Universidade de São Paulo
FFCLRP - Departamento de Física
Pós-graduação em Física aplicada à Medicina e Biologia

ISAIAS JOSÉ AMARAL SOARES

**Desenvolvimento de uma base de funções
paramétricas para interpolação de imagens
médicas**

Tese apresentada à Faculdade de
Filosofia, Ciências e Letras de Ribeirão
Preto da Universidade de São Paulo,
como parte das exigências para a
obtenção do título de Doutor em Ciên-
cias, Área: Física aplicada à Medicina
e Biologia.

Ribeirão Preto - SP
2013

ISAIAS JOSÉ AMARAL SOARES

**Desenvolvimento de uma base de funções
paramétricas para interpolação de imagens
médicas**

Tese apresentada à Faculdade de
Filosofia, Ciências e Letras de Ribeirão
Preto da Universidade de São Paulo,
como parte das exigências para a
obtenção do título de Doutor em
Ciências.

Área de Concentração:

Física aplicada à Medicina e Biologia.

Orientador:

Luiz Otávio Murta Júnior.

Versão corrigida

Versão original disponível na FFCLRP - USP

Ribeirão Preto - SP

2013

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

FICHA CATALOGRÁFICA

Soares, Isaias José Amaral

Desenvolvimento de uma base de funções paramétricas para interpolação de imagens médicas / Isaias José Amaral Soares; orientador Luiz Otávio Murta Júnior. Ribeirão Preto - SP, 2013.

91 f.:il.

Tese (Doutorado - Programa de Pós-graduação em Física aplicada à Medicina e Biologia) - Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto da Universidade de São Paulo, 2013.

1. Interpolação. 2. B-Splines. 3. Paradigma não-extensivo. 4. Fractais.

Nome: Soares, Isaias José Amaral

Título: Desenvolvimento de uma base de funções paramétricas para interpolação de imagens médicas

Tese apresentada à Faculdade de Filosofia,
Ciências e Letras de Ribeirão Preto da
Universidade de São Paulo, como parte
das exigências para a obtenção do título
de Doutor em Ciências.

Aprovado em: ____/____/____.

Banca Examinadora

Prof(a). Dr(a). : _____ Instituição: _____

Julgamento: _____ Assinatura: _____

Prof(a). Dr(a). : _____ Instituição: _____

Julgamento: _____ Assinatura: _____

Prof(a). Dr(a). : _____ Instituição: _____

Julgamento: _____ Assinatura: _____

Prof(a). Dr(a). : _____ Instituição: _____

Julgamento: _____ Assinatura: _____

Prof(a). Dr(a). : _____ Instituição: _____

Julgamento: _____ Assinatura: _____

Dedico esse trabalho ao Pai. Palavras me faltam, mas ainda assim, Ele me compreende.

Dedico aos que amo, aos que me amam. Aos que leem a alma com o olhar, aos que amam.

Dedico aos que nunca desistem, por já se sentirem vitoriosos.

Dedico, enfim, a todos os que, de coração aberto, desejam fazer desse Universo, um lugar melhor para se viver.

Isaias.

Agradecimentos

agradecimentos

Eu gostaria de agradecer aos meus pais, Joana e José, pelo apoio constante. Quero agradecer à minha família, pela paciência e compreensão nos momentos em que tive de me ausentar e trabalhar até tarde, em especial, minhas amadas Solange e Ariel. Também, a todo apoio que recebi de meu orientador, o prof. Murta, que me acompanhou e inspirou durante todo o período desse trabalho, e por quem nutro elevada admiração. Quero agradecer aos meus amigos, pela força e apoio em muitos momentos. Quero agradecer ao apoio financeiro da CAPES, que me permitiu uma maior tranquilidade nessa fase.

A todos, agradeço com carinho.

Isaias.

“ ‘Tentar’ não existe.
Faça.
Ou não faça.”
Mestre Yoda

x

Resumo

SOARES, I.J.A. **Desenvolvimento de uma base de funções paramétricas para interpolação de imagens médicas**. 2013. 91 f. Tese (Doutorado - Programa de Pós-graduação em Física aplicada à Medicina e Biologia) - Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto, Universidade de São Paulo, Ribeirão Preto - SP, 2013.

O uso de imagens é crucial na medicina, e seu uso no diagnóstico de doenças é uma das principais ferramentas clínicas da atualidade. Porém, frequentemente necessitam de pós-processamento para serem úteis. Embora ferramentas clássicas sejam utilizadas para esse fim, elas não dão tratamento específico a certas características de imagens fractais, como as provindas de sistemas biológicos. Nesse enfoque, este trabalho objetivou a criação de novas bases de interpolação utilizando a Q-Estatística para verificar se seriam estas seriam adequadas à representação de objetos com características fractais que as bases clássicas. Foram criados dois tipos de splines: uma unidimensional e outra bidimensional, que permitiram um tipo diferente de interpolação, fundamentado na q-Estatística. Os testes demonstraram a potencialidade dessas ferramentas para uso em sinais e imagens médicas, com acentuada redução do erro de interpolação no caso unidimensional (em até 351,876%) e uma redução sutil no caso bidimensional (0,3%). Como resultado adicional, foram criados filtros de imagens e avaliados seus resultados em imagens médicas, que resultaram em melhorias de até 1.340% de ganho efetivo na remoção de ruídos de natureza fractal (marrom). Os resultados sugerem que as q-bases desenvolvidas foram

capazes de representar melhor imagens e sinais médicos, bem como é interessante o uso dos filtros desenvolvidos na remoção de diversos tipos de ruído do tipo $1/f^\beta$.

Palavras-chave: 1. Interpolação. 2. B-Splines. 3. Paradigma não-extensivo. 4. Fractais.

Abstract

SOARES, I.J.A. **Development of parametric basis function for interpolation of medical images**. 2013. 91 f. Thesis (Ph.D. - Postgraduate program in Physics applied to Medicine and Biology) - Faculty of Philosophy, Sciences and Literature, University of São Paulo, Ribeirão Preto - SP, 2013.

The use of images is crucial in modern medicine, and diagnostic imaging is a major clinical tools used in detecting, monitoring and completion of many treatments. However, often the images need to be post-processed for display to health professionals or automated analysis, searching for signs of abnormalities. Although classical tools are used for that purpose, they do not give special treatment to certain characteristics of fractal images, such as those coming from biological systems. These characteristics are produced, in general, by complex dynamic systems as a result of internal interactions of sub-system components, giving the system a fractal character.

In this context, the main objective of this work was to propose interpolation bases using the Q-statistic, creating bases of Q-interpolation, and verify if such bases would be best suited to the representation of objects with fractal characteristics than classical bases, assumed the premise that such a theory model best this kind of phenomenon than classical theory. Based on this hypothesis, we created two types of splines: one-dimensional and one-dimensional, called Q-splines, which allow a different type of interpolation and they can capture behaviors as super-additive or sub-additive among the consti-

tients of a spline. These models have demonstrated numerically the potential use of this type of interpolation for use in signals and medical images, reducing the interpolation error by up to 351.876 % in the one-dimensional case and 0.3 % in two dimensional.

As secondary results, were defined two families of image filters, called anisotropic Q-filters and isotropic Q-filters, and their results were evaluated in real medical images. In virtually all analyzes it was possible to obtain the best results from conventional approaches, sometimes with improvements of 1.340 % in some filters, in removing noise fractal nature (brown). The results were more modest for the interpolation of two-dimensional images, however, generally proved exciting and encouraging, clearly showing that these new approaches are not only viable, but also can produce better results compared to classical approaches.

Based on these results, we concluded that the Q-bases developed are best able to represent not only signs but medical imaging (1D and 2D) although its use can be improved by the adoption of approaches adapted to the vector representation of information, that favor the use of *splines*. Similarly, the Q-filters were more suitable for the processing of medical signals when compared to conventional approaches.

Key-words: 1. Interpolation. 2. B-Splines. 4. Nonextensive paradigm. 4. Fractals.

Lista de Figuras

Lista de Figuras

2.1	Constantino Tsallis.	6
2.2	Relações entre as estruturas algébricas do tipo $(G,*)$	14
2.3	Os cinco primeiros polinômios de Hermite.	20
2.4	Exemplo de uma <i>spline</i> e seus trechos.	25
2.5	Um exemplo de pontos a serem interpolados, para $\alpha=3$ ($W_3(x)$).	28
2.6	Interpolação sobre $W_3(x)$ com os valores das derivadas nu- los nos nós.	31
2.7	Interpolação sobre $W_3(x)$ com os valores das derivadas va- lendo $V_0 = 0$, $V_F = \text{sen}(i)$, para cada trecho da <i>spline</i>	32
2.8	Funções de mistura da <i>spline</i> de Hermite interpolante.	33
2.9	Exemplo de uma superfície de Hermite.	36
2.10	Comparativo de desempenho entre CPUs e GPUs e suas ta- xas de transferência de memória (adaptado do site oficial da Nvidia).	38
2.11	Comparativo das proporções de unidades físicas por funci- onalidade entre CPUs e GPUs (adaptado de [37]).	40
2.12	Arquitetura de memória (adaptado de [37]).	40
2.13	Escalabilidade dos núcleos de processamento CUDA (adaptado de [38]).	41
2.14	Modelo de execução de uma aplicação CUDA.	42
2.15	Modelo de aplicação CUDA.	43
2.16	Bloco de processamento: cada <i>kernel</i> pode invocar um bloco de núcleos de processamento de até cinco dimensões.	44

2.17 Modelo de memória de uma aplicação CUDA na GPU (adaptado de [38]).	45
2.18 Fluxo de processamento de um programa CUDA.	48
3.1 Gráfico da q-Gaussiana unidimensional para $\sigma = 1$ e diversos valores de q . Essa função tende à função delta quando $q \rightarrow -\infty$	55
3.2 Gráfico da q-Gaussiana unidimensional para $\sigma = 2$ e diversos valores de q (a marcação de funções é a mesma que na Figura 3.1).	55
3.3 Alguns representantes da família das q-Gaussianas.	57
3.4 Alguns parâmetros de criação dos filtros q-Gaussianos.	58
3.5 Alguns exemplos de filtros da família das q-Gaussianas gerados pelo protótipo, com seus respectivos espectros de Fourier.	59
3.6 Amostragem sobre a curva de Weierstrass e a respectiva curva interpolante.	64
3.7 Tela do protótipo em Java para teste das q-Gaussianas.	67
3.8 Estrutura do protótipo em Java.	67
4.1 Exemplo de uma imagem do conjunto geral de fractais e dos níveis de ruído introduzidos nas imagens.	73
4.2 Ganho dos q-filtros em imagens de fractais gerais para ruído gaussiano com $r=3$	74
4.3 Ganho dos q-filtros em imagens de cérebro humano para ruído gaussiano com $r=3$	75
4.4 Ganho dos q-filtros de desvio padrão = 1.0 em imagens de fractais gerais para ruído gaussiano com $r=60$	76
4.5 Ganho dos q-filtros de desvio padrão = 1.0 em imagens de cérebro humano para ruído gaussiano com $r=60$	77
4.6 Ganho dos q-filtros de desvio padrão = 1.0 para a imagem 'im000000' de cérebro humano com alto nível de ruído.	78

4.7 Exemplo da imagem do conjunto geral de cérebro ‘im000000.jpg’ e dos níveis de ruído introduzidos na ima- gem.	78
4.8 Ganhos do filtro q-Gaussiano isotrópico com a adição de ruído marrom (10%).	83
4.9 Ganhos do filtro q-Gaussiano anisotrópico com a adição de ruído marrom.	83
4.10 Ganhos do filtro q-Gaussiano anisotrópico com a adição de ruído branco.	84
4.11 Ganhos do filtro q-Gaussiano isotrópico com a adição de ruído branco.	84
4.12 Ganho total médio dos filtros q-Gaussianos anisotrópicos em relação aos diferentes tipos de ruído.	85
4.13 Comparativo entre a filtragem q-Gaussiana e a clássica. . .	86
4.14 Exemplo do efeito de borramento entre filtragem clássica e q-Gaussiana.	88
4.15 Exemplo do efeito do filtro q-Gaussiano em uma imagem de RM cerebral T2 e seus respectivos histogramas.	92
4.16 Diversos representantes da “q-Base de Hermite”.	94
4.17 Exemplos de “q-Spline” com valores de q entre $-0,98$ a $2,9999$	98
4.18 Curvas de Weierstrass.	99
4.19 Exemplo das características não aditivas da q-Spline. . . .	100
4.20 Visualização do erro médio quadrático em função de q para $W_2(x)$. Os pontos destacados correspondem aos $q_{E_{min}}$ de cada curva.	101
4.21 Visualização do erro médio quadrático em função de q para $W_3(x)$. Os pontos destacados correspondem aos $q_{E_{min}}$ de cada curva.	102
4.22 Visualização do erro médio quadrático em função de q para $W_4(x)$. Os pontos destacados correspondem aos $q_{E_{min}}$ de cada curva.	103

4.23	Visualização do erro médio quadrático em função de q para $W_5(x)$. Os pontos destacados correspondem aos $q_{E_{min}}$ de cada curva.	104
4.24	Visualização da superfície q -Spline para $q=2.9$	105
4.25	Imagem cerebral (I00032) com diferentes níveis de interpolação, para $q=1.38$	107
4.26	Visualização do erro médio quadrático em função de q para imagens de tórax. Os pontos destacados correspondem aos $q_{E_{min}}$ de cada curva.	108
4.27	Diferença entre os diferentes tipos de interpolação (imagem I00005, tórax).	108
4.28	Visualização do erro médio quadrático em função de q para imagens de abdome. Os pontos destacados correspondem aos $q_{E_{min}}$ de cada curva.	109
4.29	Diferença entre os diferentes tipos de interpolação (imagem I00005, abdome).	110
4.30	Visualização do erro médio quadrático em função de q para imagens de cérebro. Os pontos destacados correspondem aos $q_{E_{min}}$ de cada curva.	111
4.31	Diferença entre os diferentes tipos de interpolação (imagem I00032, cérebro).	112
4.32	Visualização da influência da super/sub-aditividade na interpolação.	112
A.1	Exemplo de ajustes de curva.	145
A.2	Representação antiga da astrologia.	147
A.3	Nicolau Copérnico.	149
A.4	Augustin Louis Cauchy.	152

Lista de Tabelas

2.1 Tabela de funcionalidades em função da capacidade de computação (<i>compute capability</i>).	49
4.1 Comparativo de ganhos para filtros q-Gaussianos anisotrópicos para imagens cerebrais, nível 10% de ruído.	81
4.2 Comparativo de ganhos para filtros q-Gaussianos isotrópicos para imagens cerebrais, nível 10% de ruído.	82
4.3 Comparativo de ganhos para filtros q-Gaussianos anisotrópicos em imagens T2 de abdome, para diferentes tipos de ruídos a 10%.	89
4.4 Comparativo de ganhos para filtros q-Gaussianos isotrópicos em imagens T2 de abdome, para diferentes tipos de ruídos a 10%.	90
4.5 Comparativo de ganhos para filtros q-Gaussianos anisotrópicos em imagens T2 de abdome, para diferentes tipos de ruídos a 20%.	90
4.6 Comparativo de ganhos para filtros q-Gaussianos isotrópicos em imagens T2 de abdome, para diferentes tipos de ruídos a 20%.	91
4.7 Erro médio quadrático para diferentes amostragens sobre a curva $W_2(x)$ ($\alpha = 2$).	100
4.8 Erro médio quadrático para diferentes amostragens sobre a curva $W_3(x)$ ($\alpha = 3$).	102
4.9 Erro médio quadrático para diferentes amostragens sobre a curva $W_4(x)$ ($\alpha = 4$).	103

4.10 Erro médio quadrático para diferentes amostragens sobre a curva $W_5(x)$ ($\alpha = 5$).	104
4.11 Erro médio quadrático para diferentes amostragens para a imagem I00005 (tórax).	108
4.12 Erro médio quadrático para diferentes amostragens sobre a curva para a imagem I00013 (abdome).	110
4.13 Erro médio quadrático para diferentes amostragens sobre a curva para a imagem I00032 (cérebro).	111

Sumário

Lista de Figuras	xv
-------------------------	-----------

Lista de Tabelas	xix
-------------------------	------------

1 Introdução	1
1.1 Considerações Preliminares	1
1.2 Objetivos	2
1.3 Contribuições	3
1.4 Organização	3
1.5 Estilos usados ao longo do texto	4
2 Fundamentos teóricos	5
2.1 Introdução	5
2.2 Mecânica Estatística Não-Extensiva	5
2.3 Noções de Álgebra	10
2.3.1 Introdução	10
2.3.2 Algumas estruturas algébricas	10
2.3.3 Espaços vetoriais polinomiais	17
2.4 Interpolação e Aproximação	22
2.4.1 Introdução	22
2.4.2 <i>Splines</i>	23
2.4.3 B-Splines	26
2.4.4 Splines bidimensionais	32
2.5 Paralelização de software: CUDA	37
2.5.1 Introdução	37
2.5.2 Arquitetura	39

2.5.3	Linguagens	40
2.5.4	Modelo de desenvolvimento	42
2.5.5	Interface de programação	44
2.5.6	Capacidade de computação	47
2.6	Processando imagens em C/C++: OpenCV	48
3	Metodologia	51
3.1	Introdução	51
3.2	Um breve histórico	51
3.3	Métodos utilizados	54
3.3.1	Elaboração e testes dos q-filtros	54
3.3.2	Elaboração e teste das q-Splines	62
3.4	Os protótipos computacionais	66
4	Resultados e discussão	71
4.1	Introdução	71
4.2	Experimentos com a q-Gaussiana e q-filtros	71
4.2.1	Primeira etapa: protótipo Java	71
4.2.2	Segunda etapa: testes preliminares em C++/CUDA/OpenCV	79
4.2.3	Terceira etapa: ampliação dos testes em C++/CUDA/OpenCV	87
4.3	Testes com as q-Splines	93
4.3.1	Primeira etapa: Elaboração das q-bases e estudo analítico	93
4.3.2	Segunda etapa: Testes unidimensionais	99
4.3.3	Terceira etapa: Formulação bidimensional	105
4.3.3.1	Formulação e testes analíticos	105
4.3.3.2	Testes computacionais	106
4.4	Conclusões	113
4.5	Trabalhos futuros	117
	Referências Bibliográficas	119

	xxiii
Glossário	127
Apêndice A - Interpolação: notas históricas	145
A.1 Introdução	145
A.2 Histórico	147
Índice Remissivo	155

Introdução

Introdução

1.1 Considerações Preliminares

O desenvolvimento tecnológico tem possibilitado a cada dia novas e modernas formas de diagnóstico médico, cada vez mais precisas e econômicas. Em meio a uma gama crescente de opções de avaliação, o uso de imagens tem se tornado tão fundamental, que uma especialidade médica surgiu, inteiramente dedicada a esse assunto, o atual **diagnóstico por imagem**¹. Após a descoberta dos raios-x por Wilhelm Conrad Röntgen (1845-1923), uma série de outras ferramentas possibilitou o uso de imagens para fins de diagnóstico, como os ultrassons, ressonância magnética e medicina nuclear. Esses recursos constituem a base tecnológica utilizada em uma imensa gama de exames na medicina moderna, todos fundamentados na produção e análise de imagens do corpo humano.

Situada um passo à frente, a “Intervenção orientada por imagem” tem por objetivo a realização de procedimentos minimamente invasivos, como no caso das angioplastias. Esses procedimentos de desobstrução de artérias eram realizados somente por cirurgia aberta há não muito tempo, e hoje podem ser efetuados via cateterismo [2]. Neste

¹Diagnóstico por imagem é uma especialidade médica voltada ao uso das tecnologias de aquisição de imagens para realização de diagnósticos. No Brasil, essa especialidade é reconhecida pelo Conselho Federal de Medicina sob o nome de "Radiologia e diagnóstico por imagem" [1].

caso, o papel das imagens passa a ser essencial, já que são utilizadas no manejo de instrumentos no interior do corpo do paciente.

O processamento e arquivamento digital de imagens médicas está se tornando cada vez mais necessário, uma vez que a quantidade de informações geradas pelas novas tecnologias é muito grande. Nesse contexto, o estudo de métodos de interpolação de imagens médicas vem contribuir nas alternativas de representação da informação e no consequente impacto das melhorias obtidas no seu uso diário.

