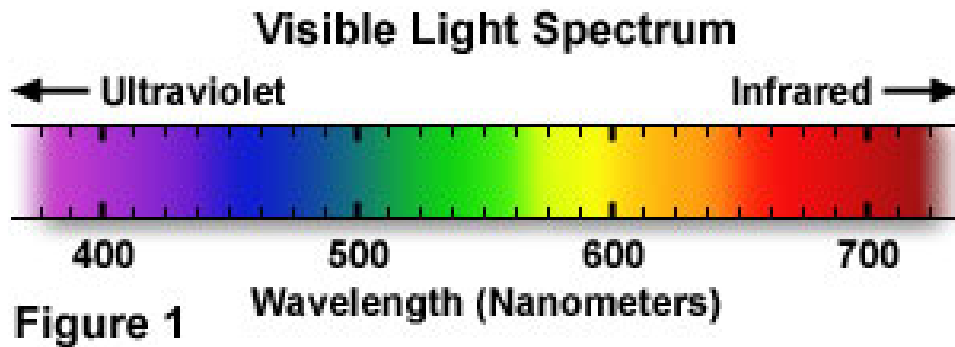


Figura A.2: A divisão do espectro das cores visíveis de acordo com o comprimento da onda.

Fonte: <http://micro.magnet.fsu.edu/primer/lightandcolor/images/lightsourcesfigure1.jpg>

Cor	Comprimento (nm)	Frequência (THz)
Vermelho	≈ 625-780	≈ 480-405
Laranja	≈ 597-622	≈ 510-480
Amarelo	≈ 570-597	≈ 530-510
Verde	≈ 492-570	≈ 600-530
Azul	≈ 455-492	≈ 620-600
Anil	≈ 450-455	≈ 680-620
Violeta	≈ 390-450	≈ 769-665

Tabela A.1: A variação (Δ) do comprimento de onda (λ) e a frequência (f) das cores no espectro de luz visível

Relacionando a Tabela A.1 com a Tabela 2.1 (utilizando as frequências da oitava central a partir do *Lá*), pode-se chegar à Tabela A.2, que mostra a relação das cores com as notas musicais, através do cálculo da frequência (f) por

$$f = \frac{v}{\lambda} \quad (\text{A.1})$$

onde v é a velocidade de propagação da luz no meio e λ , o comprimento da onda (CASTRO, 2011-06). Para melhor interpretação do espectro de cores, foram adicionados também os semi-tons ($\sharp$).

Desta maneira, associando tais valores às teclas de um piano, as teclas são coloridas conforme a Figura A.3, com as cores podendo abranger mais de uma nota, diferindo na tonalidade da execução.

Nota	Frequência (Hz)	Cor	Comprimento (nm)	Frequência (THz)
<i>Dó</i>	523,3	Verde	521	575
<i>Dó#</i>	554,4	Azul	492	610
<i>Ré</i>	587,3	Anil	464	646
<i>Ré#</i>	622,3	Violeta	438	684
<i>Mi</i>	659,3	Violeta	414	725
<i>Fá</i>	698,5	Violeta	390	768
<i>Fá#</i>	737,0	Vermelho	737	407
<i>Sol</i>	392,0	Vermelho	696	431
<i>Sol#</i>	415,3	Vermelho	657	457
<i>Lá</i>	440,0	Laranja	620	484
<i>Lá#</i>	466,2	Amarelo	585	513
<i>Si</i>	493,9	Verde	552	543

Tabela A.2: As notas musicais e suas cores.



Figura A.3: As cores representadas nas teclas de um piano.

Fonte: http://audiolist.org/forum/images/faq/cores_notas_1.gif

APÊNDICE B – Protótipo da análise baseada em cores

Após ser elaborada uma abordagem teórica dos conceitos que relacionam cores e músicas, quando aplicados à fractais e músicas fractais, um protótipo foi desenvolvido para verificar, na prática, esta abordagem. Usando o *software Newton Fractal* para gerar a imagem do fractal para determinada equação de teste, o protótipo faz a leitura desta imagem pixel a pixel, coluna por coluna. Estes pixels são interpretados e transformados em notas musicais a serem reproduzidas no sintetizador da *JavaSound*.

A abordagem, utilizada para transformação dos pixels em notas músicas, é trivial: cada coluna é lida pixel a pixel e os valores da composição de cada pixel (vermelho (R), verde (G) e azul (B)) de uma coluna são obtidos e transformados em um tom de cinza - através da fórmula que leva em consideração a capacidade de absorção do olho humano (SILVEIRA, 2009):

$$cinza = R * 0,3 + G * 0,59 + B * 0,11. \quad (B.1)$$

Após ser obtido o valor do tom de cinza de cada pixel da coluna, uma média aritmética entre estes tons é feita, sendo esta o valor assumido para a coluna. Tendo 256 tons de cinza, o tom médio resultante de cada coluna é mapeado (numa razão 2 para 1) para uma das 128 notas do padrão MIDI. Desta forma, obtém-se uma nota para cada coluna analisando todos seus pixels.

O protótipo permite que uma imagem seja carregada e o processo de conversão em tons de cinza seja acompanhado pela miniatura da mesma. A imagem em tamanho maior é mantida para que seja acompanhado o progresso, visualizando as cores originais do fractal.

A interface do protótipo pode ser visualizada na Figura B.1. Já a Figura B.2 mostra o diagrama das principais classes implementadas no protótipo, trabalhando essencialmente com computação gráfica na interpretação da imagem em cores.

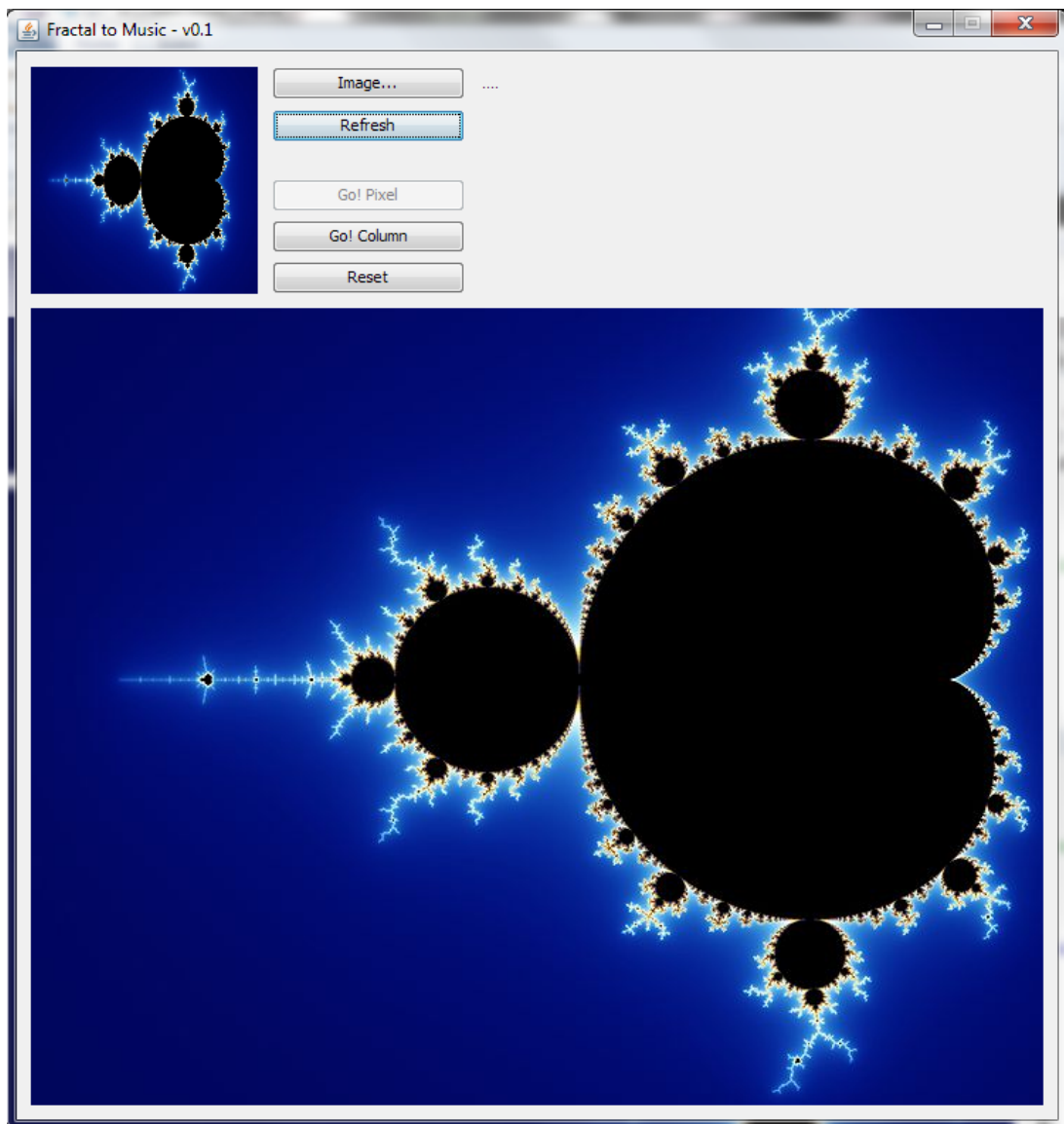


Figura B.1: A interface do protótipo quando carregada uma imagem do conjunto de Mandelbrot.