A interpolação é fundamental em algumas transformações utilizadas em processamento de imagens, como a ampliação, mudança de coordenadas, e alinhamento ou registro de imagens. E muitas vezes, da eficiência da interpolação depende a qualidade de todo o processamento adicional. Neste sentido, a presente proposta pretende contribuir para o campo de imagens médicas, avaliando as ferramentas existentes e propondo novas, com base na q-Estatística.

1.2 Objetivos

O objetivo principal deste trabalho é propor bases de interpolação de modo a incorporar as características da q-Estatística (q-bases) e verificar se tais bases são mais adequadas à representação de imagens médicas que as bases clássicas, assumida a premissa de que tal teoria modele melhor esse tipo de situação que a teoria clássica. As metas para o cumprimento desse objetivo incluíram explorações mais aprofundadas da q-Gaussiana, por meio do desenvolvimento e análise de filtros envolvendo essa função, bem como elaboração da formulação matemática de novas bases para B-Splines 1D e 2D, as quais deveriam incorporar características da **q-estatística**², bem como testes e

²A q-estatística representa um novo paradigma, proposto em 1988 por Constantino Tsallis [3] e fundamenta-se em uma nova definição para a entropia, da qual diversas consequências surgem, entre elas, a não-aditividade da entropia da soma de dois sistemas. A definição de entropia proposta por Tsallis é $S_q =$

$$k \frac{1 - \sum_{i=1}^W p_i^q}{q - 1} \quad \left(\sum_{i=1}^W p_i = 1, q \in \mathbb{R} \right).$$

validação dos conceitos desenvolvidos.

1.3 Contribuições

As principais contribuições desse trabalho são:

- Elaboração de splines 1D e 2D com base na q-estatística (q-Splines);
- Testes e validação analíticos das splines criadas;
- Testes numéricos em curvas sintéticas e em imagens médicas das q-interpolações propostas;
- Elaboração de uma infra-estrutura de software para testes automatizados de interpolação de imagens médicas;
- Elaboração de um algoritmo para a criação de q-filtros genéricos;
- Desenvolvimento de um protótipo para a criação de q-filtros genéricos;
- Avaliação numérica do desempenho de q-filtros em função de q.

1.4 Organização

Este trabalho está organizado da seguinte forma: no capítulo 2, são apresentados conceitos fundamentais gerais. A seguir, no capítulo 3, são descritos a metodologia e os procedimentos adotados para a investigação conduzida durante o trabalho, e os resultados são apresentados no capítulo 4, acompanhados da análise dos experimentos. Ainda nesse capítulo, são apresentadas uma análise crítica dos pontos importantes dos resultados, bem como algumas perspectivas de trabalhos futuros.

Também incluímos um resumo da pesquisa histórica realizada no decorrer dos estudos preparatórios na área de interpolação (**Apêndice A**).

1.5 Estilos usados ao longo do texto

Ao longo desse texto são utilizados alguns estilos, para marcar, enfatizar e denotar certas características, como, por exemplo, a existência de uma definição constante do glossário ao final da obra. Tais estilos são:

- *Itálico*: usado para enfatizar uma palavra, e para termos estrangeiros;
- **Negrito**: usado para termos que tenham uma definição explícita no final do texto dessa tese;
- ‘Aspas simples’: usadas para destacar termos e em figuras de linguagem (palavras sendo utilizadas de modo um pouco diferente do habitual);
- “Aspas duplas”: usadas para designar termos e em citações.

Fundamentos teóricos

2.1 Introdução

finalidade deste capítulo é introduzir alguns conceitos fundamentais utilizados neste trabalho do modo mais sucinto possível, permitindo ao leitor acompanhar o texto sem uso excessivo de literatura complementar. Essa abordagem auto contida pode parecer enfadonha ou repetitiva para aqueles que dominem certas áreas (e nesse caso, sugerimos que pulem as seções em questão), mas certamente trará benefícios à maioria dos leitores, além de enfatizar a abordagem adotada no restante do material. Informações adicionais ou não essenciais à compreensão deste trabalho são apresentadas nos apêndices, eventualmente complementando estes tópicos introdutórios.

2.2 Mecânica Estatística Não-Extensiva

A **Mecânica estatística** é um ramo da ciência que se utiliza da Teoria das Probabilidades para calcular propriedades macroscópicas de sistemas físicos constituídos por muitas partículas [4]. Tais sistemas podem ser um copo de água, um tanque de gás, ou mesmo um cubo de gelo. Com base em leis Físicas e em análises probabilísticas, propriedades macroscópicas são determinadas na forma de estatísticas de maior probabilidade de ocorrência, para um dado conjunto de condições do

sistema. Tais estatísticas representam características como temperatura, volume, pressão ou quaisquer outras propriedades de interesse. Sob essa ótica, essas propriedades não são constantes, mas oscilam em torno de um valor médio de máxima probabilidade.

Fundamentada nestes princípios, a “**Mecânica estatística não-aditiva**”, “**Q-estatística**” ou “**Estatística de Tsallis**” representa um paradigma proposto em 1988 por **Constantino Tsallis** [3, 5, 6].

Figura 2.1: *Constantino Tsallis.*

Essa proposta baseia-se na idéia de que, durante a análise de um sistema, a ferramenta informacional específica (entropia) mais interessante a ser utilizada deve ser definida com base nas suas próprias características geométrico-dinâmicas [7]. Desse modo, sistemas com propriedades diferentes exigiriam ferramentas diferentes, mais adequadas às suas peculiaridades.

Estudos recentes [8] apontam evidências experimentais e numéricas limitações para o domínio de validade da teoria de BG¹ canônica. Durante muitos anos, não houve avanços sólidos em favor da aceitação desta teoria. Mesmo a definição precisa dos seus domínios de validade permanecem um problema ainda não resolvido (como por exemplo, a distribuição do equilíbrio canônico associado, à qual Takens considera um dogma [9]). A lista de situações para as quais a teoria de BG falha (isso quando não é totalmente violada), cresce a cada dia. Dentre tais situações pode-se citar o ferromagnetismo [10], interações de longo alcance [11, 12, 13, 14, 15] (i.e, gravitação com $d=3$ [16, 17]), turbulência bidimensional de plasma puro-elétron [18], sistemas granulares [19], neutrinos solares [20] e buracos negros [21] dentre uma série de outros fenômenos.

A proposta de Tsallis deu origem a um intenso debate sobre o assunto, motivando, até a presente data, mais de 3.000 trabalhos e

¹BG: Boltzmann-Gibbs.

teses feitos por mais de 5.000 pesquisadores em 51 países [6, 8]. Estabeleceu-se assim um debate sobre os fundamentos desse tema básico da Física [8]. Foi um acontecimento ímpar, no qual se discutiram desde os fundamentos até a validade da estatística de Boltzmann e Gibbs (BG). Um dos mais notáveis resultados surgiu em fevereiro de 2010, quando cientistas do *Large Hadron Collider* (LHC, ou “Grande Colisor de Hádrons”) mostraram que as medidas de distribuição de partículas observadas em colisões ocorridas em altíssimas energias: 2,36 teraelétron-volts (TeV) são descritas de melhor maneira utilizando a teoria generalizada de Tsallis [22, 23], evidenciando a força da teoria.

A proposta de Tsallis expande a teoria clássica de Boltzmann-Gibbs, superando muitas de suas limitações. Fenômenos antes não tratáveis passaram a fazer parte do elenco de situações passíveis de solução. Um ponto importante é que a nova proposta não apenas expande, como também engloba a teoria clássica de BG, e define a entropia generalizada pela seguinte expressão [3, 24, 25, 26]:

$$S_q = k \frac{1 - \sum_{i=1}^W p_i^q}{q - 1} \quad \left(\sum_{i=1}^W p_i = 1, q \in \mathbb{R} \right) \quad (2.1)$$

na qual k é uma constante positiva e W é o número total de possibilidades microscópicas do sistema². Para o caso em que $q < 0$, é importante desconsiderar todas aquelas possibilidades que possuem probabilidades que não sejam estritamente positivas, do contrário S_q divergiria. Tal cuidado não se faz necessário para $q > 0$, propriedade na qual a entropia é dita *expansível*, ou seja, podemos verificar que $S_q(p_1, p_2, \dots, p_W, 0) = S_q(p_1, p_2, \dots, p_W)(\forall \{p_i\}; \forall q)$. Portanto, a Equação (2.1) recai (para $p_i^{q-1} = e^{(q-1)\ln p_i} \sim 1 + (q-1)\ln p_i$) na entropia de BG ($-k_B \sum_{i=1}^W p_i \ln p_i$) no limite quando $q \rightarrow 1$, sendo k_B a constante de Boltzmann.

²Tais possibilidades estão relacionadas com os níveis de energia do sistema. Quanto mais níveis, mais configurações possíveis o sistema apresenta, e menor a probabilidade da ocorrência de uma configuração específica (microestado), p_i .

O **índice entrópico** q é um parâmetro introduzido por Tsallis na expressão da entropia, e está relacionado aos micro-estados possíveis, caracteriza o grau de não-extensividade da entropia. Dados dois sistemas A e B independentes³, então temos a seguinte regra de pseudo-extensividade da entropia

$$\frac{S_q(A+B)}{k} = \frac{S_q(A)}{k} + \frac{S_q(B)}{k} + (1-q) \frac{S_q(A)}{k} \frac{S_q(B)}{k} \quad (2.2)$$

Nessa relação, quando $q < 1$ tem-se **super-aditividade** (a entropia é dita **super-extensiva**), quando $q = 1$, tem-se **aditividade** (**entropia extensiva**) e quando $q > 1$, tem-se **sub-aditividade** (**entropia sub-extensiva**).

A partir do trabalho de Tsallis, diversos trabalhos surgiram, incorporando os novos conceitos às teorias clássicas, como a generalização do Teorema do Limite Central, e ainda, a elaboração de uma q -álgebra, q -exponenciais, q -logaritmos, q -gaussianas uma q -transformada de Fourier [27] e um 'q-cálculo' [28]. Umarov e Tsallis ainda definiram as quatro operações fundamentais: Q-soma, Q-diferença, Q-produto e a Q-divisão, lançando-se as bases para a Q-Álgebra, segundo a qual:

$$x \oplus_q y = x + y + (1-q)xy \quad (2.3)$$

$$x \ominus_q y = \frac{x - y}{1 + (1-q)y} \quad (2.4)$$

$$x \otimes_q y = [x^{1-q} + y^{1-q} - 1]_+^{\frac{1}{1-q}} \quad (2.5)$$

$$x \oslash_q y = (x^{1-q} - y^{1-q} + 1)^{\frac{1}{1-q}} \quad (2.6)$$

na qual $[a]_+ = \max\{0, a\}$. disso, definiram ainda a Q-exponencial como

$$e_q^x = [1 + (1-q)x]_+^{\frac{1}{1-q}} \quad (2.7)$$

³Probabilisticamente falando, ou seja, $p_{ij}(A+B) = p_i(A)p_j(B)$.

$$e_q^{x \oplus_q y} = e_q^x e_q^y \quad (2.8)$$

$$e_q^{x+y} = e_q^x \otimes_q e_q^y \quad (2.9)$$

e o Q-logaritmo,

$$\ln_q(x) = \frac{x^{1-q} - 1}{1-q}, (x > 0) \quad (2.10)$$

$$\ln_q(xy) = \ln_q(x) \oplus_q \ln_q(y) \quad (2.11)$$

$$\ln_q(x \otimes_q y) = \ln_q(x) + \ln_q(y) \quad (2.12)$$

E, por fim, temos a Q-Gaussiana, definida como

$$G_q(\beta, x) = \frac{\sqrt{\beta}}{C_q} e_q^{-\beta x^2} \quad (2.13)$$

com

$$C_q = \begin{cases} \frac{2}{\sqrt{1-q}} \int_0^{\pi/2} (\cos t)^{\frac{3-q}{1-q}} dt = \frac{2\sqrt{\pi}\Gamma(\frac{1}{1-q})}{(3-q)\sqrt{1-q}\Gamma(\frac{3-q}{2(1-q)})}, & -\infty < q < 1 \\ \sqrt{\pi} & q = 1 \\ \frac{2}{\sqrt{q-1}} \int_0^\infty (1+y^2)^{\frac{-1}{q-1}} dy = \frac{\sqrt{\pi}\Gamma(\frac{3-q}{2(q-1)})}{\sqrt{q-1}\Gamma(\frac{1}{q-1})} & 1 < q < 3 \end{cases}$$

e $\beta = 1/(2\sigma^2)$.

Para a q-derivada⁴ de uma função $f(x)$, temos a seguinte expressão:

$$\frac{d_q}{dx} f(x) = D_q f(x) = \lim_{y \rightarrow x} \frac{f(x) - f(y)}{x \ominus_q y} = [1 + (1-q)x] \frac{d}{dx} f(x) \quad (2.14)$$

e a q-integral de uma função $f(x)$ é dada por

$$\int_q f(x) d_q x = \int \frac{f(x)}{1 + (1-q)x} dx \quad (2.15)$$

⁴As versões de q-derivada e q-integral descritas aqui são apenas uma das versões possíveis, pois há definições alternativas (ditas 'duais'). Mais detalhes em [28].

na qual $d_q x = \lim_{y \rightarrow x} x \ominus_q y = \frac{1}{1 + (1 - q)x} dx$.

Para fechar o q-cálculo, ressaltamos que $\int_q D_q f(x) d_q x = D_q \int_q f(x) d_q x = f(x)$.

Todas essas definições foram utilizadas no presente trabalho a fim de se obter as definições de filtros e novas bases de interpolação, foco de nosso estudo.

2.3 Noções de Álgebra

2.3.1 Introdução

A palavra “Álgebra” pode assumir diversos significados, e um deles diz respeito a um ramo da Matemática que estuda os conjuntos e as relações entre eles⁵. Em um outro contexto, uma Álgebra pode representar uma *estrutura algébrica*, ou seja, *um conjunto* de elementos de uma certa classe e suas *operações associadas*, e cada uma dessas estruturas poderia ser denominada como uma *Álgebra* particular.

2.3.2 Algumas estruturas algébricas

Neste tópico, elaboramos uma resumida introdução a algumas estruturas algébricas⁶, as quais estão intimamente ligadas à necessidade humana de se representar objetivamente fenômenos e conceitos relacionados à experiência do indivíduo, a fim de manipulá-los e comunicá-los de modo inequívoco a outros indivíduos. Um fenômeno pode ser descrito por uma pessoa em termos das sensações que lhe

⁵Enquanto ramo da Matemática, a Álgebra evoluiu a partir das noções primitivas de Aritmética, definindo padrões de conjuntos e operações que não apenas são capazes de representar, quantificar e relacionar elementos do mundo real (capazes de serem experimentados de algum modo pelos sentidos do indivíduo), como também de lidar com elementos puramente abstratos, por meio da lógica matemática.

⁶Uma cobertura completa seria demasiadamente longa e desnecessária para os propósitos desse trabalho, porém, aos leitores que desejarem se aprofundar no tema, citamos os livros [29, 30], e diversas outras bibliografias podem ser encontradas com uma busca rápida na internet.

chegam pelos órgãos do sentido, como imagens, cheiros e demais impressões subjetivas. No entanto, uma outra abordagem é possível, na qual tais impressões são traduzidas em abstrações, delimitações e marcações objetivas. É aí que entra o formalismo matemático. Um exemplo é a criação e aplicação de um sistema de coordenadas (que é uma abstração, um conceito mental) no contexto de um sistema factual. Tal processo é chamado de **coordenatização**, e permite que uma representação subjetiva dê lugar a uma objetiva, por meio de quantidades (números) e expressões matemáticas. Representado desse modo, um fato pode ser preservado indefinidamente⁷, compreendido inequivocamente por indivíduos que nem experimentaram a ocorrência original, e o mais importante, pode ser manejado de modo a produzir novas conclusões acerca do fato original. Isso tudo permite a sua reprodutibilidade em outro lugar do mundo. Assim, a sistematização da representação de ideias permite a criação de um lugar comum a todos os que se dedicam à investigação de tais fatos. Essa ideia foi muito bem representada na máxima de Galileu: “Medir tudo o que pode ser medido, e tornar mensurável tudo aquilo que ainda não o seja” citeshafarevich87.

Da busca por equivalentes numéricos e representação de quantidades e situações, surgiram as representações do ponto no plano e espaço como **tuplas**⁸ de números em um sistema de coordenadas, bem como dos números imaginários (na busca das raízes de uma equação), funções elípticas, superfícies de Riemann e diversos outros conceitos destinados a representar fatos ou fenômenos cada vez mais intrincados da realidade. De tempos em tempos as representações existentes se tornaram inadequadas, levando a questionamentos na Matemática, e à elaboração de novas estruturas, capazes de acompanhar a expansão do saber humano. A evolução do conceito de ‘número’ seguiu a linha: ‘1, 2, muitos’ → números naturais → inteiros → racionais → reais

⁷Tal poder de representação confere à Matemática um caráter de *linguagem*, ainda que se restrinja aos conceitos que possam ser representados matematicamente.

⁸Tuplas ou **ênuplas** são sequências finitas e ordenadas de valores que, em geral, representam um elemento (ou uma variedade) em um espaço n-dimensional.

→ números complexos. Matrizes, no entanto, não podem ser incluídas nessa cadeia, apesar de poderem ser consideradas análogas aos números sob diversos aspectos. Estas, juntamente com os quatérnions e demais sistemas hipercomplexos, constituem-se em diferentes estruturas algébricas, que podem ser aplicadas aos mais diversos contextos.

Tais estruturas são fundamentais na representação matemática, e, basicamente podem ser definidas como um conjunto com operações associadas que satisfazem a certos axiomas. Dependendo de quais são as operações, e quais os axiomas, diferentes estruturas são definidas, sendo a mais básica dela o **magma**. Um magma é um par $(G, *)$, no qual G é um conjunto, munido de uma única **operação binária** $*$: $G \times G \rightarrow G$, ou seja, $\forall a, b \in G : a * b \in G$. Operações binárias são fechadas por definição⁹, no entanto, nenhum outro axioma é imposto a tal operação. O magma representa uma das mais básicas estruturas algébricas, sendo portanto o ponto de partida para diversas outras estruturas. A ele podem ser adicionados diversos conceitos diferentes, ‘propriedades’, também chamados de ‘axiomas’, que podem ser aplicados à estrutura $(G, *)$ ¹⁰ e, a cada nova combinação, pode-se atribuir uma nomenclatura diferente. Em linhas gerais, essas propriedades são:

- **Totalidade:** quando a operação $*$ é *total*, ou seja, pode ser aplicada a todos os elementos de G ¹¹.
- **Identidade:** quando a operação possui um elemento neutro e para o qual, todo elemento que for operado com ele resulta no próprio elemento, ou seja $\exists e$ tal que $\forall a \in G, a * e = a$.

⁹Uma operação binária $*$ fechada é uma relação entre o **produto cartesiano** de um conjunto consigo mesmo e o próprio conjunto, $S \times S \rightarrow S$, levando elementos do conjunto a um outro também pertencente ao conjunto. Tal propriedade também é chamada de *fechamento*.

¹⁰Os axiomas em geral se aplicam à características das operações, do conjunto, ou a ambos.

¹¹No caso em que se aplique apenas a um subconjunto de G , a operação $*$ é dita *parcial*.

- **Associatividade:** quando a operação é associativa, ou seja, $\forall a, b, c \in G, (a*b)*c = a*(b*c)$.
- **Invertibilidade:** quando o conjunto possui uma propriedade tal que, dada a imagem da operação, seus argumentos podem ser recuperados a partir da operação inversa. Essa propriedade supõe a reversibilidade da operação $*$, ou seja, a possibilidade da reversão do efeito da operação¹², bem como a existência de um elemento inverso.
- **Comutatividade:** quando a ordem dos argumentos da função $*$ não altera o resultado, ou seja, $\forall a, b \in G, a*b = b*a$;
- **Idempotência:** uma operação é dita *idempotente* quando apresenta o mesmo resultado mesmo após ser aplicada várias vezes sobre o mesmo valor, ou seja, $\forall a \in G, a*a = a$. Se isso ocorrer somente para um dado elemento a , então esse elemento é dito elemento idempotente da operação $*$, embora, nesse caso, a operação não o seja.

A incorporação de combinações dessas propriedades define várias classes de magmas, entre elas:

- **Quase-grupos:** magmas não vazios nos quais a divisão é sempre possível, o que equivale a dizer que para todo elemento pertencente ao magma, há um inverso;
- **Laço:** quase-grupos com elemento identidade;
- **Semigrupos:** magmas nos quais a operação é associativa;
- **Semi-reticulados:** Semigrupos nos quais a operação é comutativa e idempotente;

¹²O conceito de inversão varia de acordo com a operação e a estrutura algébrica em questão. Como exemplo, a subtração em relação à adição, e de inverso (multiplicativo) em relação à multiplicação. Nos espaços de funções, a inversão seria a transformação que leva a imagem de uma função em seus argumentos, ou em um elemento no domínio que corresponda àquela imagem.

- **Monóides:** Semigrupos com elemento identidade;
- **Grupos:** monóides nos quais a operação é inversível (possuem elementos inversos, o que implica também em *divisibilidade*, e vice-versa);
- **Grupos abelianos:** grupos nos quais a operação é comutativa.

Tais estruturas, como mencionamos, são constituídas por um conjunto e apenas uma operação, e, de modo esquemático, podemos relacioná-las do seguinte modo:

Figura 2.2: Relações entre as estruturas algébricas do tipo $(G,*)$.

Até o momento, consideramos apenas um conjunto e uma operação, no entanto, há estruturas mais elaboradas, por exemplo, que utilizam um conjunto e duas operações binárias (por exemplo, soma e multiplicação), formando uma estrutura do tipo $(G, +, *)$. Nesse sentido, podemos criar novos axiomas para regular os comportamentos das operações, tais como:

- **Distributividade à esquerda:** quando uma das operações pode ser ‘distribuída’ em relação a outra à esquerda¹³, sem alterar o

¹³A distinção à esquerda e à direita é importante, pois nem sempre as operações consideradas são comutativas.

resultado final. Como exemplo disso, citamos o caso da distributividade à esquerda da multiplicação em relação à soma, temos que $\forall a, b, c \in G, a*(b + c) = (a*b) + (a*c)$;

- **Distributividade à direita:** quando uma das operações pode ser 'distribuída' em relação a outra à direita sem alterar o resultado final. Como exemplo, citamos a distributividade da multiplicação em relação à soma, na qual temos que $\forall a, b, c \in G, (a + b)*c = (a*c) + (b*c)$;
- **Distributividade:** quando a operação é distributiva à esquerda e à direita em relação a outra operação.
- **Absorção:** se o conjunto atende essa lei, então, para duas operações binárias '\$' e '%', tem-se que $\forall a, b \in G, a$(a%b) = a%(a$b) = a$. Um exemplo de conjuntos que seguem essa lei são conjuntos booleanos.

A adição de mais uma operação e novos axiomas resulta em novas estruturas do tipo $(G, +, *)$, e duas convenções comuns são definir essas operações como a soma e o produto. Tais estruturas podem ser classificadas como:

- **Anel:** um conjunto com duas operações binárias, uma operação de grupo abeliano definida como adição, e uma operação de semigrupo definida como a multiplicação, que satisfaça a distributividade em relação à soma;
- **Corpo:** um anel no qual os elementos não-zero formam um grupo abeliano sob multiplicação;
- **Reticulado:** um conjunto com duas operações comutativas, associativas e idempotentes, que satisfazem a lei de absorção;
- **Álgebra booleana:** um reticulado limitado, distributivo e complementado;

- **Campo:** um anel comutativo no qual cada elemento não nulo possui um inverso multiplicativo.

Considerando estruturas algébricas com dois conjuntos e ao menos duas operações binárias, destacamos algumas estruturas, como:

- **Módulo:** um módulo, ou **R -módulo** (à esquerda) M sobre um anel R é um grupo abeliano M com um mapeamento $\phi : R \times M \rightarrow M$ satisfazendo:
 - Identidade para multiplicação;
 - Distributividade da multiplicação em relação à soma;
 - Associatividade da multiplicação.

Um *módulo* é comutativo à direita quando é definido de modo análogo, porém com os elementos de R multiplicados pela direita ($\phi : M \times R \rightarrow M$).

Um **bimódulo** é um módulo que é um R -módulo à esquerda e à direita, de tal modo que o resultado das multiplicações são os mesmos (embora isso não exija necessariamente que a multiplicação seja comutativa, como ocorre no caso de espaços matriciais).

Se o anel R é comutativo, então os R -módulos à esquerda são o mesmo que os R -módulos à direita, e são denominados simplesmente de R -módulos.

- **Espaço vetorial:** um R -módulo no qual o anel R é um campo.

O meio mais comum de pensar as estruturas é na forma de conjuntos numéricos mas, de fato, muitas delas representam espaços de funções e, do ponto de vista matemático, quaisquer espaços com quaisquer elementos e operações poderiam ser representados dessa forma. Esta abordagem é possivelmente uma das mais interessantes, por sua força, generalidade e aplicabilidade.

2.3.3 Espaços vetoriais polinomiais

Como mencionamos anteriormente, espaços vetoriais são estruturas algébricas compostas por dois conjuntos e duas operações algébricas. No caso de espaços vetoriais polinomiais, o espaço vetorial polinomial de grau n (P^n) é formado pelo conjunto dos polinômios de grau menor ou igual a n , um conjunto de escalares ($\mathfrak{R}$), bem como duas operações: a soma de polinômios e a multiplicação por escalares, obedecendo aos seguintes axiomas:

1. **Comutatividade** em relação à soma, ou seja, $\forall \vec{u}, \vec{v} \in P^n$, $\vec{u} + \vec{v} = \vec{v} + \vec{u}$;
2. **Associatividade** em relação à soma, ou seja, dados três polinômios, $\forall \vec{u}, \vec{v}, \vec{w} \in P^n$, $(\vec{u} + \vec{v}) + \vec{w} = \vec{u} + (\vec{v} + \vec{w})$;
3. **Identidade** em relação à soma, ou seja, $\forall \vec{u} \in P^n$, $\exists \vec{0} \in P^n$ tal que $\vec{u} + \vec{0} = \vec{u}$ ($\vec{0}$ é o polinômio nulo);
4. **Invertibilidade** em relação à soma, ou seja, $\forall \vec{u} \in P^n$, $\exists -\vec{u} \in P^n$ tal que $\vec{u} + (-\vec{u}) = \vec{0}$;
5. **Distributividade** do produto por escalar em relação à soma vetorial, ou seja, $\forall \vec{u}, \vec{v} \in P^n$ e $\forall a \in \mathfrak{R}$, $a(\vec{u} + \vec{v}) = a\vec{u} + a\vec{v}$;
6. **Distributividade** da soma de escalares em relação ao produto por escalar, ou seja, $\forall \vec{u} \in P^n$ e $\forall a, b \in \mathfrak{R}$, $(a + b)\vec{u} = a\vec{u} + b\vec{u}$;
7. **Associatividade** do produto por escalar, ou seja, $\forall \vec{u} \in P^n$ e $\forall a, b \in \mathfrak{R}$, $(ab)\vec{u} = a(b\vec{u})$;
8. **Identidade** em relação ao produto por escalar, ou seja, $\forall \vec{u} \in P^n$, $1\vec{u} = \vec{u}$;

No âmbito desse espaço, se considerarmos um vetor de coeficientes $C = (c_0, c_1, \dots, c_n)$ e uma de suas bases, $B^n(x) =$

$(b_0^n(x), b_1^n(x), \dots, b_n^n(x))^{14}$, um polinômio genérico pode ser expresso como:

$$p^n(x) = \langle C, B^n(x) \rangle = c_0 b_0^n(x) + c_1 b_1^n(x) + \dots + c_n b_n^n(x) \quad (2.16)$$

$$p^n(x) = C^T \cdot B^n(x) = [c_0 \ c_1 \ \dots \ c_n] \begin{bmatrix} b_0^n(x) \\ b_1^n(x) \\ \vdots \\ b_n^n(x) \end{bmatrix} \quad (2.17)$$

Assumindo a base canônica como $B^n(x) = (1, x, x^2, \dots, x^n)$, então podemos representar um polinômio qualquer como:

$$p^n(x) = \sum_{i=0}^n c_i x^i \quad (2.18)$$

$$p^n(x) = C^T B^n(x) = \langle C, B^n(x) \rangle = c_0 + c_1 x + \dots + c_n x^n. \quad (2.19)$$

É interessante notar que, como qualquer outro espaço vetorial, podemos representar os polinômios com outras bases, e estabelecer mudanças de bases para mudar as representações. Como exemplo, consideremos o espaço dos polinômios de grau 3 (P^3). Um polinômio, variedade ou vetor desse espaço poderia ter a seguinte representação:

$$p^3(x) = CB^3(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3 = [c_0 \ c_1 \ c_2 \ c_3] \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \end{bmatrix}. \quad (2.20)$$

¹⁴Nesse caso, $b_i^n(x)$ é o i -ésimo polinômio da base, e cada $b_i^n(x)$, $i = 0, 1, \dots, n$, é um polinômio de grau menor ou igual a n .

Há diversas bases válidas para esse espaço, e como exemplo, tomemos uma base de **Hermite** [31]. Uma base de grau n de Hermite, $H^n(x)$, é um conjunto de $n + 1$ polinômios de Hermite de grau menor ou igual a n (tipicamente $H^n(x) = (h_0(x), h_1(x), \dots, h_n(x))$), definidos como:

$$h_n(x) = (-1)^n e^{x^2/2} \frac{d^n(e^{-x^2/2})}{dx^n}, \quad n = 1, 2, 3, \dots \quad (2.21)$$

que possui a seguinte expansão em série:

$$h_n(x) = \sum_{i=0}^{\infty} (-1)^i \frac{n^{[2i]}}{2^i i!} x^{n-2i} = x^n - \frac{n^{[2]}}{2 \cdot 1!} x^{n-2} + \frac{n^{[4]}}{2^2 \cdot 2!} x^{n-4} - \frac{n^{[6]}}{2^3 \cdot 3!} x^{n-6} + \dots, \quad (2.22)$$

com¹⁵ $n^{[r]} = n(n-1)(n-2)\dots(n-r+1)$. Os sete primeiros polinômios de Hermite são:

$$h_0(x) = 1,$$

$$h_1(x) = x,$$

$$h_2(x) = x^2 - 1,$$

$$h_3(x) = x^3 - 3x,$$

$$h_4(x) = x^4 - 6x^2 + 3,$$

$$h_5(x) = x^5 - 10x^3 + 15x,$$

$$h_6(x) = x^6 - 15x^4 + 45x^2 - 15,$$

⋮

e os primeiros cinco deles podem ser visualizados na Figura 2.3¹⁶. Utilizando a base cúbica de Hermite, $H^3(x) = (1, x, x^2 - 1, x^3 - 3x)$, podemos reescrever nosso polinômio cúbico em termos dessa base, partindo da sua representação na base canônica e utilizando a matriz

¹⁵A expressão $n^{[r]} = n(n-1)(n-2)\dots(n-r+1)$ representa a permutação de n , r a r , ou seja, $n^{[r]} = P(n, r) = n!/(n-r)!$.

¹⁶Vale realçar que não se deve confundir os polinômios de Hermite (ou quaisquer outros polinômios ou bases polinomiais) com as funções de mistura (*blending*) das *B-Splines*, que representam um conceito diferente, como veremos mais adiante.

Figura 2.3: Os cinco primeiros polinômios de Hermite.

mudança de base para representá-lo em termos da nova base. Abaixo apresentamos duas transições entre essas bases, explicitando as matrizes de mudança de base nas relações abaixo pela sua atuação sobre a base canônica¹⁷:

$$\begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 3 & 0 & 1 \end{pmatrix} \begin{bmatrix} 1 \\ x \\ x^2 - 1 \\ x^3 - 3x \end{bmatrix} \quad (2.23)$$

e

$$\begin{bmatrix} 1 \\ x \\ x^2 - 1 \\ x^3 - 3x \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -3 & 0 & 1 \end{pmatrix} \begin{bmatrix} 1 \\ x \\ x^2 \\ x^3 \end{bmatrix} \quad (2.24)$$

Substituindo a base canônica pela matriz mudança de base e a nova base (Equação 2.23) em 2.20, podemos reescrever o polinômio

¹⁷O procedimento de se encontrar as matrizes de mudança de base, bem como as matrizes de transformações lineares se encontra detalhadamente descrito em [32].

cúbico como:

$$p^3(x) = Cb^3(x) = [c_0 \ c_1 \ c_2 \ c_3] \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 3 & 0 & 1 \end{pmatrix} \begin{bmatrix} 1 \\ x \\ x^2 - 1 \\ x^3 - 3x \end{bmatrix} \quad (2.25)$$

Fazendo $[c_0 \ c_1 \ c_2 \ c_3] \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 3 & 0 & 1 \end{pmatrix} = [c'_0 \ c'_1 \ c'_2 \ c'_3]$, então temos

$$p^3(x) = [c'_0 \ c'_1 \ c'_2 \ c'_3] \begin{bmatrix} 1 \\ x \\ x^2 - 1 \\ x^3 - 3x \end{bmatrix}, \quad (2.26)$$

recaindo na equação 2.16 ou 2.17. Desse modo, temos o nosso polinômio representado em termos da base de Hermite, e diversas outras bases polinomiais podem ser utilizadas, diretamente ou por meio do mesmo processo¹⁸.

Dentre os conjuntos de polinômios que podem ser utilizados para construir tais bases, estão os polinômios Legendre, Jacobi, Chebyshev e Laguerre, que são todas bases polinomiais ortogonais¹⁹ [31].

¹⁸Nesse caso, podemos representar nossos vetores, ou polinômios *diretamente* na base desejada, ou representá-lo em termos da base canônica $(1, x, \dots, x^n)$ e efetuarmos a mudança de base dessa para a base desejada, como exemplificado no texto.

¹⁹Nesse espaço, o conceito de ortogonalidade é análogo à ortogonalidade entre vetores no espaço tridimensional, sendo definido como um produto interno nulo, ou seja, se f e g são duas funções contínuas em um intervalo $[a, b]$, o produto interno

A mudança de base pode ser desejável e mesmo necessária, de modo a moldar a representação do polinômio resultante à conveniência do problema a ser abordado, e não altera de nenhum modo o problema em si, já que todas as bases válidas de um espaço o representam integralmente. A ortogonalidade das bases acima citadas é interessante por possibilitar representações mínimas das variedades polinomiais nas respectivas bases, já que suas componentes são linearmente independentes. Obviamente, bases não ortogonais podem também ser utilizadas, porém elas aumentam a complexidade da representação dos polinômios a serem abordados.

2.4 Interpolação e Aproximação

2.4.1 Introdução

Interpolação é o processo de se encontrar uma função f tal que, dado um conjunto de pontos em um espaço n -dimensional, f **interpola** esses pontos, ou seja, para cada um desses pontos p_i , a imagem $f_i = f(p_i)$ faz parte de $Im(f)$, o conjunto imagem de f .

As técnicas de interpolação são tão importantes e necessárias na vida prática da Humanidade que antecederam até mesmo à formalização do **método científico**²⁰, pilar da ciência moderna e da sistematização do conhecimento.

com relação à função peso contínua $w(x)$ não negativa e que não se anula completamente em nenhum subintervalo de $[a, b]$, definido como

$$(f \cdot g)_w = \int_a^b f(x)g(x)w(x)dx$$

é nulo, ou seja, $(f \cdot g)_w = 0$. Cada base pode ser ortogonal em relação à uma função peso diferente, o que demanda certo cuidado na escolha da base, bem como na análise de suas propriedades.

²⁰O **método científico**, que fundamenta a ciência atual, surgiu a partir das ideias de Roger e Francis Bacon (séc. XVI) – de que o saber provinha da razão e da experimentação – e foi consolidado por Descartes, na obra “Discurso do Método”. Nesta obra, Descartes descreve sua abordagem para a produção de conhecimento dito científico, e acaba por estabelecer um padrão de critérios balizadores para a investigação científica. O “Método cartesiano” ou “Determinismo mecanicista” é empregado até a atualidade, tendo sido reavaliado por Karl Popper, que afirmou que a ‘verdadeira ciência’ é aquela na qual as teorias científicas devem ser passíveis de serem refutadas [33, 34].

Movidas pela necessidade, as comunidades humanas buscaram na “astrologia”²¹ os meios de se otimizar o plantio. Surgiram então os calendários e com eles, as técnicas de interpolação, pois era necessário se calcular as posições de astros a fim de se ‘preverem’ as estações. Os registros mais antigos do uso dessa técnica datam de 300 a.C. (mais detalhes no **Apêndice A**).

Por muito tempo, a interpolação vinha sendo um problema muito popular entre os matemáticos, pois muitas aplicações práticas demandavam tal abordagem. No entanto, no final do século XIX, buscaram-se novas formas de se descrever dados, como representações não-interpolantes, a fim de se representar estatisticamente conjuntos de dados em termos de suas medidas-resumo como, por exemplo, seus **momentos** (média, variância, etc). Uma possível justificativa para essa mudança de abordagem é que, nesse período (séculos XVIII e XIX) expandia-se pelo mundo a revolução industrial, e a demanda por melhoria nos processos produtivos e métricas de qualidade serviu como estímulo aos estudos acadêmicos na busca de soluções práticas nesse âmbito. Ganharam então novos impulsos a Estatística e os métodos da Teoria da Aproximação, como resposta a essa demanda, compartilhando de diversos fundamentos com a Teoria da Interpolação, voltada naquele momento não a funções que *interpolassem* os dados, mas àquelas que os *representassem* melhor, que os *aproximassem* melhor. No presente trabalho, embora possamos nos referir a resultados e teoremas frequentemente associados à Teoria da Aproximação, nossa abordagem se concentra sobretudo nas técnicas de interpolação.

2.4.2 Splines

Em Matemática, uma **spline** representa uma função matemática constituída por partes, um conjunto de funções definidas sobre um dado domínio. Por meio do particionamento de um certo intervalo no domínio de validade da *spline*, atribui-se uma **subfunção**, que é de-

²¹Inicialmente, não havia distinção entre astrologia e astronomia.

finida sobre cada uma dessas partições, de modo que a função *spline* seja a união de todas essas subfunções em todo o intervalo da *spline*. O grau da *spline* é o grau máximo que cada uma dessas subfunções pode assumir. Assim, uma *spline* de grau 3 pode ter subfunções de graus 0, 1, 2 ou 3.

A *spline* foi definida originalmente como o que hoje conhecemos por “*spline* polinomial”, devido ao fato de que cada subfunção que formava a *spline* era um polinômio. No entanto, existem formulações não polinomiais, como por exemplo, as “*splines* trigonométricas” [35]. Formalmente falando, se tomarmos um intervalo $[a, b]$ particionado em $m + 1$ trechos, de tal modo que $a = \tau_0 < \tau_1 < \dots < \tau_m < \tau_{m+1} = b$ sejam pontos de partição em $[a, b]$, uma *spline* $S^n(x)$ de grau n com **nós** nos pontos $\tau_1, \tau_2, \dots, \tau_m$, é uma função constituída por partes que possui as seguintes propriedades:

1. $S^n(x)$ é um polinômio de grau menor ou igual a n em cada subintervalo $[\tau_{i-1}, \tau_i]$, com $1 \leq i \leq m + 1$;
2. $S^n(x)$ possui derivadas contínuas até a ordem $n - 1$ em $[a, b]$.

Obviamente, como a *spline* é um conjunto de funções, dizer que o grau de $S^n(x)$ é menor ou igual a n é dizer o mesmo sobre o grau de cada subfunção $s_i^n(x)$ que a compõe. Assim, em cada trecho $[\tau_{i-1}, \tau_i]$ ²², a *spline* $S^n(x)$ é representada pelas subfunções $s_i^n(x)$ dos trechos correspondentes ($S^n(x) = s_i^n(x)$). E sem perda de generalidade, podemos assumir que os intervalos a serem adotados para o parâmetro são igualmente espaçados²³, com $x \in [0, 1]$, de modo que, para cada trecho da

²²Existem várias combinações possíveis para esse intervalo. Em geral, escolhe-se um dos nós (à esquerda ou à direita) que estará incluído no intervalo, e outro não ($[\tau_{i-1}, \tau_i[$ ou $] \tau_{i-1}, \tau_i]$), de modo a não existirem duas imagens para um mesmo valor do domínio.

²³Se os intervalos não forem igualmente espaçados, é possível realizar uma reparametrização desse intervalo unitário, de modo que se ajuste a quaisquer comprimentos desejados. Se considerarmos a i -ésima partição (correspondente ao i -ésimo trecho da *spline*) como iniciando em x_{i-1} e terminando em x_i , então podemos criar um novo parâmetro t , tal que $t = \frac{x - x_{i-1}}{x_i - x_{i-1}}$, e, ao invés de utilizar $s_i^n(x)$ para $x \in [a, b]$, utilizamos $s_i^n(t)$, com $t \in [0, 1]$ para cada trecho.

spline, $0 \leq x \leq 1$.