O som resultante da análise do conjunto de Mandelbrot da Figura B.1 está disponível no *link* <http://soundcloud.com/nestorsalamon/fractal-music-mandelbrot>

Esta abordagem demasiado simples serviu para testar a viabilidade da leitura de um fractal e sua conversão em som, levando em consideração as cores. Também provou que esta relação entre cores e sons não vai capturar as propriedades matemáticas características dos fractais a serem analisadas - como no caso de as cores para o mesmo fractal serem modificadas. Fez-se necessária, então, outra abordagem de leitura destas características.

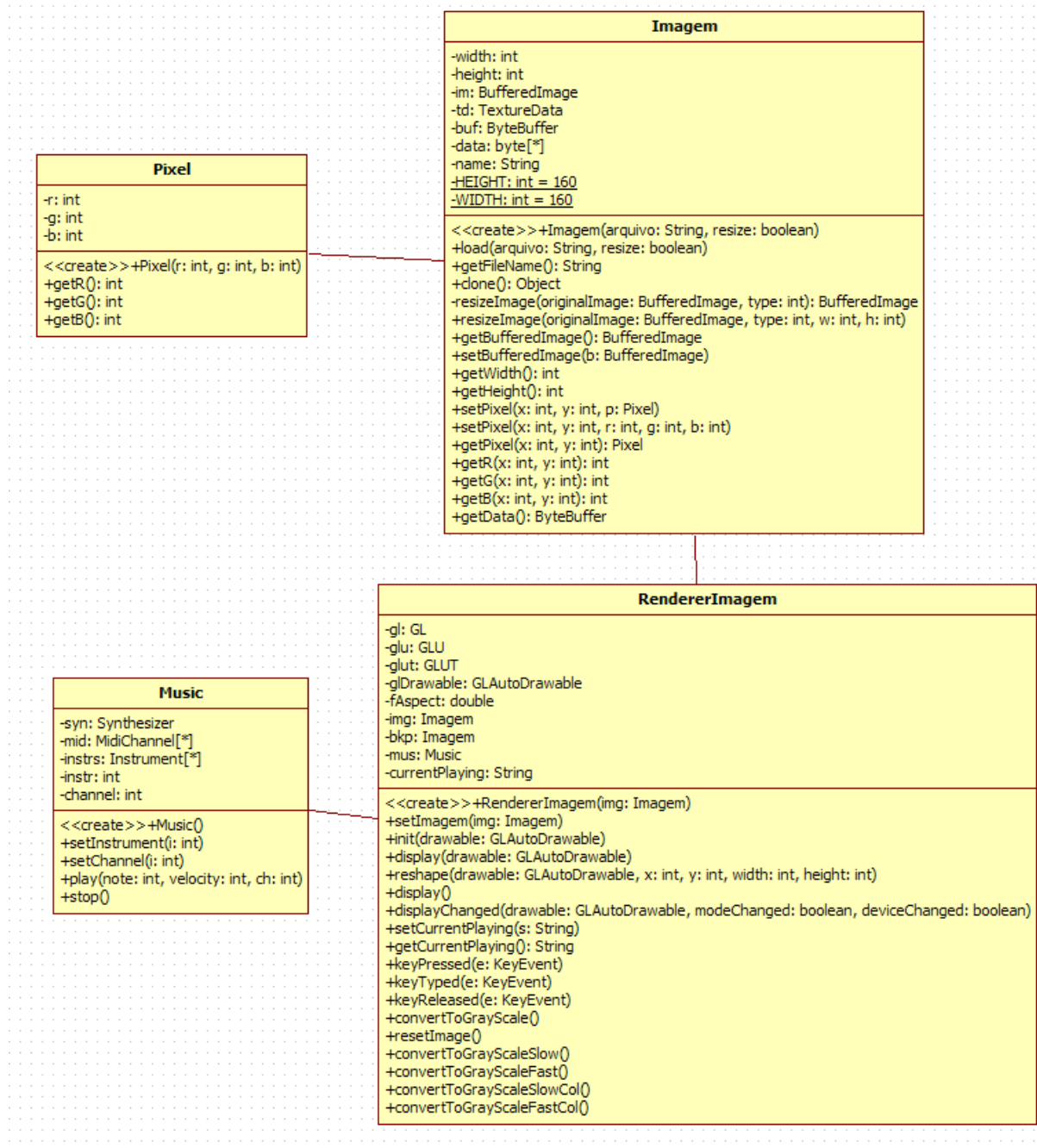


Figura B.2: As principais classes do protótipo implementado.