Genericamente, qualquer conjunto de polinômios de grau n válidos na partição poderiam constituir uma *spline*, mesmo que não houvesse nenhuma relação entre seus diferentes trechos (como exemplificado na Figura 2.4). É interessante notar a simplicidade e generalidade de tal definição, donde provém sua verdadeira força, pois dela surgiram inúmeras classes de *splines*, em diversas dimensões e para diferentes propósitos. Essa definição estabelece uma superclasse de funções que podem ser ditas *spline*, mantendo ainda a possibilidade de infinitas novas subclasses. Um exemplo disso é que não há restrições quanto à continuidade das *splines*.

Uma *spline* polinomial de grau 0 teria, como subfunção $s_i^n(x)$, um número (real ou complexo), e, possivelmente diferentes números a cada intervalo, constituindo assim uma função constituída por patamares, ou 'degraus'. Splines de graus maiores podem adicionar, dependendo do modo como fo-

Figura 2.4: Exemplo de uma *spline* e seus trechos.

rem definidas, novos graus de continuidade (e portanto, gerando curvas mais suaves), e assim por diante. A definição nem ao menos fixa critérios de conexão entre as subfunções da *spline*, de interpolação nem de aproximação, o que nos leva a inferir que quaisquer conjuntos de funções que atendam apenas simplesmente aos dois critérios acima, com quaisquer regras de formação ou arranjo, constituam de fato uma *spline*.

2.4.3 B-Splines

B-Splines são uma categoria de *splines* cujas subfunções são definidas em termos de bases de funções. A inovação, nesse caso, consiste em interpretar as subfunções (que são polinômios, no caso da *spline* polinomial) como um elemento de um espaço vetorial de funções, por exemplo, o espaço vetorial dos polinômios de grau n , P^n . De modo análogo ao que temos em Álgebra Linear, na qual espaços vetoriais são os conjuntos de vetores, juntamente com as regras de operação entre eles, aqui também, os espaços de funções são tratados como espaços vetoriais, e a lógica das operações vetoriais, adaptada a esses espaços, fornece ferramentas práticas e poderosas de análise e representação de dados, regras e situações²⁴.

Considerando $B^n(x) = (b_0^n(x), b_1^n(x), \dots, b_n^n(x))$ uma base qualquer²⁵ de P^n , e um vetor de coeficientes $C = (c_0, c_1, \dots, c_n)$, uma *B-Spline* polinomial $S^n(x)$ de grau n é composta por funções dadas por uma das representações²⁶ abaixo (todas equivalentes):

$$S_i^n(x) = \langle C, B^n(x) \rangle = c_0 b_0^n(x) + c_1 b_1^n(x) + \dots + c_n b_n^n(x) \quad (2.27)$$

$$S_i^n(x) = C^T \cdot B^n(x) = [c_0 \ c_1 \ \dots \ c_n] \begin{bmatrix} b_0^n(x) \\ b_1^n(x) \\ \vdots \\ b_n^n(x) \end{bmatrix} \quad (2.28)$$

²⁴A expressividade de tal representação jamais pode ser subestimada, pois essa generalização pode ampliar significativamente o modo com o qual os modelos são definidos, e de fato o fazem. Vide a sua utilização nos ramos mais avançados da Física, como a Mecânica Quântica.

²⁵Tal base deveria ter, obrigatoriamente, $n + 1$ componentes linearmente independentes, para gerar todo o espaço P^n .

²⁶Pelo fato de que as *splines* são conjuntos de subfunções, o i -ésimo trecho (ou componente) da *spline* também terá essas mesmas formulações.

$$s_i^n(x) = \sum_{k=0}^n c_k \cdot b_k^n(x) \quad (2.29)$$

$$s_i^n(x) = c \cdot b^n(x) \quad (2.30)$$

nas quais $b_k^n(x)$ representa o k -ésima componente da base. A equação 2.27 foi definida em termos do produto interno, enquanto que a 2.28 utiliza a notação matricial. Já a equação 2.30 utiliza a notação tensorial, com omissão de sub-índices (notação de Einstein), obtida a partir da representação em forma de somatório da equação 2.29.

A fim de exemplificar os conceitos apresentados, tomemos uma base de Hermite de grau 3, dada por $H^3(x) = (1, x, x^2 - 1, x^3 - 3x)$, e busquemos encontrar os coeficientes $C = (c_0, c_1, \dots, c_n)$ tais que

$$s_i^n(x) = C^T \cdot H^3(x) = [c_0 \ c_1 \ c_2 \ c_3] \begin{bmatrix} 1 \\ x \\ x^2 - 1 \\ x^3 - 3x \end{bmatrix}. \quad (2.31)$$

Sem nenhuma restrição, os c_k , $k = 0, \dots, n$, podem ser quaisquer. E é aqui que surge o ponto de liberdade criativa da ferramenta: por meio das restrições impostas na busca dos valores das constantes c_k , diversas características diferentes podem ser adicionadas às curvas *spline*, influenciando decisivamente em suas estruturas e comportamentos, e capazes de originar infinitas *splines*. Tais restrições podem ser as mais diversas, e algumas delas são analisadas a seguir.

No caso da abordagem de Hermite, as restrições consistem em definir os valores dos pontos no início e fim de cada *trecho*, bem como suas derivadas nos pontos de junção, ou nós. Como exemplo, consideremos os pontos $D = \{(0; 0), (1/6; 0, 140886), (1/3; 0, 304617), (1/2; 0, 308425), (2/3; 0, 243694), (5/6; 0, 201809), (1; 0)\}$, amostra-

dos sobre a **função de Weierstrass**²⁷, destacados na Figura 2.5.

Figura 2.5: Um exemplo de pontos a serem interpolados, para $\alpha=3$ ($W_3(x)$).

Nesse exemplo, o intuito é criar uma *spline* cúbica que os interpole. Para cada trecho da *spline*, temos uma equação do tipo²⁸

$$s_i^n(t) = [c_0 \ c_1 \ c_2 \ c_3] \begin{bmatrix} 1 \\ t \\ t^2 - 1 \\ t^3 - 3t \end{bmatrix} = c_0 + c_1 t + c_2(t^2 - 1) + c_3(t^3 - 3t). \quad (2.32)$$

Seguindo a abordagem de Hermite, temos as restrições de contorno:

²⁷Essa função, definida como $W_\alpha(x) = \sum_{k=0}^{\infty} \frac{\sin(\pi k^\alpha x)}{\pi k^\alpha}$, inicialmente para $\alpha = 2$ e k natural, apesar de ser referenciada por muitos como ‘função de Weierstrass’, foi na verdade proposta por Riemann em 1861 em suas palestras, que afirmou que ela era contínua em todos os pontos, porém não diferenciável em um conjunto denso nos reais. Ela possui derivada para os valores de x pertencentes ao conjunto dos números ímpares. Posteriormente, provou-se que ela não é diferenciável para qualquer valor irracional [36].

²⁸No caso, estamos utilizando a reparametrização da seção anterior, $t = \frac{x - x_i}{x_{i+1} - x_i}$, para $t \in [0, 1]$.

$$s_i^n(0) = P_0, \quad s_i^n(1) = P_F, \quad \frac{d}{dt}s_i^n(0) = V_0, \quad \frac{d}{dt}s_i^n(1) = V_F, \quad (2.33)$$

na qual P_0 é o valor inicial da *spline* quando $t = 0$, P_F é o valor quando $t = 1$, V_0 é o valor da derivada no ponto P_0 quando $t = 0$, e V_F é o valor da derivada no ponto P_F quando $t = 1$. A definição das constantes P_0, P_F, V_0 e V_F representa uma **parametrização**²⁹ da *solução* do problema, de modo que não precisemos resolver novamente o sistema para todos os trechos, mas simplesmente possamos substituir os valores desejados (vetor de restrições) na equação do trecho correspondente, e assim obter a *spline* no domínio $[a, b]$. Temos então, o seguinte sistema para cada subfunção:

$$\left\{ \begin{array}{ll} s_i^n(0) = c_0 - c_2 = P_0 & c_0 = -2P_0 + 3P_F - 2V_0 - V_F \\ s_i^n(1) = c_0 + c_1 - 2c_3 = P_F & c_1 = 6P_0 - 6P_F + 4V_0 + 3V_F \\ D_t s_i^n(0) = c_1 - 3c_3 = V_0 & c_2 = -3P_0 + 3P_F - 2V_0 - V_F \\ D_t s_i^n(1) = c_1 + 2c_2 = V_F & c_3 = 2P_0 - 2P_F + V_0 + V_F \end{array} \right. \rightarrow \quad (2.34)$$

e então,

$$s_i^n(t) = [P_0 \ P_F \ V_0 \ V_F] \begin{pmatrix} -2 & 6 & -3 & 2 \\ 3 & -6 & 3 & -2 \\ -2 & 4 & -2 & 1 \\ -1 & 3 & -1 & 1 \end{pmatrix} \begin{bmatrix} 1 \\ t \\ t^2 - 1 \\ t^3 - 3t \end{bmatrix} \quad (2.35)$$

Essa é a representação da solução do sistema 2.34 na base de Hermite, porém, a fim de simplificarmos os cálculos computacionais, podemos utilizar a base canônica, e condensar todas essas mudanças numa matriz de transformação, reescrevendo a base utilizando a equação 2.24 e obtendo

²⁹Parametrização é o processo de generalização de modelos, objetos, métodos ou comportamentos por meio da definição de parâmetros relevantes ao contexto.

$$s_i^n(t) = [P_0 \ P_F \ V_0 \ V_F] \begin{pmatrix} -2 & 6 & -3 & 2 \\ 3 & -6 & 3 & -2 \\ -2 & 4 & -2 & 1 \\ -1 & 3 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -3 & 0 & 1 \end{pmatrix} \begin{bmatrix} 1 \\ t \\ t^2 \\ t^3 \end{bmatrix}$$

resultando em

$$s_i^n(t) = \underbrace{[P_0 \ P_F \ V_0 \ V_F]}_{R_H} \underbrace{\begin{pmatrix} 1 & 0 & -3 & 2 \\ 0 & 0 & 3 & -2 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & -1 & 1 \end{pmatrix}}_{M_H} \underbrace{\begin{bmatrix} 1 \\ t \\ t^2 \\ t^3 \end{bmatrix}}_{B_{can}(t)} \quad (2.36)$$

$$s_i^n(t) = R_H \cdot M_H \cdot B_{can}(t)$$

na qual o vetor R_H representa o vetor de restrições de Hermite, a matriz M_H representa a matriz característica de Hermite³⁰, e $B_{can}(t)$ a base canônica dos polinômios cúbicos. Alguns autores representam a equação 2.36 de modo diferente, motivo pelo qual apresentamos mais duas outras versões dela³¹:

$$s_i^n(t) = [P_0 \ P_F \ V_0 \ V_F] \begin{pmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{pmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix} \quad (2.37)$$

³⁰Esta e outras matrizes características, como Hermite e Bézier, representam a transformação que leva os parâmetros de entrada (ou, o chamado vetor de restrições) no espaço dos coeficientes c_k , que constam nas equações 2.27 a 2.30. Representada na forma matricial, a solução torna-se parametrizada em termos dos vetores de restrição, evitando a necessidade de se recalcular a solução do sistema a cada trecho da *spline*.

³¹Tais versões decorrem da comutatividade do produto interno usual em $\mathfrak{R}^4$.

e

$$s_i^n(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \begin{bmatrix} P_0 \\ P_F \\ V_0 \\ V_F \end{bmatrix} \quad (2.38)$$

Uma vez definida essa solução paramétrica, podemos determinar as equações de cada trecho da *spline*, definindo para cada um deles o vetor de restrições R_H correspondente, em termos da matriz de pontos D (Figura 2.5) e dos valores das derivadas em cada um dos nós da *spline*. Para montar a interpolação, podemos impor que as derivadas sejam nulas, resultando na curva da Figura 2.6.

Figura 2.6: Interpolação sobre $W_3(x)$ com os valores das derivadas nulos nos nós.

Poderíamos mudar as condições como, por exemplo, iniciar a derivada V_0 de cada trecho com zero, e terminando em $sen(i)$, no qual i é o índice do nó final do trecho, resultando na *spline* da Figura 2.7.

As funções de mistura, ou funções de *blending* surgem quando buscamos determinar a influência dos parâmetros na curva, ao invés de olharmos apenas sua definição em termos da base. Tais funções definem em que medida os parâmetros (ou pontos de controle) influenciam a forma da curva final ou, em outras palavras, qual a parcela

Figura 2.7: Interpolação sobre $W_3(x)$ com os valores das derivadas valendo $V_0 = 0$, $V_F = \sin(i)$, para cada trecho da spline.

de contribuição de cada um deles na curva final, ou ainda, seu fator de ponderação na constituição da *spline* final. A equação dessas funções é obtida quando reagrupamos a equação do trecho em função dos pontos de controle, como na equação abaixo:

$$s_i^n(t) = P_0(1 - 3t^2 + 2t^3) + P_F(3t^2 - 2t^3) + V_0(t - 2t^2 + t^3) + V_F(-t^2 + t^3) \quad (2.39)$$

O gráfico dessas funções para a abordagem de Hermite é apresentado na Figura 2.8, com a legenda dos respectivos parâmetros de ponderação:

É importante notar que essas funções não correspondem à base na qual a curva está representada, mas sim ao fator de ponderação do vetor de restrições e sua influência sobre a curva, na constituição da curva *spline* no trecho que corresponde à uma partição do domínio (no caso, um trecho k , delimitado por $[\tau_{k-1}, \tau_k]$, $k = 1, 2, \dots, m + 1$).

2.4.4 Splines bidimensionais

A beleza da formulação matemática das *splines*, principalmente das *B-Splines*, mostra seu valor prático quando buscamos uma expansão na formulação, obtendo *splines* de dimensão maior, como superfícies e objetos tridimensionais. Tais entidades podem ser obtidas a par-

Figura 2.8: Funções de mistura da spline de Hermite interpolante.

tir do produto tensorial³² entre *splines* unidimensionais, com ajustes menores na formulação, como no caso do vetor ou matriz de restrições para o caso bidimensional. Desse modo, uma superfície spline $\sigma^n(t, u)$ poderia ser definida como $\sigma^n(t, u) = S^n(t).S^n(u)$. Essa definição vale para qualquer spline, porém exemplificaremos seu uso com *splines* de Hermite. No caso de uma superfície, fazer o produto de *splines* corresponde a fazer o produto de suas subfunções. Considerando uma spline

³²Nesse caso, cada *spline* pode ser vista como um *tensor*, escrita em um espaço polinomial, e seu produto com outras resulta em uma variedade linear de dimensão maior, a depender de quantas *splines* estão sendo multiplicadas. Na prática, um produto tensorial é uma operação matemática cujo significado não é necessariamente definido em termos físicos, mas que pode ser definido em termos de um contexto específico. Portanto, há outros modos de se formular *splines*, operando também com produtos tensoriais, como no caso das “superfícies spline triangulares”, não abordadas aqui por não constituírem o foco do trabalho.

de grau n com subfunção

$$s_i^n(t) = \underbrace{\begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix}}_{B_{can}^T(t)} \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{pmatrix}}_{M_H^T} \underbrace{\begin{bmatrix} P_{0_t} \\ P_{F_t} \\ V_{0_t} \\ V_{F_t} \end{bmatrix}}_{R_{H_t}^T} \quad (2.40)$$

$$s_i^n(t) = B_{can}^T(t) \cdot M_H^T \cdot R_{H_t}^T$$

e outra de mesmo grau com subfunção

$$s_i^n(u) = \underbrace{\begin{bmatrix} P_{0_u} & P_{F_u} & V_{0_u} & V_{F_u} \end{bmatrix}}_{R_{H_u}} \underbrace{\begin{pmatrix} 1 & 0 & -3 & 2 \\ 0 & 0 & 3 & -2 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & -1 & 1 \end{pmatrix}}_{M_H} \underbrace{\begin{bmatrix} 1 \\ u \\ u^2 \\ u^3 \end{bmatrix}}_{B_{can}(u)} \quad (2.41)$$

$$s_i^n(u) = R_{H_u} \cdot M_H \cdot B_{can}(u)$$

então a superfície será dada por

$$\sigma^n(t, u) = S^n(t) \cdot S^n(u) \quad (2.42)$$

$$\sigma^n(t, u) = B_{can}^T(t) \cdot M_H^T \cdot R_{H_t}^T \cdot R_{H_u} \cdot M_H \cdot B_{can}(u)$$

fazendo $R_{2D} = R_{H_t}^T \cdot R_{H_u}$ a matriz de restrições da superfície, temos

$$\sigma^n(t, u) = B_{can}^T(t) \cdot M_H^T \cdot R_{2D} \cdot M_H \cdot B_{can}(u) , \quad (2.43)$$

sendo a matriz de restrições R_{2D} dada por

$$R_{2D} = \begin{bmatrix} P_{0t} \\ P_{Ft} \\ V_{0t} \\ V_{Ft} \end{bmatrix} [P_{0u} \ P_{Fu} \ V_{0u} \ V_{Fu}] = \begin{pmatrix} P_{0t}P_{0u} & P_{0t}P_{Fu} & P_{0t}V_{0u} & P_{0t}V_{Fu} \\ P_{Ft}P_{0u} & P_{Ft}P_{Fu} & P_{Ft}V_{0u} & P_{Ft}V_{Fu} \\ V_{0t}P_{0u} & V_{0t}P_{Fu} & V_{0t}V_{0u} & V_{0t}V_{Fu} \\ V_{Ft}P_{0u} & V_{Ft}P_{Fu} & V_{Ft}V_{0u} & V_{Ft}V_{Fu} \end{pmatrix}$$

$$= \begin{pmatrix} S_t^n(0)S_u^n(0) & S_t^n(0)S_u^n(1) & S_t^n(0)\frac{\partial S_u^n(0)}{\partial u} & S_t^n(0)\frac{\partial S_u^n(1)}{\partial u} \\ S_t^n(1)S_u^n(0) & S_t^n(1)S_u^n(1) & S_t^n(1)\frac{\partial S_u^n(0)}{\partial u} & S_t^n(1)\frac{\partial S_u^n(1)}{\partial u} \\ \frac{\partial S_t^n(0)}{\partial t}S_u^n(0) & \frac{\partial S_t^n(0)}{\partial t}S_u^n(1) & \frac{\partial S_t^n(0)}{\partial t}\frac{\partial S_u^n(0)}{\partial u} & \frac{\partial S_t^n(0)}{\partial t}\frac{\partial S_u^n(1)}{\partial u} \\ \frac{\partial S_t^n(1)}{\partial t}S_u^n(0) & \frac{\partial S_t^n(1)}{\partial t}S_u^n(1) & \frac{\partial S_t^n(1)}{\partial t}\frac{\partial S_u^n(0)}{\partial u} & \frac{\partial S_t^n(1)}{\partial t}\frac{\partial S_u^n(1)}{\partial u} \end{pmatrix}$$

e desse modo

$$R_{2D} = \begin{pmatrix} \sigma^n(0,0) & \sigma^n(0,1) & \frac{\partial \sigma^n(0,0)}{\partial u} & \frac{\partial \sigma^n(0,1)}{\partial u} \\ \sigma^n(1,0) & \sigma^n(1,1) & \frac{\partial \sigma^n(1,0)}{\partial u} & \frac{\partial \sigma^n(1,1)}{\partial u} \\ \frac{\partial \sigma^n(0,0)}{\partial t} & \frac{\partial \sigma^n(0,1)}{\partial t} & \frac{\partial^2 \sigma^n(0,0)}{\partial t \partial u} & \frac{\partial^2 \sigma^n(0,1)}{\partial t \partial u} \\ \frac{\partial \sigma^n(1,0)}{\partial t} & \frac{\partial \sigma^n(1,1)}{\partial t} & \frac{\partial^2 \sigma^n(1,0)}{\partial t \partial u} & \frac{\partial^2 \sigma^n(1,1)}{\partial t \partial u} \end{pmatrix} \quad (2.44)$$

Na matriz de restrições R_{2D} temos quatro regiões distintas, ou blocos, descritas respectivamente da esquerda para a direita e de cima para baixo: o primeiro bloco refere-se às posições dos pontos que apoiarão a superfície, no segundo, as inclinações das derivadas com relação ao parâmetro u , no terceiro bloco, as mesmas inclinações mas em relação ao parâmetro t , e no último bloco, as derivadas segundas mistas em cada uma das extremidades da superfície (esses valores indicam as taxas de crescimento das derivadas primeiras da superfície em cada uma dessas extremidades). A superfície gerada é visualizada na figura 2.9.