APÊNDICE C – Diagramas de Atividades

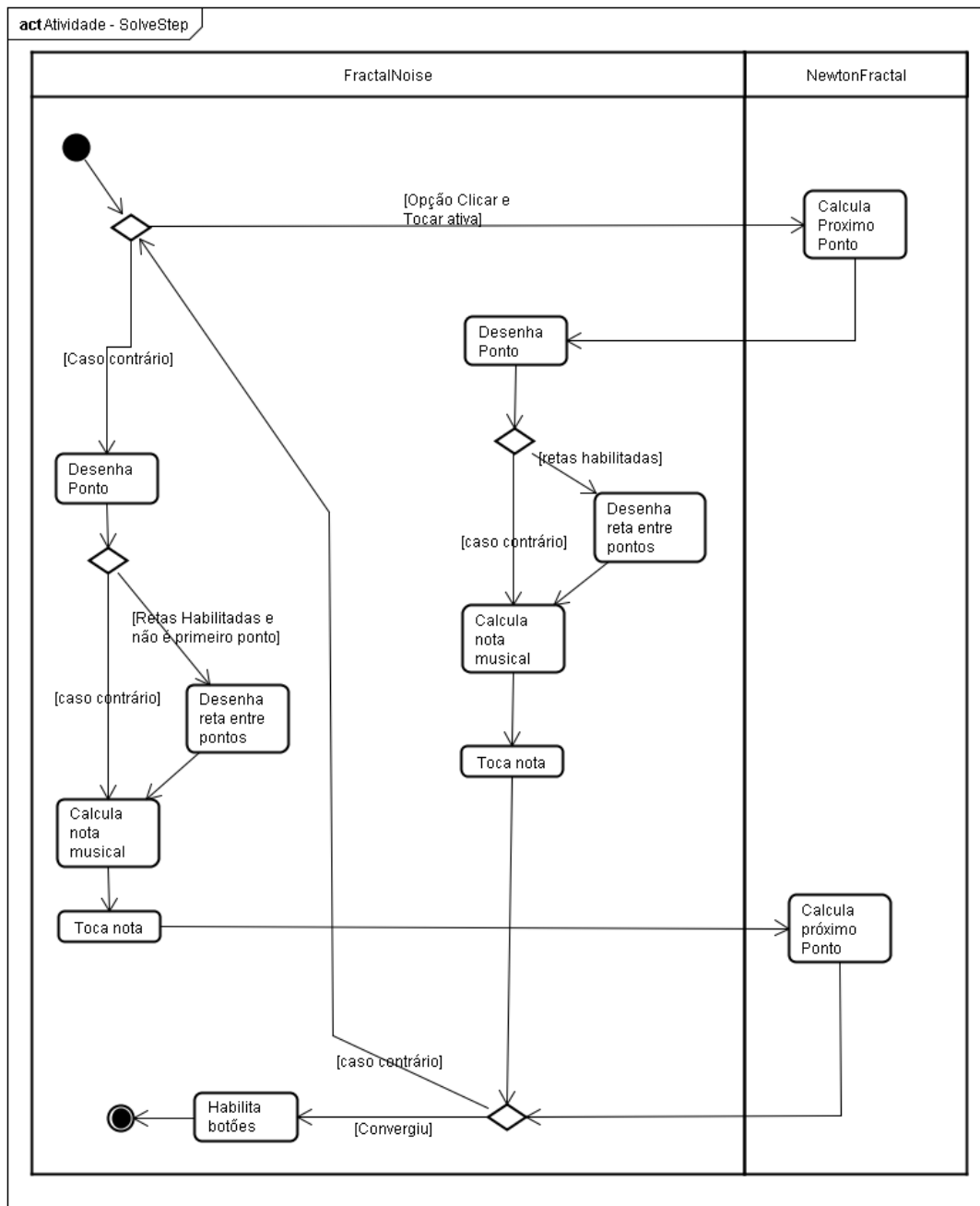


Figura C.1: Diagrama de atividades representando o fluxo da análise *Solve Step*