(a) Superfície *spline* de Hermite com $\frac{\partial^2 \sigma(1, 0)}{\partial t \partial u} = 1$

(b) Superfície *spline* de Hermite com $\frac{\partial^2 \sigma(1, 0)}{\partial t \partial u} = -10$

(c) Superfície *spline* de Hermite com $\frac{\partial^2 \sigma(1, 0)}{\partial t \partial u} = 30$

(d) Superfície (c) vista de outro ângulo.

Figura 2.9: Exemplo de uma superfície de Hermite.

Note que, à medida que a derivada segunda se torna negativa, a concavidade para baixo se intensifica, e à medida que esta se torna positiva, a concavidade tende a ser para cima, comportamento análogo ao observado no caso da parábola.

Outro ponto importante é que não houve necessidade de se re-

solver novamente o sistema 2.30 para o caso bidimensional, nem de se determinar novamente a solução em termos da matriz de restrições. Ao invés disso, a notação do produto tensorial possibilitou o uso da solução já calculada previamente, apenas estendida à abordagem bidimensional, o que é bastante interessante do ponto de vista computacional, pois de fato, tal uso se estende a uma abordagem n-dimensional.

2.5 Paralelização de software: CUDA

2.5.1 Introdução

CUDA³³ é uma plataforma de desenvolvimento paralelo destinada a tirar proveito da grande capacidade de processamento dos processadores presentes nas placas de vídeo NVIDIA (**GPUs** - Graphic Processing Unit). A intensa demanda por utilização científica de processamento despertou a indústria para esse segmento. No início de 2000, quando as placas gráficas passaram a incorporar processadores programáveis para geração de gráficos, diversos pesquisadores, utilizando linguagens de programação como OpenGL e DirectX aproveitavam essa possibilidade para efetuar processamento científico³⁴, em um uso incomum do poder de computação das placas gráficas, voltada ao processamento genérico (como passou a ser denominado todo o tipo de cálculo não relacionado à finalidade gráfica). Em essência, os processadores das placas gráficas eram ‘enganados’, de modo que dados lhes eram passados, computados e armazenados em valores de pixel, que podiam então ser recuperados pelas equipes de pesquisa. Esse tipo de uso da GPU era extremamente tortuoso, apresentava várias dificuldades, como imprecisão, limitações, adaptação do algoritmo a um formato capaz de ‘enganar’ a GPU, comportamento no ponto flutuante

³³*Computer Unified Device Architecture*: Arquitetura Unificada de Dispositivo de Computação.

³⁴O processador da placa gráfica (GPU) passou a incorporar diversos subprocessadores programáveis, elaborados com a finalidade de acelerar a criação e exibição de conteúdo visual e produzir efeitos mais realistas em tempo real, sobretudo para jogos. Tais unidades foram denominadas de *pixel shaders*.

mudando de fabricante para fabricante, entre outros. Ainda assim, os experimentos mostravam um futuro promissor para esse tipo de computação (Figura 2.10).

Figura 2.10: Comparativo de desempenho entre CPUs e GPUs e suas taxas de transferência de memória (adaptado do site oficial da Nvidia).

Diante do esforço quase desesperado de diversos pesquisadores em aproveitar essa capacidade de computação, a consciência de um mercado latente se fez cada vez mais presente na mente dos fabricantes e desenvolvedores, tornando-os cada vez mais sensíveis aos apelos dessa comunidade. Nesse cenário, a NVIDIA lançou, em novembro de 2006, a primeira versão de arquitetura CUDA, com a GeForce 8800 GT. O mercado então observou uma rápida expansão do segmento, sobretudo no ramo científico, com aplicações em:

- *Simulações em Física:* fluidos, partículas, corpos rígidos, corpos flexíveis, tensões e esforços, gravitação, eletromagnetismo e ótica, testes de hipóteses, entre outros;
- *Biologia e medicina:* simulações de vírus, obtenção de imagens por ultrassom de alta velocidade, entre outros;
- *Simulações em química:* dinâmica molecular, reações entre diferentes compostos, entre outros;

- *Geologia*: cálculo de energias e esforços, simulações geológicas, entre outros;
- *Economia*: cálculo de riscos de carteiras, avaliação das condições de mercado, entre outros.

Pelo seu baixo custo/benefício, em razão da alta capacidade de processamento e elevada eficiência energética das placas gráficas, o processamento em GPU tem por vezes recebido a alcunha de ‘computação verde’, e a tendência dos fabricantes caminha no sentido de incorporar tais unidades aos processadores de uso geral, reduzindo consumo e aumentando a capacidade de processamento e o paralelismo real das unidades computacionais.

2.5.2 Arquitetura

A arquitetura CUDA explora intensamente o paralelismo, dado que o processamento gráfico é altamente paralelizável por natureza. Nesse caso, a ênfase recai sobre os cálculos, em detrimento da quantidade de memória e generalidade dos processadores (o que resulta em processadores mais simples, porém em grande quantidade, como mostra a Figura 2.11³⁵).

Além de privilegiar mais unidades para o processamento aritmético, demandas associadas foram aprimoradas nas GPUs, otimizando cada vez mais seu desempenho por meio da adição de diferentes níveis paralelizados de memória, gerenciamento dinâmico de *threads* por hardware, acessos mais rápidos à memória e diversos recursos para potencializar o paralelismo e a atuação sincronizada dos núcleos de computação, resultando em uma estrutura altamente escalável (Figuras 2.12 e 2.13).

³⁵Nesta figura, ULA significa Unidade Lógica Aritmética, DRAM é Dynamic Random Access Memory, ou Memória Dinâmica de Acesso Aleatório.

(a) Proporções de transistores por função na CPU.

(b) Proporções transistores por função na GPU.

Figura 2.11: Comparativo das proporções de unidades físicas por funcionalidade entre CPUs e GPUs (adaptado de [37]).

Figura 2.12: Arquitetura de memória (adaptado de [37]).

2.5.3 Linguagens

Dentre as linguagens de programação paralela que se destacaram, podemos citar:

- **MPI:** *Message Passing Interface* (Interface de transmissão de mensagens), nascida em 1992 e destinada a aplicações de supercomputação, especialmente para clusters;

Figura 2.13: Escalabilidade dos núcleos de processamento CUDA (adaptado de [38]).

- **OpenMP:** *Open Multiprocessing* (Multiprocessamento aberto), lançado pela organização não governamental *OpenMP Architecture Review Board - OpenMP ARB*, em outubro de 1997, consórcio representante das empresas AMD, IBM, Intel, Cray, HP, Fujitsu, Nvidia, NEC, Microsoft, Texas Instruments, Oracle Corporation, entre outras;
- **CUDA:** uma API (*Application Program Interface* - Interface de Programação de Aplicação) de alto nível para as GPUs da Nvidia, tornada pública em 15 de fevereiro de 2007;
- **OpenCL:** *Open Computing Language* (Linguagem de computação aberta), desenvolvida pela Apple (detentora dos direitos) e aperfeiçoada com a colaboração das equipes técnicas da AMD, IBM, Intel, e Nvidia. O grupo Khronos Compute Working, formado por representantes de fabricantes de CPU, GPU, microprocessadores e

companhias de software lançaram sua primeira versão aberta em 18 de novembro de 2008.

Essas linguagens impulsionaram e constituíram parte do ferramental utilizado nas pesquisas em processamento paralelo, umas mais que outras nessa fase de expansão da programação paralela.

2.5.4 Modelo de desenvolvimento

Em suma, uma aplicação CUDA corresponde a uma aplicação que possui parte de seu código sendo executada pela CPU e parte na GPU. Os dados necessários à computação na GPU são enviados e recuperados por meio da linguagem CUDA, como exemplificado na Figura 2.14³⁶. A esta estrutura podem ser agregadas tantas bibliotecas quanto se queira, tanto no nível do código executado na CPU quanto no trecho referente à GPU, desde que respeitadas as restrições de hardware tanto quanto à memória quanto armazenamento de dados.

Figura 2.14: Modelo de execução de uma aplicação CUDA.

No nível mais baixo do desenvolvimento, temos a linguagem CUDA, sua API³⁷ e o suporte a várias linguagens oferecido pelo fabricante. Sobre essa camada, há diversas bibliotecas, como cuFFT, cuBLAS, e muitas outras. Uma aplicação pode utilizar recursos de quais-

³⁶As taxas de transferência da Figura 2.14 foram medidas para uma Geforce 560 TI, uma placa de R\$ 700,00 e 384 núcleos CUDA.

³⁷API: *Application Program Interface*: Interface de Programação de Aplicação, conjunto de ferramentas utilizadas para se programar uma aplicação, geralmente relacionada a um determinado contexto ou tecnologia.

quer camada, de modo que sua estrutura pode assumir um padrão semelhante ao da Figura 2.15³⁸.

Figura 2.15: Modelo de aplicação CUDA.

A programação em CUDA exige que sejam escritos diversos fragmentos de código, cada um destinado a ser executado em um processador diferente, os quais são chamados de “*kernels*”. Há uma organização lógica das linhas de execução (também chamadas de “*threads*”), que permite sua indexação e, por conseguinte, que cada *kernel*, mesmo possuindo o mesmo fragmento de código, atue de modo diferente e específico para cada linha de execução, já que os índices que lhe são atribuídos via hardware são diferentes. Essa organização agrupa as *threads* em blocos (*blocks*), e os blocos são agrupados em estruturas denominadas “*grids*”. Um *kernel* pode ser designado para ser executado em um ou mais blocos e, dentro destes, por uma ou mais *threads*. Os *grids* são conjuntos bidimensionais de blocos, ao passo que os blocos são estruturas tridimensionais, resultando em uma estrutura 5-dimensional, como mostra a Figura 2.16.

Cada vez que um *kernel* é lançado (ou iniciado), o programador pode definir qual será a estrutura lógica de processamento a ser utilizada, podendo utilizar, no máximo, um bloco 5-dimensional de processadores. Obviamente, os acessos à memória local do *grid*, bem como à memória global da placa devem se dar de modo a preservar a correteza do algoritmo (como o acesso é concorrente, deve se assegurar

³⁸Nesta figura, de qualquer camada de software se pode acessar os recursos das camadas inferiores, direta ou indiretamente.

Figura 2.16: Bloco de processamento: cada kernel pode invocar um bloco de núcleos de processamento de até cinco dimensões.

o sincronismo entre as *threads*). Cada *thread* possui um espaço privado de memória, e cada bloco possui um espaço de memória privado, compartilhado somente entre as *threads* internas ao bloco, e os *grids*, bem como as *threads* do sistemas, podem acessar a memória global da placa de vídeo (Figura 2.17). A velocidade de acesso diminui à medida que avançamos dos níveis mais interiores (mais privados) para os níveis mais exteriores (memória globalmente compartilhada da placa de vídeo), e a adequada utilização dessas características pode impactar significativamente na execução do programa.

2.5.5 Interface de programação

As principais interfaces de programação e ferramentas disponibilizadas para desenvolvimento em CUDA são:

- **nvcc:** prepara código para compilação, e utiliza um compilador nativo para compilar. Dentre suas características, ele:

Figura 2.17: Modelo de memória de uma aplicação CUDA na GPU (adaptado de [38]).

- Utiliza a API CUDA;
- Suporta linguagens como C, C++, Fortran, OpenCL, OpenMP, entre outras;
- Utiliza C++ com restrições: Polimorfismo, valores padrão, sobrecarga de operadores, namespaces e classes (para GPUs com capacidade de computação 2.x).
- **CUDA Toolkit:** kit de ferramentas de desenvolvimento disponibilizado pela NVIDIA voltado a CUDA;
- **CUDA GDB:** depurador CUDA fornecido pelo fabricante com o toolkit;

- **Nsight:** depurador para windows (Visual Studio) e para linux (Eclipse);
- **Bibliotecas CUDA:** conjuntos de ferramentas que utilizam a arquitetura CUDA a fim de otimizar a realização de processos e cálculos científicos. Dentre as bibliotecas CUDA disponibilizadas por diversas organizações e grupos, destacam-se: NPP, CUBLAS, THRUST, MATLAB, MATHEMATICA, OpenCV, CUFFT, e outras;
- **Invólucros CUDA:** interfaces que permitem a utilização de CUDA com outras linguagens que não foram implementadas nativamente, como Java e Python e possivelmente outras, com a recente abertura do código-fonte pela Nvidia.;
- **Integração estática e dinâmica:** possibilidade de integração com qualquer linguagem via ligação de bibliotecas estáticas e dinâmicas.

Além dessas ferramentas, uma extensa documentação é encontrada no site da Nvidia e na internet, bem como fóruns e comunidades de apoio a desenvolvedores em diversos ramos. A transição de um paradigma sequencial para um paradigma paralelo de programação nem sempre é trivial, e requer alguma maturação e cuidados no momento de se paralelizar um algoritmo sequencial. Geralmente, tal processo envolve algumas questões, como:

- **Paralelização de laços:** laços que não tenham dependência dinâmica entre suas iterações são facilmente paralelizáveis;
- **Particionamento de processo:** o processo a ser paralelizado deve possuir algum paralelismo que possa ser explorado usando a GPU;
- **Sincronismo:** essencial e fonte de muitos erros incomuns durante o desenvolvimento;

- **Balanceamento de carga:** a carga de processamento deve, sempre que possível, ser equitativamente distribuída às unidades disponíveis de processamento;
- **Gargalos:** devem ser verificados se há gargalos de memória, influência das taxas de transferência de dados e demais situações que restrinjam o pleno desempenho da solução.

Uma vez elaborada a solução, esta pode ser executada de dois modos diferentes:

- Como ‘código-objeto’ a ser carregado na forma de driver, pela API de drivers CUDA;
- Como ‘aplicação’, na qual o código CUDA é ligado à aplicação diretamente, produzindo um arquivo executável.

Como exemplo de um possível fluxo de execução, podemos ter a seguinte sequencia: código *Host*³⁹ → cópia de dados para o dispositivo (GPU) → execução na GPU → cópia de dados para o *Host* → finaliza. Essa sequencia representa, de modo sucinto, uma sequencia básica de uma aplicação CUDA (Figura 2.18).

2.5.6 Capacidade de computação

A capacidade de computação, ou *compute capability*, é um parâmetro que qualifica a GPU e indica quais as funcionalidades disponíveis ao usuário. Em geral, baliza o desenvolvimento de uma determinada solução, permitindo definir quais delas serão necessárias ao seu correto funcionamento, além de qualificarem um equipamento perante um comprador, que pode saber, de antemão, se tal equipamento será adequado à sua solução. Um detalhamento profundo da capacidade de computação não é objetivo dessa seção, no entanto, a título de exemplo, detalhamos algumas das diferenças de funcionalidades entre as diferentes versões desse qualificador na Tabela 2.1.

³⁹*Host* representa a CPU, e *código Host* é o código que é executado na CPU.

Figura 2.18: Fluxo de processamento de um programa CUDA.

2.6 Processando imagens em C/C++: OpenCV

A OpenCV - *Open Source Computer Vision Library* (Biblioteca de Visão Computacional de Código Aberto) - é uma biblioteca que busca prover uma infraestrutura voltada à aplicação em visão computacional. Como esse ramo geralmente inclui diversos tipos de algoritmos diferentes, estes são agrupados em diferentes módulos, a fim de prover uma estrutura relativamente organizada, que pode ser utilizada para fins acadêmicos ou comerciais, sob a licença BSD. Tais módulos implementam, por exemplo, a calibração de câmera, extração de características de imagens e vídeos, manipulações em sequências de vídeo, uso de morfologia matemática e diversos outros algoritmos de processamento de imagens e classes para construção de interface gráfica, entre outras funcionalidades.

Essa biblioteca foi originalmente desenvolvida pela Intel em 2000 como uma biblioteca multiplataforma, e adota um foco bastante vol-

Funcionalidades*	Compute capability					
	1.0	1.1	1.2	1.3	2.x	3.0
Funções atômicas inteiras operando em palavras de 32 bits na memória global	Não	Sim				
atomicExch() operando em valores ponto flutuante de 32 bits na memória global	Não	Sim				
Funções atômicas inteiras operando em palavras de 32 bits na memória compartilhada	Não		Sim			
atomicExch() operando em valores ponto flutuante de 32 bits na memória compartilhada	Não		Sim			
Funções atômicas inteiras operando em palavras de 64 bits na memória global	Não		Sim			
Funções warp vote	Não		Sim			
Operações em ponto flutuante de precisão dupla	Não			Sim		

* As funcionalidades ausentes são suportadas em todas as versões (adaptado do site oficial da Nvidia).

Tabela 2.1: Tabela de funcionalidades em função da capacidade de computação (compute capability).

tado à eficiência e velocidade das implementações, característica que impõe ao usuário (programador) que leia atentamente a documentação, devido à grande quantidade de otimizações implementadas⁴⁰. Foi primariamente escrita em C, sendo adicionado suporte a C++ a partir da versão 2.0, e reforçado em versões posteriores. No entanto, apesar de seu código ser em C/C++, também oferece suporte a outras lingua-

⁴⁰Tais otimizações fazem com que nem sempre o código seja tão intuitivo (por exemplo, a implementação de um certo algoritmo pode não ser tão genérica quanto o é sua definição teórica, geralmente em virtude de sua complexidade), daí a necessidade da leitura atenta dos manuais da biblioteca, para saber as limitações e pressupostos da implementação, e em quais condições se obterá o resultado pretendido.

gens, como Java, Python e Visual Basic. A versão 1.0 foi lançada no final de 2006, a 2.0 em setembro de 2009, atualmente se encontra na versão 2.4 e seu desenvolvimento continua até os dias atuais.

A escolha dessa biblioteca para o nosso projeto se deu em virtude da curva de aprendizado e de seu suporte a CUDA, já incorporado em diversos de seus algoritmos. Além disso, o fato de ser multiplataforma (sendo até mesmo utilizada para construção de aplicativos para celulares) contribuiu nessa escolha. Essa biblioteca foi utilizada em conjunção com a libDicom, a fim de carregar as imagens de ressonância magnética nesse formato (DICOM⁴¹).

⁴¹DICOM é abreviação de *Digital Imaging and Communications in Medicine* (imagens e comunicação digital em Medicina), e consiste em um padrão que permite intercâmbio de imagens e informações associadas entre equipamentos de diagnóstico geradores de imagens, computadores e hospitais de modo unificado [39, 40].

Metodologia

3.1 Introdução

Neste capítulo são apresentadas as linhas gerais do trabalho, bem como um breve resumo dos estágios de desenvolvimento do trabalho, seguidos dos métodos utilizados. Em síntese, cinco estágios foram desenvolvidos: 1) estudos preliminares da q-Estatística e filtros q-Gaussianos 2) tradução de códigos e estudos aprofundados com filtros q-Gaussianos, 3) elaboração das q-Bases para interpolação *Spline* e 4) implementação, testes e validação da *q-Spline*. Resultados interessantes surgiram em cada uma das etapas, os quais são detalhados a seguir.

3.2 Um breve histórico

Efetuuou-se um estudo da q-estatística¹ voltado à elaboração de filtros de imagens. Essa etapa teve duas finalidades principais: a) familiarização com a q-Gaussiana e conceitos mais profundos da q-Estatística, favorecendo a posterior elaboração de uma q-base para uma *spline* utilizando esses conceitos e b) conhecer e avaliar o desempenho dos filtros gerados a partir das q-Gaussianas. Inicialmente, ensaios preliminares foram implementados utilizando-se o ImageJ, em

¹Também chamada “Estatística não-extensiva”, “Estatística não-aditiva” ou simplesmente “Estatística de Tsallis”.

ambiente Java, primeiramente sobre um conjunto de 160 imagens fractais aleatórias, colhidas na internet, e posteriormente sobre um conjunto de 160 imagens cerebrais de Ressonância Magnética Nuclear T2 (RMN), de uma base anônima utilizada pelo laboratório, obtida pelo HC, campus de Ribeirão Preto. Numa segunda etapa, foram realizados testes usando C++, OpenCV e LibDicom.