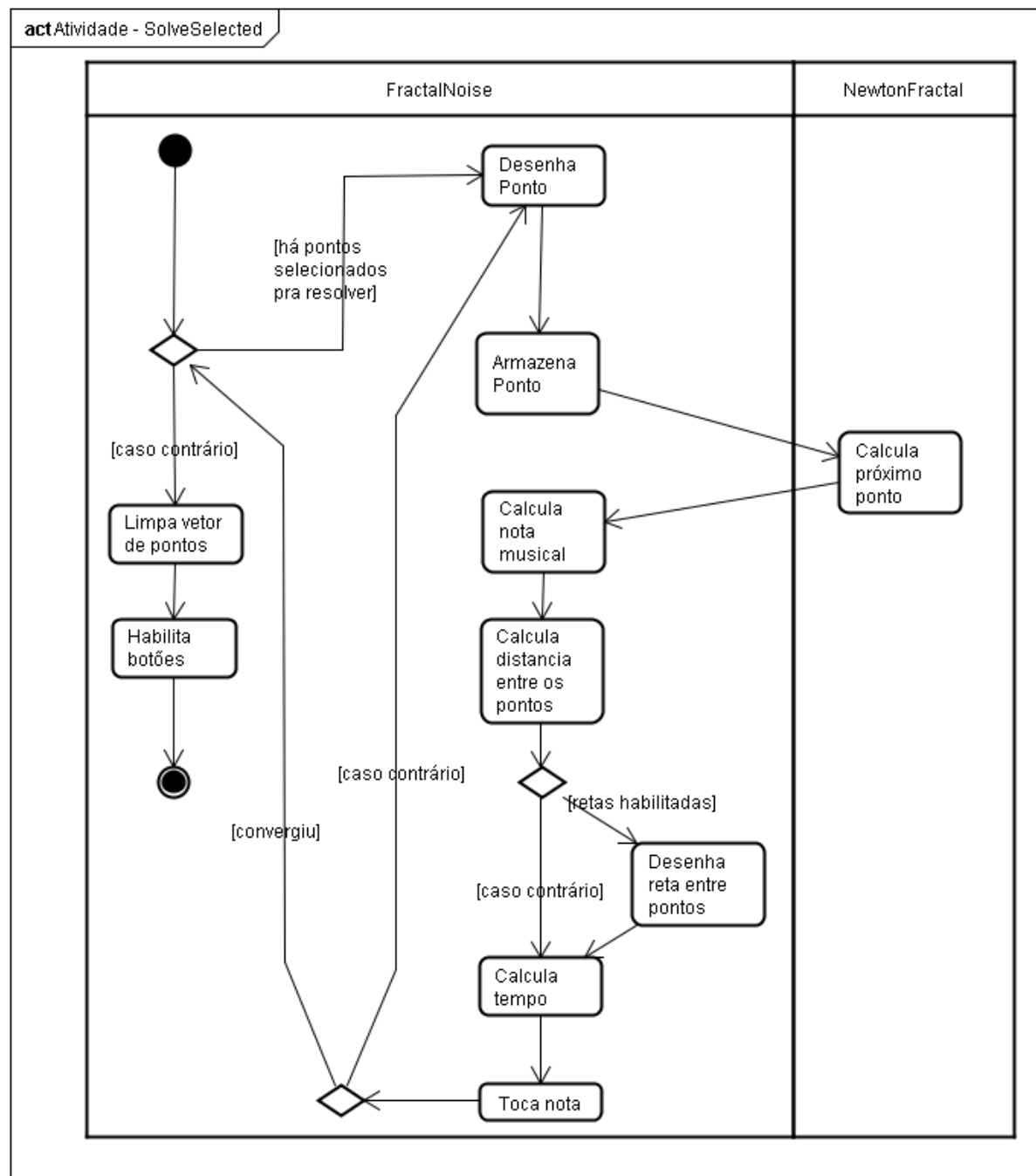


Figura C.2: Diagrama de atividades representando o fluxo da análise *Solve Selected*

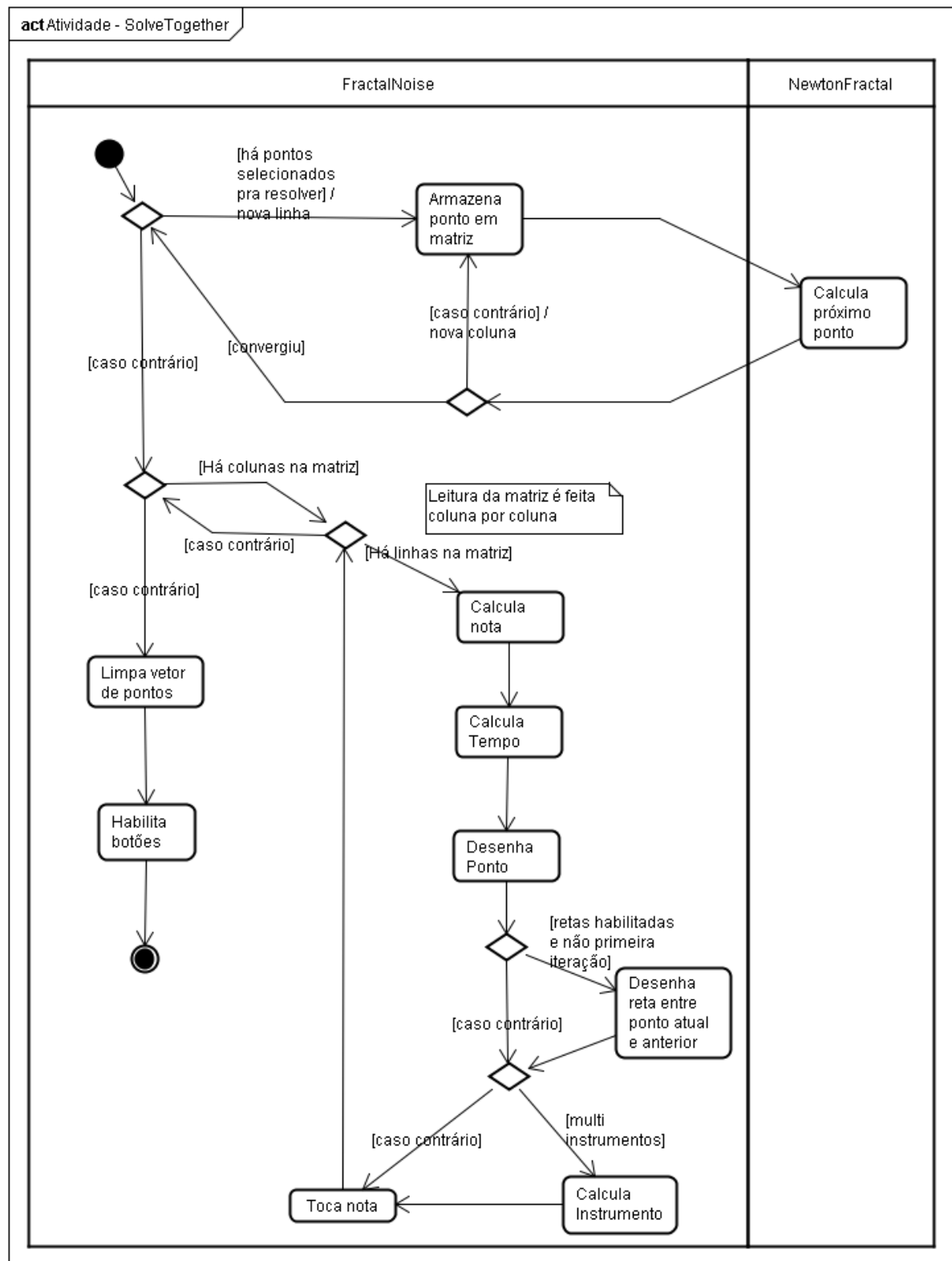


Figura C.3: Diagrama de atividades representando o fluxo da análise *Solve Together*