Foi utilizado o software *Mathematica* a fim de validar e visualizar as equações dos filtros. Os testes de aplicação dos filtros sobre o conjunto de imagens de teste demorava aproximadamente 16h, e consistiam na adição de ruído branco às imagens, com posterior filtragem e medição da **Relação sinal-ruído**² e da **Similaridade estrutural**³. Esta última foi utilizada nesse trabalho sob a forma de um plugin do ImageJ em nossa implementação Java. Na ocasião da qualificação, a banca sugeriu o uso da tecnologia CUDA, o que melhoraria significativamente a velocidade dos testes, permitindo uma análise mais abrangente do assunto. Seguindo essas sugestões, iniciou-se uma etapa de adaptação do código Java para C++, utilizando CUDA e OpenCV, entre outras bibliotecas.

Houve dúvida acerca de qual biblioteca de manipulação e processamento de imagens seria a melhor para nosso projeto. As principais opções eram o OpenCV e o Insight Toolkit (ITK), e, dado que a integração com CUDA era fundamental, optamos pelo OpenCV, por já incluir essa abordagem em boa parte da biblioteca. Esta se mostrou adequada para abrir, carregar e salvar imagens, bem como provia uma interface gráfica para o programa. Porém houve problemas no processamento de imagens. Descobrimos que essa biblioteca é mista, mantendo códigos em C e C++, o que muitas vezes forçou a troca de representação

²A Relação sinal-ruído, também chamada de **SNR** – *Signal Noise Ratio*, representa a relação da potência do sinal sobre a potência do ruído, ou seja, $SNR = \frac{P_{sinal}}{P_{ruído}}$.

³A similaridade estrutural é uma métrica de qualidade elaborada sob a premissa de que o sistema perceptual humano é altamente adaptado à extração de características estruturais de uma cena, proposta por Wang e Sheikh em 2004 [41], e implementado no Matlab e na forma de um plugin do ImageJ.

ao longo do código, pois nem todos os algoritmos C tinham um equivalente em C++. Muito do código C++ servia apenas para intermediar funções implementadas em C, que era mantido por conta de otimizações e de uma possível redução no tempo de execução.

Outro ponto importante é que alguns métodos não apresentaram o mesmo comportamento em todas as situações, não eram tão gerais quanto deveriam ser devido a otimizações e hipóteses simplificadoras adotadas pelos desenvolvedores⁴. Também foram encontradas limitações no sistema de representação matricial do OpenCV quando operando no ambiente CUDA, as quais foram resolvidas com o uso de ferramentas desenvolvidas durante o projeto. No tocante ao CUDA, seu uso era transparente para quaisquer funcionalidades da biblioteca, porém a incorporação de um código CUDA personalizado exigiu um certo esforço, por não existir nenhum manual ou documentação explícita acerca desse tópico.

Apesar dessas dificuldades, a escolha da biblioteca foi mantida, dado o avanço do código já escrito, e a mudança de biblioteca exigiria uma reestruturação quase completa de todo o trabalho de tradução. Este foi um período de muitos desafios, porém de grande aprendizado em diversos aspectos das tecnologias propostas e sua integração, já que o CUDA é relativamente novo e a documentação do OpenCV é muito sucinta⁵. Não obstante os desafios, os resultados foram interessantes.

Foram criadas classes template C++ para facilitar o uso da biblioteca OpenCV, bem como para integrar o uso desta com bibliotecas auxiliares, como é o caso da LibDicom, que permite ler o formato **DICOM** das imagens médicas.

⁴Como um exemplo, citamos o processo de convolução de uma matriz com uma imagem, que teve de ser reescrito pelo autor (que criou uma versão CUDA desse processo) e incorporado ao conjunto de ferramentas desenvolvidas no projeto. Nesse ínterim, as dificuldades encontradas indicam a necessidade da criação de tutoriais que facilitem esse tipo de implementação, e tais tutoriais passaram a se incluir nas possibilidades futuras de trabalhos escritos do autor.

⁵Muitas vezes foi necessário extrair informações diretamente do seu código fonte do OpenCV.

Essa etapa inicial nos proporcionou um excelente panorama do uso e aplicação dos filtros q-Gaussianos. A adaptação do código permitiu ampliar a gama de testes que podiam ser realizados, reduzindo expressivamente o tempo de execução, proporcionando uma melhoria média (ou *speedup*) em torno de 10x. Desse modo, novos dados foram incluídos, bem como novos testes em relação aos testes iniciais. Novas ideias e uma compreensão mais profunda da q-Gaussiana permitiram a elaboração de bases de interpolação que incorporam a principal característica da q-estatística, a não-aditividade. Essas bases foram formuladas e validadas no *Mathematica* e a representação da *spline* resultante foi testada em um sinal fractal sintético unidimensional, seguida da implementação dos testes bidimensionais em imagens biológicas reais.

3.3 Métodos utilizados

3.3.1 Elaboração e testes dos q-filtros

A primeira etapa do estudo da q-Gaussiana foi o estudo de sua equação unidimensional, cuja expressão é dada por⁶:

$$G_q(x) = \frac{1}{C_q \sqrt{2\sigma^2}} e_q^{\frac{-1}{2\sigma^2}(x-\mu_x)^2}, \quad (3.1)$$

com

$$C_q = \begin{cases} \frac{2}{\sqrt{1-q}} \int_0^{\pi/2} (\cos t)^{\frac{3-q}{1-q}} dt = \frac{2\sqrt{\pi}\Gamma(\frac{1}{1-q})}{(3-q)\sqrt{1-q}\Gamma(\frac{3-q}{2(1-q)})}, & -\infty < q < 1 \\ \sqrt{\pi} & q = 1 \\ \frac{2}{\sqrt{q-1}} \int_0^\infty (1+y^2)^{\frac{-1}{q-1}} dy = \frac{\sqrt{\pi}\Gamma(\frac{3-q}{2(q-1)})}{\sqrt{q-1}\Gamma(\frac{1}{q-1})} & 1 < q < 3 \end{cases}$$

na qual μ_x é a média em x . Essa equação produz uma família de q-Gaussianas cujo domínio se concentra em torno do ponto $x=0$ à medida

⁶Formulação elaborada com base em [42], com $\beta = 1/(2\sigma^2)$.

que o valor de q tende a $-\infty$. Por outro lado, sua forma tende à reta $y=0$ quando q se aproxima de 3. Para valores maiores ou iguais a 3, a q -Gaussiana não está definida.

Figura 3.1: Gráfico da q -Gaussiana unidimensional para $\sigma = 1$ e diversos valores de q . Essa função tende à função delta quando $q \rightarrow -\infty$.

Figura 3.2: Gráfico da q -Gaussiana unidimensional para $\sigma = 2$ e diversos valores de q (a marcação de funções é a mesma que na Figura 3.1).

A versão bidimensional foi obtida pelo produto simples de duas q -Gaussianas unidimensionais, com o uso da q -Álgebra. A expressão

obtida é dada por:

$$G_q(x, y) = \frac{e_q^{\frac{-(x-\mu_x)^2+(y-\mu_y)^2}{2\sigma^2}} e_q^{(1-q)(x-\mu_x)^2+(y-\mu_y)^2}}{2\sigma^2 C_q^2}, \quad (3.2)$$

a qual recai na Gaussiana 2D clássica para $q=1$. O gráfico dessa equação gera uma família de infinitos q -filtros, dos quais alguns são representados na Figura 3.3. É interessante notar os efeitos da super/sub aditividade nesta figura: a potencialização ou a atenuação dos valores da variável ‘ q -aleatória’⁷ bidimensional. Esses e outros representantes foram utilizados no presente estudo, bem como avaliados seus efeitos sobre imagens com diferentes dimensões fractais.

Foi desenvolvido um protótipo em Java, capaz de gerar quaisquer q -filtros possíveis, bem como criado um processo automatizado de filtragem e aplicação de métricas⁸, com posterior armazenamento dos resultados desse processo sobre conjuntos de imagens. Duas categorias de filtros podem ser criadas: filtros isotrópicos e filtros anisotrópicos⁹. Filtros isotrópicos possuem simetria radial, e são criados tomando-se a q -Gaussiana unidimensional e girando ao redor do eixo $z(x = y = 0)$. Filtros anisotrópicos são filtros que não apresentam simetria radial, e são criados pela amostragem de uma máscara sobre a equação 3.2, de modo a englobar um determinado *raio de corte*. Dentre os parâmetros utilizados para a criação dos filtros, podemos destacar¹⁰:

1. O **índice entrópico** q : o valor do índice entrópico para o filtro;
- 2.

⁷Esses graficos podem ser considerados como a distribuição de probabilidade de uma variável que é dada pela combinação de duas variáveis de distribuição q -Gaussiana (ou, como nos referimos no texto, uma variável ‘ q -aleatória’).

⁸A que foi denominado *Modo lote*, pois avaliava todo o lote de imagens de uma vez.

⁹Ressalvamos porém, que os filtros aqui desenvolvidos não são espacialmente dinâmicos, ou seja, não mudam suas características de acordo com as características de cada ponto da imagem. Em vez disso, eles são aplicados na imagem por meio de convolução. Tais filtros receberam essa denominação por terem sido gerados a partir de *funções q -Gaussianas anisotrópicas* (que atuam diferentemente em direções angulares diferentes, ou, em outras palavras, que não apresentam simetria radial).

¹⁰Obviamente, o protótipo lança mão de diversos outros parâmetros, como caminhos de diretórios de imagens e resultados, número de *threads* a serem executadas, etc.

(a) q-Gaussiana 2D com $q=-1$ e $\sigma=1$. (b) q-Gaussiana 2D com $q=-1$ e $\sigma=3$. (c) q-Gaussiana 2D com $q=2,5$ e $\sigma=1$. (d) q-Gaussiana 2D com $q=2,5$ e $\sigma=3$.**Figura 3.3:** Alguns representantes da família das *q*-Gaussianas.

Desvio padrão (σ): o valor para o desvio padrão desejado para o filtro;

3. **Média em x (μ_x):** valor da média em x desejada para o filtro;
4. **Média em y (μ_y):** valor da média em y desejada para o filtro;
5. **Raio de corte:** raio utilizado para a amostragem do filtro, considerando um círculo de corte na origem¹¹.
6. **Tamanho da janela:** tamanho da matriz que representa a má-

¹¹Observação: esse raio denota a menor distância do centro da janela do filtro até a aresta que representa o lado da janela do filtro.

cara do filtro (em *pixels*). Essa matriz é uma amostragem da função sobre o domínio quadrado mínimo que contém o raio de corte especificado;

Figura 3.4: Alguns parâmetros de criação dos filtros q-Gaussianos.

Alguns desses parâmetros são ilustrados na Figura 3.4. O sistema de construção de filtros foi definido de modo paramétrico, o que permitiu a criação de qualquer filtro q-Gaussiano, e se mostrou especialmente útil para experimentação. Dada a grande variedade de q-filtros possíveis, foram escolhidos alguns representantes mais significativos, buscando sobretudo diferenças qualitativas que pudessem representar razoavelmente a ampla gama de variações que podem ser obtidas variando-se q e σ . Os q-filtros se mostraram bastante sensíveis à mudança de parâmetros. A Figura 3.5 mostra alguns filtros com seus respectivos espectros de Fourier, criados a partir da Transformada Rápida de Fourier (FFT), a fim de investigar as faixas de frequências sobre as quais os filtros atuam nas imagens de origem. Nessa figura, vemos que alguns filtros q-Gaussianos induzem sincs no espectro de frequências (figuras ?? e ??), devido ao fato de também apresentarem descontinuidade espacial. Provavelmente, essa característica estará presente somente nos filtros que apresentarem tais descontinuidades, porém não houve uma investigação ampla nesse sentido.

Um dos desafios nesse protótipo era a impossibilidade de processar todo o lote de uma única vez, devido às limitações em relação à quantidade de memória utilizada pela Máquina Virtual Java, já que

Figura 3.5: Alguns exemplos de filtros da família das *q*-Gaussianas gerados pelo protótipo, com seus respectivos espectros de Fourier.

um teste para uma das imagens já ocupava uma quantidade grande de memória, isso sem considerar os vários ensaios para cada imagem, além de todos os ensaios necessários para cobrir o conjunto todo. A solução foi processar e salvar os dados paulatinamente, liberando a memória utilizada ao final do processamento de cada imagem. Como conjuntos de teste, foram utilizados dois conjuntos de imagens fractais, um formado por 162 imagens de fractais diferentes, escolhidos aleatoriamente na internet, e outro constituído por 160 cortes diferentes de tomografia cerebral do mesmo indivíduo (Ressonância Magnética T2). Para cada imagem desses conjuntos, foram realizados seis ensaios, consistindo nas seguintes etapas, detalhadas a seguir:

1. Classificar as imagens segundo sua dimensão fractal, dividindo-as em subconjuntos de mesma dimensão;
2. Para $\sigma \in \{1.0, 2.0, 3.0\}$ e $q \in C = \{-3.0, -2.9, -2.8, \dots, 2.8, 2.9, 2.99\}$:
 - (a) Criar uma cópia da imagem original, adicionada de ruído aleatório com distribuição gaussiana e $\sigma^2 = 1$;
 - (b) Avaliar a Relação Sinal-Ruído e Similaridade Estrutural da imagem ruidosa inicial;
 - (c) Aplicar o q-filtro para o valor correspondente de q e σ ;
 - (d) Avaliar a Relação Sinal-Ruído e Similaridade Estrutural da imagem resultante;
 - (e) Armazenar os valores calculados em disco.

Após o processamento de cada imagem, era calculada a média e desvio padrão para cada grandeza calculada nesses ensaios, a fim de compor um resultado final médio correspondente ao desempenho de cada filtro para a imagem em questão. Esse mesmo processo foi aplicado aos dois conjuntos.

Na versão em código nativo utilizando C++/OpenCV/CUDA, a rapidez do processo permitiu a ampliação da investigação, com a utilização de cinco tipos de ruídos: branco (aleatório), rosa ($1/f$), marrom ($1/f^2$), azul ($1/f^{-1}$) e violeta ($1/f^{-2}$). Numa primeira etapa utilizando o código traduzido foram estudadas 160 imagens de cortes tomográficos cerebrais T2¹². Na avaliação desse conjunto, foram aplicados os seguintes procedimentos:

1. Classificação das imagens segundo a dimensão fractal;
2. Para cada imagem, para cada valor de q e para cada tipo de ruído, realizar N ensaios de:

¹²Naquele momento buscava-se por outras imagens com as quais pudéssemos realizar mais testes e comparar os resultados.

- (a) Criação dos filtros;
- (b) Adição de ruído com amplitude de 10% do contraste máximo da imagem;
- (c) Cálculo da Relação Sinal-Ruído inicial (SNR_i), para obtenção de um valor de referência;
- (d) Aplicação do filtro sobre a imagem ruidosa;
- (e) Cálculo da Relação Sinal-Ruído final (SNR_f);
- (f) Cálculo do ganho dos filtros, dado por $Ganho = SNR_f / SNR_i$;
- (g) Cálculo do ganho efetivo da filtragem, dado por $G_E = (Ganho_{q-filtro} - ganho_{filtro\ clássico}) / (ganho_{filtro\ clássico})$.

3. Cálculo das estatísticas.

Com os testes e validação dos métodos nesse conjunto, foram adicionados mais dois conjuntos à base de teste, consistindo de 76 imagens de cortes de ressonância magnética T2 de tórax, e mais 88 imagens T2 de abdome. Para cada conjunto, os seguintes procedimentos foram aplicados:

1. Classificação das imagens do diretório de dados segundo sua dimensão fractal;
2. Para cada lote:
 - (a) Para cada imagem:
 - i. Realizar N ensaios de:
 - A. Criar uma cópia ruidosa da imagem (5 tipos de ruído, a 10% e 20%);
 - B. Medir a Relação Sinal Ruído (SNR) inicial;
 - C. Filtrar a imagem ruidosa;
 - D. Medir a Relação Sinal Ruído (SNR) final;
 - E. Cálculo do ganho dos filtros, dado por $Ganho = SNR_f / SNR_i$;

- F. Cálculo do ganho efetivo da filtragem, dado por $G_E = (\text{Ganho q-filtro} - \text{ganho filtro clássico}) / (\text{ganho filtro clássico})$.
- ii. Calcular a média e desvio padrão dos N ensaios anteriores para essa imagem;
- (b) Calcular a média e desvio padrão dos resultados das imagens do lote¹³;
3. Calcular a média e desvio padrão do resultado geral de todas as imagens de entrada;

Ao final de cada bateria de testes, os dados salvos no formato 'csv' eram importados no *Open Office* e no *Origin*, e então eram elaborados os gráficos para análise dos resultados.

3.3.2 Elaboração e teste das q-Splines

A etapa primordial para a elaboração das q-Splines consistiu em definir uma base que pudesse ser utilizada para constituir uma *spline*. Um dos requisitos era que a *spline* resultante fosse interpolante, bem como que incorporasse as características da q-estatística. Muito estudo matemático de base foi realizado, buscando uma solução matematicamente correta e útil, o que permitiu um aprofundamento dos conhecimentos matemáticos, tanto nos métodos de interpolação em si, quanto nos assuntos subjacentes, como o estudo de conjuntos algébricos, anéis e demais conceitos abstratos fundamentais que vieram à tona na busca da solução da questão proposta. A princípio, a busca de uma base fractal se mostrou infrutífera, já que um dos requisitos era que a *spline* fosse interpolante¹⁴, e a fractalidade em nada contribuía para esse aspecto. A sequência do estudo se deu no sentido de se elaborar 'q-versões' de bases polinomiais usuais, e os próximos candidatos considerados foram os polinômios de Bernstein, para os quais

¹³Lembrando que cada lote agrupou imagens com mesma dimensão fractal.

¹⁴A fim de possibilitar a representação de uma imagem.

tentou-se provar sua conexão com a q-Gaussiana. No entanto, após diversas tentativas, essa linha foi abandonada em favor de uma outra abordagem, a obtenção de uma ‘q-versão’ dos polinômios de Hermite.

Nesse trabalho lançamos o conceito de “q-Polinômio de Hermite”, definindo que o i -ésimo q-polinômio de Hermite de grau n deve ser expresso da seguinte forma:

$$H_{q,i}^n(x) = (-1)^n \cdot e_q^{x^2/2} \cdot \frac{d_q}{dx} e_q^{-x^2/2} = (-1)^n \cdot e_q^{x^2/2} \cdot D_q e_q^{-x^2/2} \quad (3.3)$$

na qual $D_q = d_q/dx$ é a q -derivada de uma função.

A ‘q-versão’ dos polinômios de Hermite proposta, ou como os denominamos, os “q-Polinômios de Hermite”, incorpora características importantes da q-estatística, em especial, a ‘não-aditividade’, o que lhe permite ser uma ferramenta bastante peculiar de interpolação. Seu comportamento parece modelar bem curvas de medidas (estatísticas) obtidas como a resultante de interações dinâmicas entre componentes internos ao sistema. A modelagem dessas interações pode ser controlada pelo valor do índice entrópico q , permitindo modelagem de interações super aditivas (interações construtivas), sub-aditivas (interações destrutivas), ou clássicas (aditividade normal).

Os testes na definição da q-Spline se iniciaram com ensaios analíticos no *Mathematica*. O processo de se definir uma nova spline consiste basicamente em:

1. Definir qual base de interpolação será utilizada;
2. Resolver o sistema dado pelo sistema de equações 2.33, obtido pela imposição de restrições à spline;
3. Opcionalmente, parametrizar a solução em termos de um vetor de restrições;
4. Substituir os resultados na equação da spline, à semelhança da expressão 2.35 ou 2.36 (em função da base proposta ou da base canônica dos polinômios).

Após essa formulação, estudos analíticos foram conduzidos a fim de se explorar as potencialidades da *spline* desenvolvida, utilizando curvas fractais sintéticas. Esses experimentos consistiram em procurar simular um processo semelhante ao da ampliação digital. Esse método foi utilizado tanto nos testes unidimensionais quanto bidimensionais da *q*-interpolação. No caso unidimensional, tomaram-se diversas curvas de Weierstrass, realizando-se amostragens e se construindo uma curva interpolante, como mostra a Figura 3.6. A curva azul representa uma *B-Spline* clássica de Hermite interpolante, com derivada zero em todos os pontos, e a curva violeta representa uma *q-Spline* interpolante, com $q = 2,6$. Os pontos em preto representam amostras tomadas sobre a curva fractal, e os pontos azuis assinalam uma reamostragem sobre a *q-Spline* que seria correspondente à imagem com suas dimensões hipoteticamente aumentadas. Analogamente, a resposta da interpolação clássica seria a intersecção entre a linha projetada pelo ponto amostrado sobre o eixo x e a curva *spline* clássica.

Figura 3.6: Amostragem sobre a curva de Weierstrass e a respectiva curva interpolante.

Dentre os parâmetros do experimento, um deles caracteriza a curva de Weierstrass, descrita por

$$W_a(x) = \sum_{k=0}^{\infty} \frac{\text{sen}(\pi k^a x)}{\pi k^a}, \quad (3.4)$$

que é o parâmetro α . Em nosso experimento, ele assumiu os valores $\{2, 3, 4, 5\}$. Para os pontos de interpolação (os P_I s), foram consideradas amostragens que continham de 4 a 11 pontos¹⁵, resultando em 8 amostragens para cada valor de α . Para os valores de q , foi considerado o intervalo $[-0,8, 2,9]$, com amostras espaçadas por $\Delta q = 0,01000001$ ¹⁶. Além disso, foi considerada apenas uma ampliação de fator 2, ou seja, uma vez definidos os pontos de amostragem, os pontos interpolados foram escolhidos de modo a simular uma ampliação de duas vezes, como na Figura 3.6. No caso desse experimento, essas amostras foram de 4 a 11 pontos sobre a curva de Weierstrass.

Considerando-se um valor de α e um determinado número de pontos de amostragem, foi calculado o **erro quadrático médio** (RMSE¹⁷) entre a curva hipotética interpolada e a curva verdadeira para todos os valores de q , resultando em uma curva de erro em função de q , a qual nos dá uma ideia do quão próximo os pontos interpolados se aproximam da curva original.

Resumidamente, os procedimentos implementados foram:

1. Para α em $\{2, 3, 4, 5\}$:

(a) Para $i = 4$ até 11:

i. Para $q = -0,8$ até 2,9:

- A. Criar a curva q -interpolante de W_α para i pontos para o valor atual de q ;
- B. Criar pontos hipotéticos de interpolação para $2i$ pontos;
- C. Calcular o RMSE dos pontos hipotéticos amostrados sobre a curva q -interpolada e a função de Weierstrass;

¹⁵Nesse valor estamos considerando os extremos também como pontos de interpolação, apesar de estarem sempre sobre o eixo x nesse caso.

¹⁶Esse pequeno residual (10^{-8}) foi utilizado em Δq como meio de se evitar pontos singulares nas equações.

¹⁷RMSE: *Root Mean Square Error* é a sigla em inglês para “erro quadrático médio”, e é dado por $RMSE = \frac{1}{n} \sum_{i=1}^n (a_i - b_i)^2$, considerando-se duas sequências a e b .

- D. Armazenar o valor do RMSE para esse valor de q ;
 - ii. Calcular o mínimo erro e encontrar o q correspondente;
- (b) Construção da curva de erro para esse valor de α .

Após esse estudo, efetuou-se o teste para a versão bidimensional da *q-Spline* utilizando-se de imagens biológicas para teste e validação da ferramenta. Nesta etapa, foram realizados os seguintes procedimentos:

1. Para cada imagem de um conjunto de n imagens:
 - (a) Para cada nível de zoom $z = \{2, 3, 4, 5\}$:
 - i. Reduzir o nível de zoom da imagem em z vezes (sub-amostrar a imagem);
 - ii. Criar uma interpolação de Hermite da imagem;
 - iii. Calcular o erro RMS da interpolação de Hermite;
 - iv. Criar uma interpolação de q -Hermite da imagem;
 - v. Calcular o erro RMS da interpolação de q -Hermite;
 - vi. Encontrar o q cujo erro RMS é mínimo para esse nível de zoom.

3.4 Os protótipos computacionais

O primeiro protótipo a ser elaborado foi o protótipo em Java, que utilizou a biblioteca do ImageJ, que é um pacote de processamento de imagens, a fim de carregar, exibir, processar e salvar as imagens. Foi criada uma interface Java para permitir a escolha pelo usuário dos parâmetros a serem utilizados para os testes de filtragem sobre o conjunto, a qual é exibida na Figura 3.7.

Apesar de utilizarmos a infra-estrutura do ImageJ neste protótipo, vários procedimentos tiveram de ser implementados, como o cálculo da dimensão fractal média a partir do método do *box count*, cálculo de

Figura 3.7: Tela do protótipo em Java para teste das q-Gaussianas.

SNR e convolução, função Gama e q-Gaussiana, gerador de filtros q-Gaussianos baseados em diversos parâmetros fornecidos pelo usuário, procedimentos de listagem de diretórios, carregamento e ordenação de imagens segundo a dimensão fractal calculada, entre outros procedimentos menores. Uma visão mais geral do sistema é apresentada na Figura 3.8.

Figura 3.8: Estrutura do protótipo em Java.

Essa versão armazenava os resultados no formato “csv”, que consiste em um arquivo texto cujos campos foram separados por tabulação. Tais arquivos foram posteriormente importados em planilhas do OpenOffice e Origin para produção dos gráficos apresentados ao longo do texto. Esse mesmo padrão foi mantido para o protótipo em C++/CUDA/OpenCV, desenvolvido posteriormente. Neste primeiro protótipo, foi utilizada a biblioteca do ImageJ para a geração de ruído gaussiano, que posteriormente era adicionado à imagem. Nesse método, ele classificava o nível de ruído pelo raio a ser utilizado para a máscara da gaussiana, resultando em ruído gaussiano com raio = 3 pixels, ou num ruído mais intenso, por exemplo, com raio = 60 pixels. Esses parâmetros foram utilizados, sendo que a primeira situação foi convencionada naquele momento como sendo “um baixo nível de ruído”, e a segunda como sendo “um elevado nível de ruído”.

O protótipo seguinte, desenvolvido em C++/CUDA/OpenCV a princípio somente constituiu-se de uma tradução da versão Java para essas linguagens, que evoluiu para um aprimoramento nos métodos de teste, com paralelização e otimização do código desenvolvido. Não havia um gerador de ruído, então criamos um módulo que era capaz de criar não só ruído aleatório como também ruídos do tipo $1/f^\beta$. Esse ruído foi criado pelo seguinte meio: foi gerado um ruído branco (aleatório), em seguida, ele foi transformado para o espaço de frequências, utilizando-se a transformada de Fourier, o resultado então foi multiplicado pela função $1/f^\beta$, e em seguida, o resultado foi novamente convertido para o domínio do espaço, por meio da transformada inversa de Fourier. Para cada ensaio era gerado um novo ruído diferente, sendo que foram realizados sete ensaios para cada imagem, e tomada a média como o resultado final.

Mais de 90% do código desse protótipo foi implementada utilizando-se programação genérica, com o uso de *templates* C++, que permitem a reutilização do algoritmo quando se modifica a estrutura de dados utilizada. Esse método foi utilizado inclusive para escrever o

código dos núcleos CUDA responsáveis pelo processamento realizado. Esse protótipo não possui interface gráfica, sendo executado somente pela linha de comando. Melhoramentos futuros incluem uma interface gráfica, possivelmente utilizando a biblioteca Qt.

Resultados e discussão

4.1 Introdução

Neste capítulo são apresentados os resultados obtidos durante o trabalho, bem como observações e apontamentos pertinentes, sempre que possível, respeitando a ordem cronológica dos fatos (embora em alguns períodos houve etapas desenvolvidas simultaneamente).

4.2 Experimentos com a q-Gaussiana e q-filtros

4.2.1 Primeira etapa: protótipo Java

Na primeira etapa do trabalho, o foco foi a investigação acerca da influência da dimensão fractal das imagens no processo de filtragem, a partir da análise de fractais sintéticos e imagens reais. O equipamento utilizado foi uma máquina *AMD Phenom II X4*, com 4 núcleos de processamento a 3.2 GHz cada, com 4GB de memória a 1.600 MHz. Foram utilizadas técnicas de processamento paralelo no protótipo Java, de modo a ocupar três dos quatro núcleos disponíveis no processador. Um processador foi poupado, a fim de possibilitar a execução das tarefas de sistema e evitar a pane da máquina. Num primeiro mo-

mento, foram adquiridas 100 imagens de fractais¹ aleatoriamente da internet para testes². Tais imagens consistiam em fractais gerados automaticamente por algoritmo e representações artísticas de fractais, coloridas e monocromáticas, de tamanhos de 500x500 até 1200x1200, aproximadamente. Em uma etapa posterior, foram utilizadas imagens cerebrais anônimas, adquiridas a partir do banco de dados mantido no laboratório de projetos anteriores. Tais imagens são de 16 bits de cinza, porém apresentando em média, 256 níveis de cinza, de tamanho 240x240. Os testes nas 100 imagens fractais coletadas da internet demorou 15:05:15 h, e os testes nas 160 imagens cerebrais durou 15:57:21 h.

Um dos fatos interessantes notado durante o desenvolvimento foi a diversidade de morfologias da família de filtros q-Gaussianos³. Pequenas variações nos parâmetros q e σ da q-Gaussiana bidimensional levou a uma ampla gama de filtros, teoricamente infinitos. Nessa primeira etapa, foi adotado um raio de corte circular em $\sigma = 3$ para os q-filtros⁴, com $q \in [-3; 3[$, e incrementos de 0,1. Os espectros de frequência foram calculados a fim de se verificarem quais frequências estariam sendo filtradas por cada um deles.

Observou-se que, para uma baixa taxa de ruído, a Relação Sinal-Ruído decai para ambos os conjuntos de imagem, sendo maior para valores inferiores a $q=1$ (gaussiana clássica). Esse decaimento se mos-

¹A característica fractal foi selecionada como ponto comum entre as imagens selecionadas pelo pressuposto inicial de que os elementos matemáticos oriundos da q-Estatística seriam capazes de modelar esse tipo de objeto de modo mais adequado que a Estatística Clássica.

²Posteriormente à implementação deste protótipo, encontramos um banco de dados de texturas chamado *Brodatz* (<http://www.ux.uis.no/tranden/brodatz.html>) que, apesar de não ter sido utilizado neste trabalho, é interessante para testes em processamento de imagens.

³Tal diversidade é brevemente ilustrada na Figura 3.3.

⁴Na ocasião não estava claro como eram definidos os filtros de imagem em relação ao raio de corte (ou seja, dada a gaussiana 2D, não se sabia qual deveria ser o raio de corte utilizado para se amostrar o filtro sobre a grade 2D a ser utilizada como máscara, como ilustrado na Figura 3.4). Posteriormente, nos testes binários (C++/CUDA), foi adotado o valor do raio de corte em $\sigma = 2$, valor encontrado por meio de comparação numérica de resultados do nosso algoritmo com os resultados para a mesma filtragem utilizando o mesmo filtro, implementado pelo OpenCV.

trou mais acentuado nas imagens de tomografia cerebral do que no conjunto geral de fractais. A Figura 4.1 apresenta um exemplo do ruído inserido nas imagens, e o respectivo resultado.

Figura 4.1: Exemplo de uma imagem do conjunto geral de fractais e dos níveis de ruído introduzidos nas imagens.

A seguir, é apresentada a lista de siglas utilizadas nos gráficos:

- SR - Relação Sinal-Ruído.
- SRI - Relação Sinal-Ruído Inicial: representa a Relação Sinal-Ruído da imagem original adicionada de um ruído gaussiano.
- SRF - Relação Sinal-Ruído do Filtro: representa a Relação Sinal-Ruído após a aplicação do filtro q-Gaussiano sobre a imagem ruidosa.
- SE - Similaridade Estrutural.
- SEI - Similaridade Estrutural Inicial: representa a similaridade estrutural da imagem original adicionada de um ruído gaussiano.
- SEF - Similaridade Estrutural do Filtro: representa a similaridade estrutural após a aplicação do filtro q-gaussiano sobre a imagem ruidosa.

- Ganho SR - Ganho do filtro em relação à Relação Sinal-Ruído (SRF/SRI).
- Ganho SE - Ganho do filtro em relação à Similaridade Estrutural (SEF/SEI).

Para cada valor de q (e portanto para cada um dos q -filtros gerado), obtém-se um valor diferente para os ganhos SR e SE, gerando-se um 'espectro de ganho' para os filtros gerados no intervalo $q \in [-3; 3[$, com incrementos de 0,1. Esses espectros geraram os gráficos das Figuras 4.2 a 4.6.

Figura 4.2: Ganho dos q -filtros em imagens de fractais gerais para ruído gaussiano com $r=3$.

Para ruídos mais intensos, observou-se um melhor ganho na Relação Sinal-Ruído para filtros com pico em torno de $q=1,0$ na média (Figuras 4.4 e 4.5), com patamares para valores de q contidos nos intervalos $[-3,0, -1,8]$ e $[-1,6, 0,4]$. Tais características podem variar de

Figura 4.3: Ganho dos q-filtros em imagens de cérebro humano para ruído gaussiano com $r=3$.

imagem para imagem, como por exemplo, para a imagem do conjunto de imagens cerebrais '**im00001.jpg**', na qual o pico se dá em $q=0,7$ (Figura 4.6).

Nas medidas de Similaridade Estrutural, o comportamento apresentado foi de progressivo crescimento, tanto para imagens com ruído gaussiano $r=3$ quanto quanto com ruído gaussiano $r=60$. Em imagens cerebrais com ruído gaussiano $r=3$, há um aumento desse indicador para valores de q no intervalo $[-0,4, 1,1]$. Em geral, os melhores resultados se encontram para q no intervalo de 0,9 a 2,99. A medida de Similaridade Estrutural se mostrou menos sensível à mudança nos valores do desvio padrão.

É importante observar que essa etapa também assinalou um primeiro contato com a elaboração de filtros para imagem, em especial utilizando a q-Gaussiana. O raio de corte utilizado para tais experimen-

Figura 4.4: Ganho dos q -filtros de desvio padrão = 1.0 em imagens de fractais gerais para ruído gaussiano com $r=60$.

tos foi de $r=3\sigma$, o que influenciou os resultados. Nas etapas posteriores, foram realizados ajustes nos parâmetros, os quais permitiram o surgimento de novas perspectivas sobre o assunto.

Figura 4.5: Ganho dos q -filtros de desvio padrão = 1.0 em imagens de cérebro humano para ruído gaussiano com $r=60$.

Figura 4.6: Ganho dos q -filtros de desvio padrão = 1.0 para a imagem 'im000000' de cérebro humano com alto nível de ruído.

(a) Imagem original.

(b) Imagem com ruído gaussiano $r=3$.

(c) Imagem com ruído gaussiano $r=60$.

Figura 4.7: Exemplo da imagem do conjunto geral de cérebro 'im000000.jpg' e dos níveis de ruído introduzidos na imagem.

4.2.2 Segunda etapa: testes preliminares em C++/CUDA/OpenCV

Esta segunda etapa foi de tradução, implementação nativa e ampliação dos testes, desta vez utilizando o protótipo escrito em C++/CUDA/OpenCV. A princípio foram testadas somente as imagens de Ressonância Magnética T2 (160 imagens de cérebro 480 × 480), por ser o único conjunto de dados disponível no momento. O foco foi a ampliação da investigação conduzida na etapa anterior, bem como a realização de uma **prova de conceito**⁵ acerca dos q-filtros, explorando suas potencialidades, esclarecendo pontos duvidosos e definindo suas limitações.

Nesta etapa, o experimento foi conduzido em uma máquina *Intel core i7*, com 8 processadores de 3.2 GHz e 6 GB de RAM, e uma placa *NVIDIA Tesla 2070*, sem o uso de paralelização em CPU, somente em GPU. O tempo de execução caiu de 17h para 1:10h quando realizada a paralelização do trecho de código mais usado em GPU. Um uso mais massivo de CUDA e otimizações no método, levaram o tempo a 25 min para os mesmos parâmetros iniciais (que inseria somente um tipo de ruído). Foram aplicadas as seguintes etapas no conjunto de dados:

1. Classificação das imagens segundo a dimensão fractal;
2. Para cada imagem, e para cada valor de q , realizar N ensaios de:
 - (a) Criação dos filtros;
 - (b) Adição de ruído com amplitude de 10% do contraste máximo da imagem;
 - (c) Cálculo da Relação Sinal-Ruído inicial (SNR_i), para obtenção de um valor de referência;

⁵Uma prova de conceito é um termo utilizado para denominar um modelo prático que possa provar a factibilidade ou viabilidade técnica de um conceito teórico. Em termos computacionais, representa uma implementação que, em geral, é resumida ou incompleta, de um método ou de uma ideia, realizada com o propósito de verificar que o conceito ou a teoria em questão é suscetível de ser explorado de uma maneira útil [43].

- (d) Aplicação do filtro sobre a imagem ruidosa;
- (e) Cálculo da Relação Sinal-Ruído final (SNR_f);
- (f) Cálculo do ganho dos filtros, dado por $Ganho = SNR_f / SNR_i$;
- (g) Cálculo do ganho efetivo da filtragem, dado por $G_E = (\text{Ganho } q\text{-filtro} - \text{ganho clássico}) / \text{ganho clássico}$.

3. Cálculo das estatísticas (média e desvio padrão).

Para cada imagem, o valor de q foi variado de -2 a 2,99, sendo incrementado de 0,1, e para cada um desses valores, foi gerado um filtro q -Gaussiano, o qual foi aplicado à imagem ruidosa. Foram utilizadas as seguintes categorias de ruídos: branco (aleatório), rosa ($1/f$), marrom ($1/f^2$), azul ($1/f^{-1}$) e violeta ($1/f^{-2}$). Para maior robustez, utilizou-se a média $N = 7$ ensaios. Para cada tipo de ruído, foram realizados $\{[(2,99 - (-2))/0,1] \times 7 = 350$ ensaios para cada imagem, totalizando $5 \times 160 \times 350 = 280.400$ ensaios para todos os 5 tipos de ruídos utilizados.

Cada ensaio consistiu aproximadamente de 2 convoluções, 2 cálculos de SNR e, à exceção do ruído branco, 2 transformadas de Fourier o que resulta, a grosso modo, $8 \times 280.400 = 2.240.000$ execuções dos métodos principais. Tal cifra somente pode ser alcançada num tempo satisfatório por meio do uso do CUDA. Isso sem contar a execução de métodos secundários (abrir imagens, calcular estatísticas, salvar dados, classificar imagens por dimensão fractal, etc.).

O raio de corte mínimo⁶ para a criação dos filtros foi de 2σ , de modo que para $q = 1$ se obtivesse o filtro gaussiano usual. O tamanho das máscaras de filtro foi de 5×5 pixels em todo o trabalho. As estatísticas foram calculadas em três níveis: considerando-se os resultados para uma imagem individual, os resultados para um lote de imagens com certa dimensão fractal, e os resultados gerais.

⁶Considerando a distância do centro do filtro à reta que constitui o lado da janela do filtro (que é quadrado).

Ruído	GE_{max} (%)	GE_{class} (%)	Aumento (%)	σ_{max}	q_{max}
marrom	2,078	0,484	329,524	$2,008 \cdot 10^{-4}$	0,00
rosa	3,352	2,080	61,106	$3,142 \cdot 10^{-4}$	0,00
violeta	6,688	4,958	34,902	$5,470 \cdot 10^{-4}$	0,00
azul	7,823	6,455	21,194	$6,631 \cdot 10^{-4}$	0,00
branco	11,919	11,135	7,034	$1,071 \cdot 10^{-3}$	0,30

* Para cada valor de q foram realizados $160 \times 7 = 1.120$ ensaios.

Tabela 4.1: Comparativo de ganhos para filtros q-Gaussianos anisotrópicos para imagens cerebrais, nível 10% de ruído.

Como convenção, adotamos o ganho efetivo do filtro como base de comparação com o filtro gaussiano clássico. Desse modo, representaremos como GE_{max} o máximo ganho efetivo obtido por um filtro q-Gaussiano com $q \neq 1$, e GE_{class} para o ganho efetivo clássico ($q = 1$). Ao máximo desvio padrão, denotaremos σ_{max} , e para o q de máximo ganho, q_{max} .

Como visualizado nas tabelas, foi possível encontrar valores de q para os quais o filtro q-Gaussiano apresentou significativo aumento no ganho efetivo em relação ao filtro clássico ($q = 1$), ilustrado na Tabela 4.1. Para os filtros isotrópicos o resultado foi muito semelhante, ao menos para os valores de q correspondentes aos picos de máximo ganho e para o filtro gaussiano clássico. Observou-se grande semelhança dos resultados entre a família de filtros isotrópicos e anisotrópicos. As Tabelas 4.1 e 4.2 apresentam um resumo geral dos resultados gerais obtidos pelo processo de filtragem anteriormente descrito, independentemente da dimensão fractal das imagens sobre as quais foram aplicados.