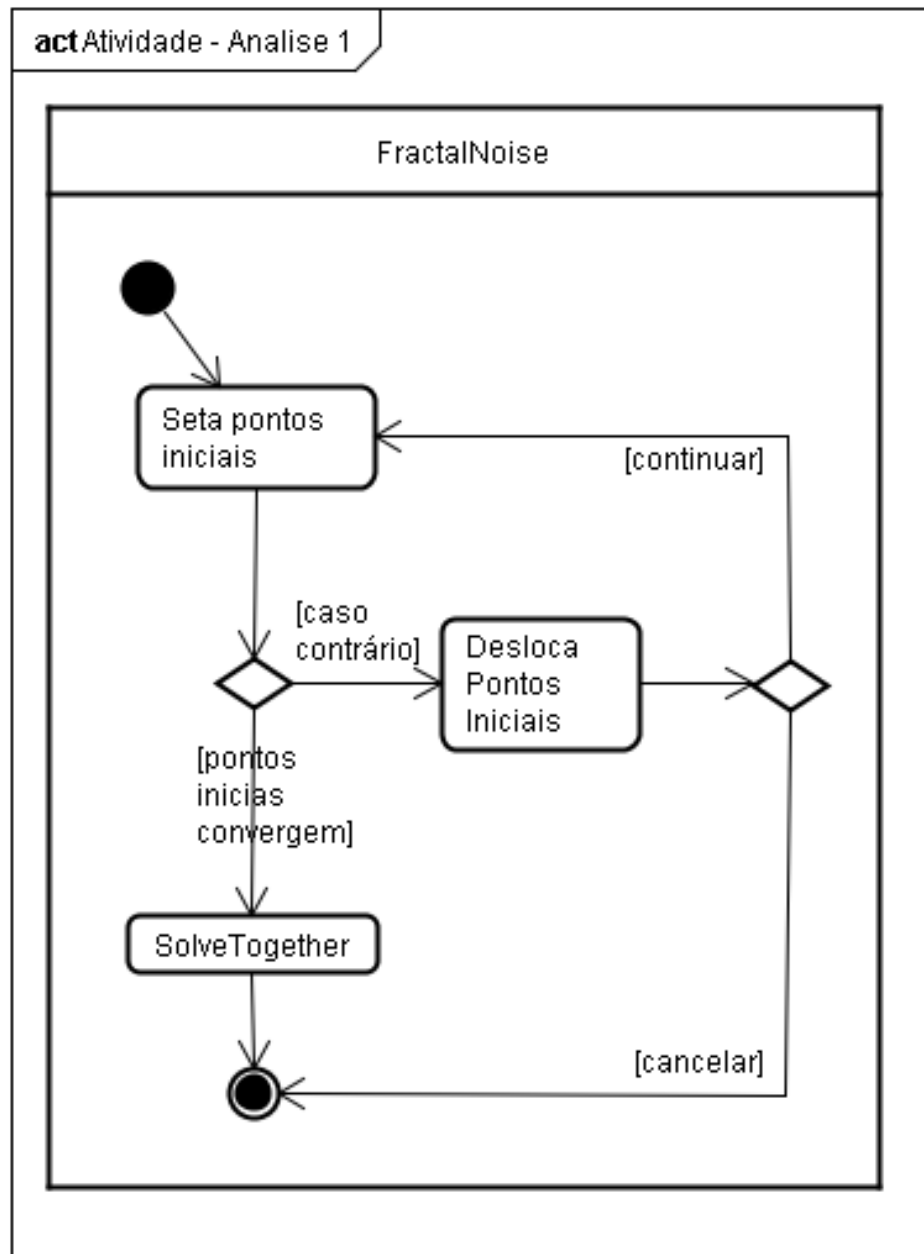


Figura C.4: Diagrama de atividades representando o fluxo da *Analyse 1*

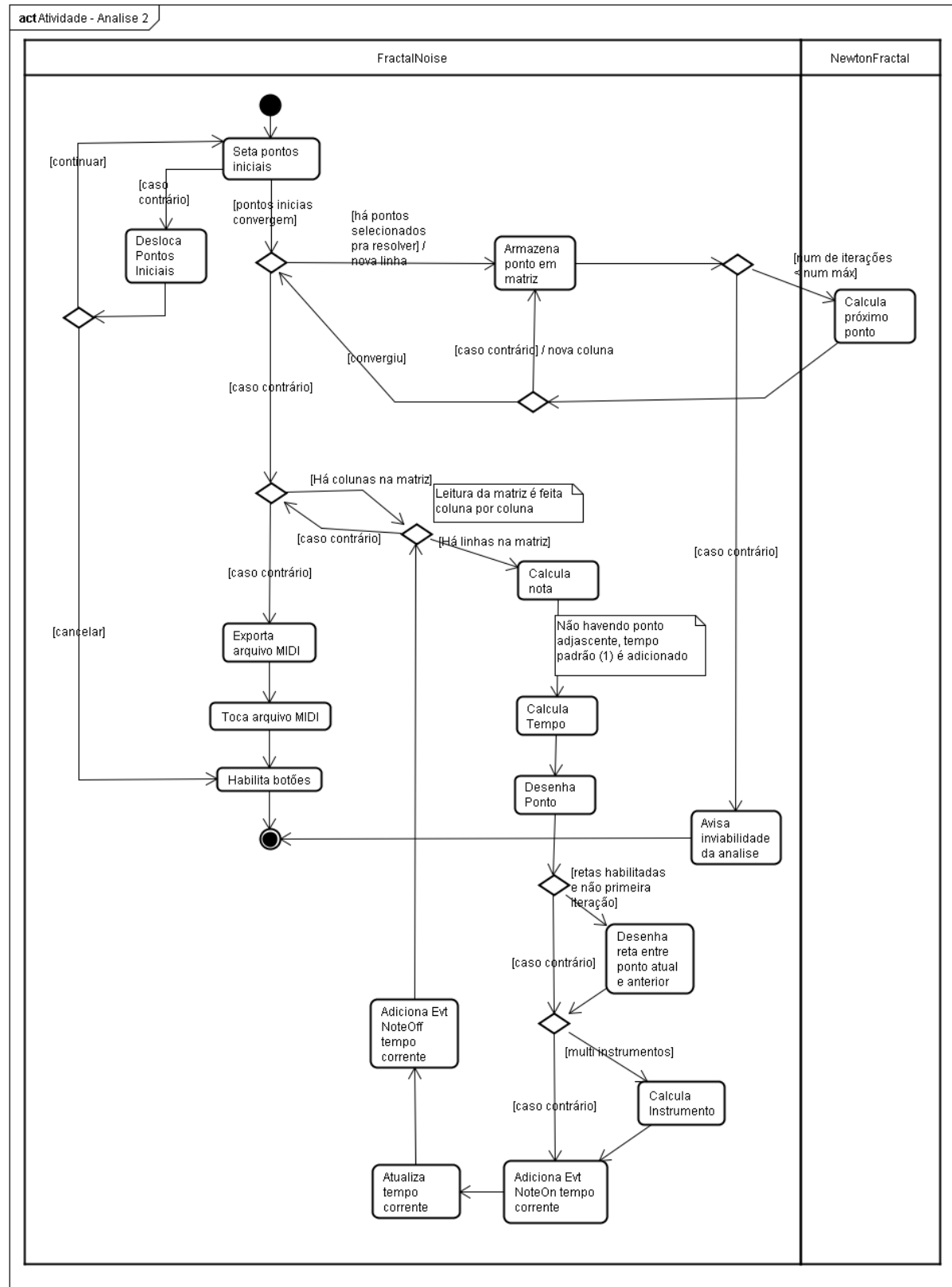


Figura C.5: Diagrama de atividades representando o fluxo da *Analise 2*

APÊNDICE D – Mapa mental

A Figura D.1 mostra uma macro visão da relação entre os assuntos abordados neste trabalho. Este mapa mental foi elaborado durante a criação da proposta de Trabalho de Conclusão para contextualizar as áreas de interesse e verificar as possibilidades de pesquisa e aplicação do conteúdo. Nele é possível perceber a interdisciplinaridade trabalhada: conteúdos de matemática, computação e artes são abordados. O mapa também foi útil na ligação destes conteúdos, guiando os estudos.

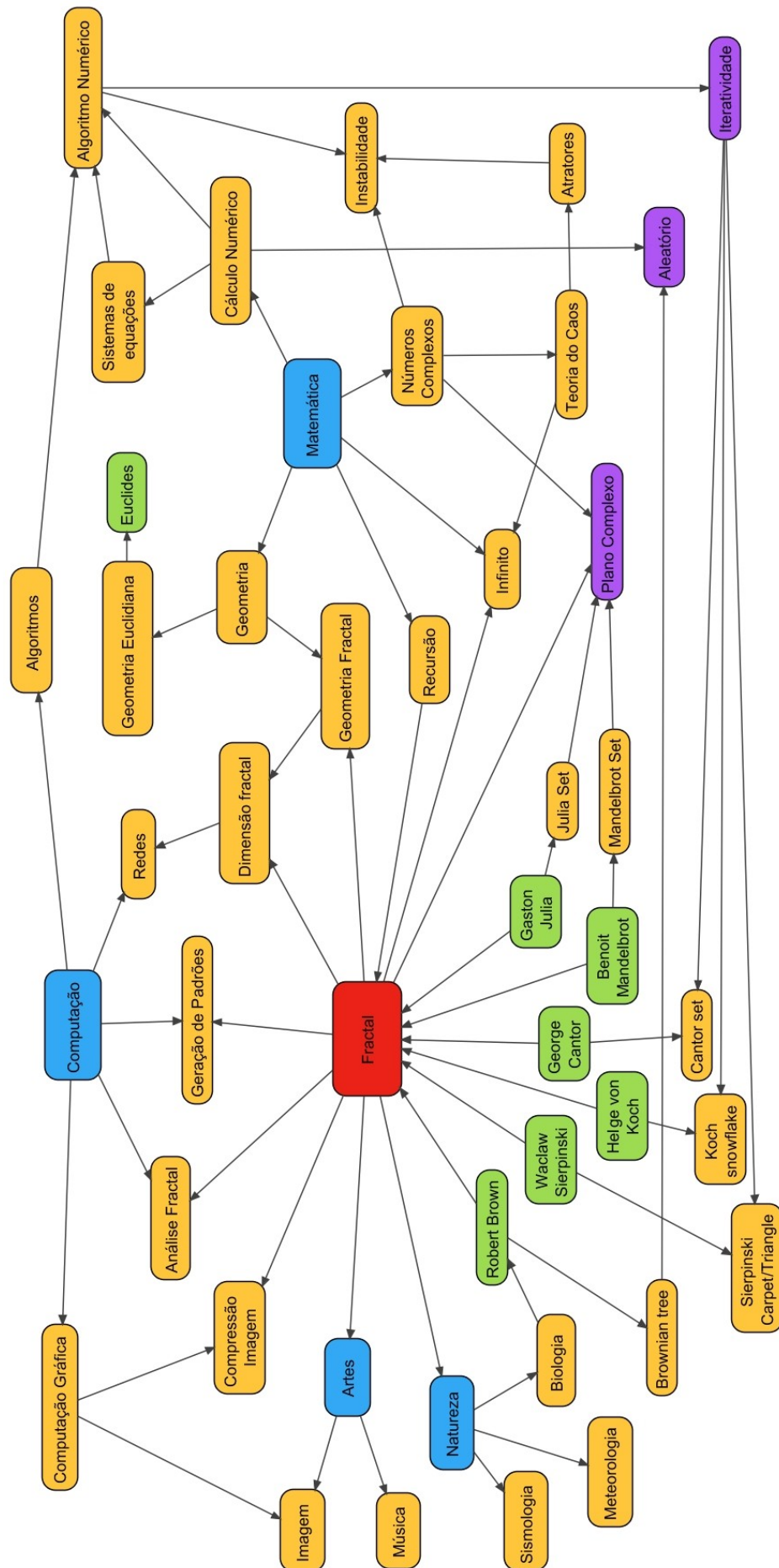


Figura D.1: O mapa mental utilizado para escolha da área e orientação do estudo.