Podemos notar que o desempenho dos diferentes tipos de filtros é bastante similar, e que portanto, o efeito da anisotropia foi sutil no processo de filtragem.

Ruído	GE_{max} (%)	GE_{class} (%)	Aumento (%)	σ_{max}	q_{max}
marrom	2,078	0,484	329,524	$2,766.10^{-4}$	0,00
rosa	3,352	2,080	61,106	$3,142.10^{-4}$	0,00
violeta	6,688	4,958	34,902	$5,470.10^{-4}$	0,00
azul	7,823	6,455	21,194	$6,631.10^{-4}$	0,00
branco	11,919	11,135	7,032	$1,070.10^{-3}$	0,50

* Para cada valor de q foram realizados $160 \times 7 = 1.120$ ensaios.

Tabela 4.2: Comparativo de ganhos para filtros q -Gaussianos isotrópicos para imagens cerebrais, nível 10% de ruído.

Numericamente falando, houve forte influência da dimensão fractal no ganho dos filtros, e em todos os experimentos, observou-se predominância do ganho de imagens com certas dimensões fractais, fato para o qual ainda não foi possível elaborar uma explicação. Esse padrão pode ser visualizado na Figura 4.9. O maior aumento em comparação ao filtro clássico foi registrado com a adição de ruído marrom e dimensão fractal igual a 1,87, no qual o filtro q -Gaussiano obteve ganho efetivo 329,524% superior em $q = 0$, tanto para filtros isotrópicos quanto anisotrópicos.

Nesses exemplos, ficou bem claro que os pontos ótimos não são o filtro clássico ($q = 1$), mas estão deslocados, geralmente em torno do ponto $q = 0$, comportamento observado em diversos outros gráficos. As barras de erros dos gráficos representam o desvio padrão do lote em questão. Um ponto relevante é que, apesar da enorme quantidade de ensaios por ponto nesses gráficos, em muitos deles o desvio padrão foi relativamente pequeno, resultando em um resultado relativamente preciso.

Um outro destaque é que a morfologia da curva de ganho variou em relação a: 1) a cada tipo diferente de ruído adicionado e 2) com cada família de filtros observada, como pode ser evidenciado na

Figura 4.8: *Ganhos do filtro q -Gaussiano isotrópico com a adição de ruído marrom (10%).*

comparação das Figuras 4.9 e 4.8 (ruído marrom) e 4.10 e 4.11 (ruído branco). De modo geral, o espectro dos filtros isotrópicos foi mais suave que o dos filtros anisotrópicos.

Figura 4.9: *Ganhos do filtro q -Gaussiano anisotrópico com a adição de ruído marrom.*

Quando observado o comportamento do ganho efetivo em relação aos valores de q , nota-se que não houve mudanças na imagem

para $q < -1$. A partir desse valor, observa-se uma descontinuidade entre -1 e -0,9. A partir de -0,9, o espectro começa a se delinear.

Figura 4.10: *Ganhos do filtro q -Gaussiano anisotrópico com a adição de ruído branco.*

Figura 4.11: *Ganhos do filtro q -Gaussiano isotrópico com a adição de ruído branco.*

A filtragem gaussiana apresentou uma atenuação do ruído com o habitual efeito de borramento de bordas. Um exemplo pode ser visualizado na Figura 4.13(a), bem como a diferença. A q -filtragem pareceu preservar melhor as bordas. Nessa imagem, quanto mais próximo do branco, maior a diferença entre a q -filtragem e a filtragem gaussiana clássica, e quanto mais próximo do preto, menor é a diferença entre

Figura 4.12: Ganho total médio dos filtros q -Gaussianos anisotrópicos em relação aos diferentes tipos de ruído.

elas. O destaque fica por conta dos detalhes e contornos, ligeiramente mais vívidos no resultado da q -filtragem.

Numericamente, a relação sinal-ruído apresenta significativo ganho efetivo do filtro (muitas vezes com ganhos efetivos expressivamente maiores, como pode ser visto nas tabelas 4.1 e 4.2), embora à inspeção visual, essa diferença seja mais sutil, como mostrado na Figura 4.13. Essa Figura apresenta um comparativo entre as filtrações q -Gaussiana e clássica, bem como seus reflexos no histograma da imagem e o aspecto visual da diferença entre elas.

Um fato interessante foi que, para alguns valores de q_{max} , a atenuação efetiva do ruído ocorreu juntamente com uma melhor preservação de componentes de alta frequência (bordas) da imagem, o que poderia ser aproveitado em um eventual pós-processamento (em algoritmos como segmentação, detecção de bordas e outros).

(a) Tomografia cerebral T2 com ruído branco.

(b) Diferença entre os resultados das filtragens (d)-(c).

(c) Aspecto após filtragem q-Gaussiana anisotrópica ($q=0,5$).

(d) Aspecto após filtragem Gaussiana.

(e) Histograma de (c)

(f) Histograma de (d)

(g) Histograma de (b)

Figura 4.13: Comparativo entre a filtragem q-Gaussiana e a clássica.

4.2.3 Terceira etapa: ampliação dos testes em C++/CUDA/OpenCV

Na sequência dessa investigação, o protótipo foi consolidado e aperfeiçoado, sendo mantida a execução no mesmo equipamento, aumentada a resolução da variação de *q* (reduzindo os incrementos para 0,01) e adicionados mais dois níveis de ruídos (totalizando 10%, 20% e 30% do contraste máximo das imagens). Além disso, ampliamos o conjunto de testes, adicionando ao conjunto de imagens cerebrais mais dois conjuntos de imagens de ressonância magnética T2: 76 imagens 360 × 360 de tórax e 88 imagens 512 × 512 de abdome. Para cada conjunto, foram aplicados os seguintes procedimentos:

1. Para cada tipo de filtro (isotrópico/anisotrópico):
 - 1.1. Classificar as imagens por dimensão fractal (*box count*);
 - 1.2. Para cada imagem:
 - A. Criar um conjunto de *q*-filtros no intervalo pré-definido;
 - B. Para cada *q*-filtro:
 1. Para cada tipo especificado de ruído:
 - a. Para cada nível especificado de ruído:
 - I. Para $i=1$ até N (ensaios por imagem por filtro):
 1. Adicionar ruído do tipo e nível atuais do contraste máximo da imagem;
 2. Calcular a relação sinal ruído inicial (SNR_i), para um valor de referência;
 3. Aplicar o *q*-filtro sobre a imagem ruidosa e calcular a relação sinal-ruído final (SNR_f);
 4. Calcular o ganho como SNR_f/SNR_i ;
 - 1.3. Calcular as estatísticas sobre os resultados.

Como podemos notar, o procedimento acima é uma sequência de laços aninhados, o que resulta em uma quantidade enorme de cálculos.

Uma sólida infraestrutura de software foi desenvolvida para permitir a conexão e encadeamento dos procedimentos aplicados, além da generalização do algoritmo por meio de templates. Nesta etapa foram mantidos os parâmetros anteriores ($q \in [-2; 2, 99]$, incremento em $q = 0,01$, $N = 7$ tamanho 5×5 e ruídos (branco, rosa, marrom, azul e violeta). Nesta fase, os testes foram realizados em uma máquina Intel core i7 com 8 núcleo de 3.2 GHz cada e 6 GB de memória RAM, com uma NVIDIA Tesla 2070 como placa de vídeo.

Há uma conexão entre o valor de q e o efeito de ‘borramento’ da imagem, dada pela forma da equação da q -Gaussiana (Equação 3.1), o qual é maior quanto mais o seja a dispersão ou achatamento da q -Gaussiana. Desse modo, para valores de q superiores a 1, o efeito de borramento tende a se intensificar, e se reduz ao passo que o valor de q se distancia de 1 no sentido de $-\infty$. Quanto maior a distância em relação ao valor 1, maior tende a ser a diferença aparente na imagem filtrada. A título de comparação, uma diminuição do borramento é mostrada na Figura 4.13(b), e um aumento na Figura 4.14.

(a) Imagem torácica com ruído marrom após filtragem clássica. (b) Imagem torácica com ruído marrom após filtragem q -Gaussiana isotrópica (maior borramento, $q = 2, 99$). (c) Diferença entre as filtragens*. *Foi aplicada uma melhoria de 50 unidades de brilho para melhor definição.

Figura 4.14: Exemplo do efeito de borramento entre filtragem clássica e q -Gaussiana.

Para cada conjunto específico de parâmetros é obtido um espectro de ganho efetivo, bem como as demais estatísticas (σ_{max} , o máximo

desvio padrão - σ_{max} , ganho efetivo clássico - GE_{class} e o ganho efetivo máximo do espectro - GE_{max}). Os resultados evidenciaram mudanças de ganho em relação ao tipo de ruído e à sua intensidade, como é exemplificado nas Tabelas 4.3 a 4.6. Além disso, o aumento do ruído reduziu significativamente o ganho dos filtros e alterou a ordem de classificação dos processos de filtragem envolvendo os diversos tipos de ruídos (Tabelas 4.5 e 4.6).

Ruído	GE_{max} (%)	GE_{class} (%)	Diferença (%)	σ_{max}	q_{max}
marrom	0,716	-0,058	+1341,509	$1,24 \cdot 10^{-4}$	-0,9
rosa	1,955	1,546	+26,435	$2,57 \cdot 10^{-4}$	0,1
violeta	5,038	4,353	+15,745	$4,86 \cdot 10^{-4}$	0,1
azul	6,406	5,886	+8,843	$5,85 \cdot 10^{-4}$	0,2
branco	10,522	10,385	+1,32	$1,08 \cdot 10^{-3}$	0,7

* Para cada valor de q foram realizados $88 \times 7 = 616$ ensaios.

Tabela 4.3: Comparativo de ganhos para filtros q -Gaussianos anisotrópicos em imagens T2 de abdome, para diferentes tipos de ruídos a 10%.

As filtragens de imagens corrompidas com ruído branco apresentaram, em geral, as menores taxas de melhoria do ganho efetivo, quando comparados os rendimentos do filtro clássico e o do q -filtro de melhor ganho do espectro de ganho para $q \in [-2; 2, 99]$. Também foi observado que os filtros anisotrópicos produziram melhores resultados do que os filtros isotrópicos. De algum modo, a anisotropia parece contribuir para o aumento da relação sinal-ruído, mas não houve tempo hábil para uma investigação mais aprofundada desse aspecto. A diferença observada entre os diferentes tipos de ruído e seus níveis sugere que os filtros q -Gaussianos são sensíveis à mudanças nesses parâmetros, característica passível de ser explorada em diversos procedimentos de análise de imagens, como por exemplo, sua caracterização.

Ruído	GE_{max} (%)	GE_{class} (%)	Diferença (%)	σ_{max}	q_{max}
marrom	2,087	0,479	+335,357	$1,99.10^{-4}$	-0,8
rosa	3,356	2,094	+60,261	$3,16.10^{-4}$	0,0
violeta	6,615	4,890	+35,299	$5,42.10^{-4}$	0,0
azul	7,882	6,496	+21,329	$6,53.10^{-4}$	0,0
branco	12,007	11,169	+1,32	$1,05.10^{-3}$	0,6

* Para cada valor de q foram realizados $88 \times 7 = 616$ ensaios.

Tabela 4.4: Comparativo de ganhos para filtros q -Gaussianos isotrópicos em imagens T2 de abdome, para diferentes tipos de ruídos a 10%.

Também foi observado que a elevada taxa de ganhos efetivos observada numericamente pode impactar visualmente na imagem, como pode ser demonstrado nas tabelas e na Figura 4.13(b). Uma possibilidade interessante seria a inserção de novos tipos de ruído a fim de verificar o espectro de ganho em uma gama mais ampla de situações.

Ruído	GE_{max} (%)	GE_{class} (%)	Diferença (%)	σ_{max}	q_{max}
marrom	1,24	0,986	+25,733	$1,24.10^{-3}$	0,0
azul	6,320	6,201	+1,9	$4,54.10^{-2}$	0,6
violeta	19,126	18,963	+0,861	$5,51.10^{-2}$	0,6
rosa	0,0	-0,570	+0,754	0,0	-1,0
branco	25,111	25,088	+0,09	$1,9.10^{-2}$	2,99

* Para cada valor de q foram realizados $88 \times 7 = 616$ ensaios.

Tabela 4.5: Comparativo de ganhos para filtros q -Gaussianos anisotrópicos em imagens T2 de abdome, para diferentes tipos de ruídos a 20%.

Em relação à dimensão fractal, há uma clara dependência entre o ganho e o processo de filtragem. Em geral, alguns conjuntos apresentam alguma ordem entre o ganho e a dimensão, mantida em

determinados trechos do espectro, como, por exemplo, o progressivo aumento de ganho com o aumento da dimensão fractal. Mas esse comportamento não é uniforme em todos os conjuntos, e, mesmo em alguns espectros, as curvas de ganho podem se entrelaçar a partir de determinados valores de q , o que evidencia que um aprofundamento da investigação seria necessário para uma melhor compreensão desse comportamento.

Não obstante os questionamentos que podem surgir (pois é esta a magia de se estudar questões inéditas), os gráficos são confiáveis, por serem relativamente ‘densos’, isto é, apresentam uma grande quantidade de ensaios por ponto. No caso das imagens de cérebro, cada ponto do gráfico de espectro de ganho é a média entre 1.120 ensaios (160×7 , pois são 7 por imagem para cada q) para cada valor de q , para as imagens de tórax, cada ponto carrega a média entre 532 ensaios (76×7) e 616 ensaios por ponto, no caso dos testes com imagens abdominais. Do mesmo modo, a variação é apresentada nas barras de erro (sendo bastante pequena em alguns casos), o que aumenta a segurança da análise.

Ruído	GE_{max} (%)	GE_{class} (%)	Diferença (%)	σ_{max}	q_{max}
marrom	2,469	2,246	+9,921	$1,45 \cdot 10^{-2}$	0,3
violeta	5,290	5,071	+5,071	$6,76 \cdot 10^{-2}$	0,6
branco	25,372	25,141	+0,918	$1,93 \cdot 10^{-2}$	1,5
azul	8,939	8,861	++0,879	$2,18 \cdot 10^{-2}$	0,7
rosa	3,850	3,833	+0,435	$5,97 \cdot 10^{-3}$	0,9

* Para cada valor de q foram realizados $88 \times 7 = 616$ ensaios.

Tabela 4.6: Comparativo de ganhos para filtros q -Gaussianos isotrópicos em imagens T2 de abdome, para diferentes tipos de ruídos a 20%.

Sob outro prisma, a Figura 4.15 apresenta regiões ampliadas de uma imagem de RM cerebral T2 e as versões filtradas, acompanhadas

de seus respectivos histogramas.

(a) Imagem torácica com ruído branco a 10%.

(b) Imagem após filtragem isotrópica gaussiana clássica.

(c) Imagem após filtragem isotrópica q-Gaussiana ($q = 0,5$).

(d) Histograma de (a).

(e) Histograma de (b).

(f) Histograma de (c).

Figura 4.15: Exemplo do efeito do filtro q-Gaussiano em uma imagem de RM cerebral T2 e seus respectivos histogramas.

O efeito da filtragem reduz a gama de valores da região de fundo (maior pico do histograma) de modo mais acentuado na q-filtragem, tornando-a mais homogênea. Ao mesmo tempo, há a atenuação da discrepância em regiões de platô (parte à direita do histograma).

4.3 Testes com as q-Splines

4.3.1 Primeira etapa: Elaboração das q-bases e estudo analítico

Para o desenvolvimento dessa etapa, utilizou-se o *Mathematica* para validação da base desenvolvida ($H_q(x) = (-1)^n \cdot e_q^{x^2/2} \cdot D_q e_q^{-x^2/2}$, Equação 3.3). Essa base muda sua morfologia de acordo com o valor do índice entrópico q . A Equação 3.3⁷ nos permite obter uma expressão polinomial para os “q-Polinômios de Hermite” de grau 3, a qual é dada por:

$$H_{q,0}^3(t) = 4^{\frac{1}{-1+q}} [4 - (-1 + q)^2 t^4]^{\frac{1}{1-q}} \quad (4.1)$$

$$H_{q,1}^3(t) = 4^{\frac{1}{-1+q}} t [4 - (-1 + q)^2 t^4]^{\frac{1}{1-q}} \quad (4.2)$$

$$H_{q,2}^3(t) = -2^{-1+\frac{2}{-1+q}} [2 + (-3 + q)t^2] [4 - (-1 + q)^2 t^4]^{\frac{1}{1-q}} \quad (4.3)$$

$$H_{q,3}^3(t) = 2^{-1+\frac{2}{-1+q}} t [-8 + 2q + (-3 + q)(-2 + q)t^2] \cdot [4 - (-1 + q)^2 t^4]^{\frac{1}{1-q}} \quad (4.4)$$

A base $B_q^3(t) = \{H_0^3(t), H_1^3(t), H_2^3(t), H_3^3(t)\}$, doravante denominada “q-Base 1D de Hermite”, pode ter seus gráficos visualizados na Figura 4.16. Nessa base, temos um ponto de singularidade ($q=1$), e sua convergência numérica é para $q \in]-1; 3[$. Em seu ponto singular ($q=1$), é utilizada a expressão clássica da base de Hermite.

Utilizamos a base $B_q^3(t)$ para resolver o sistema descrito pelas equações 2.27 a 2.30 para $n = 3$, cuja solução é dada, em termos de

⁷Expressa como $H_q^n(x) = (-1)^n \cdot e_q^{x^2/2} \cdot D_q e_q^{-x^2/2}$.

Figura 4.16: Diversos representantes da "q-Base de Hermite".

x_0 , x_f , v_o e v_f ⁸ por:

$$\begin{aligned}
 c_0 = & \frac{16^{\frac{1}{1-q}}}{(-3+q)^2(1+q)} \left\{ -2^{\frac{1+q}{-1+q}} (3+2q-q^2)^{\frac{q}{-1+q}} v_f + (2^{\frac{q}{-1+q}}) \cdot \right. \\
 & (3+2q-q^2)^{\frac{1}{-1+q}} [5 \cdot 2^{\frac{q}{-1+q}} q + 2^{\frac{1}{-1+q}} (5-3q^2)] x_f + (4^{\frac{1}{-1+q}}) \cdot \\
 & \left. \{ 4^{\frac{q}{-1+q}} (-3+q^2) v_o + 4^{\frac{1}{-1+q}} [-8q v_o + (-3+q)(1+q)(3+q) x_o] \} \right\}
 \end{aligned} \tag{4.5}$$

⁸Os parâmetros x_0 , x_f , v_o e v_f representam os pontos de controle da spline de Hermite, e a solução ficou descrita em função deles, simplificando seu uso.

$$c_1 = \frac{1}{(-3+q)^2(-2+q)(1+q)} 2^{-\frac{7}{-1+q}} \{-16^{\frac{1}{-1+q}}(3+2q-q^2)^{\frac{q}{-1+q}} \cdot$$

$$(2^{3+\frac{1}{-1+q}} - 2^{\frac{q}{-1+q}}q)v_f - 212^{\frac{6+q}{-1+q}}v_o + (3+2q-q^2)^{\frac{1}{-1+q}} \cdot$$

$$[-32^{\frac{q}{-1+q}}q^2 + 2^{2+\frac{5}{-1+q}}(4+15q+q^3)]x_f + 128^{\frac{1}{-1+q}}[q(-7+q(25+(-9+q)q))v_o - 4(-4+q)(-3+q)(1+q)x_o]\}$$
(4.6)

$$c_2 = \frac{2}{(-3+q)^2(1+q)} \{4^{\frac{1}{1-q}}(3+2q-q^2)^{\frac{q}{-1+q}}v_f + 4^{\frac{1}{1-q}}[-(-3+q) \cdot$$

$$(1+q)]^{\frac{1}{-1+q}}[-5+q(-10+3q)]x_f - (-3+q)(1+q)(2v_o + 3x_o)\}$$
(4.7)

$$c_3 = \frac{2}{(-3+q)^2(-2+q)(1+q)} \{4^{\frac{1}{1-q}}(3+2q-q^2)^{\frac{q}{-1+q}}v_f + 2^{\frac{-3+q}{-1+q}} \cdot$$

$$(3+2q-q^2)^{\frac{1}{-1+q}}[-1+(-4+q)q]x_f - (-3+q)(1+q) \cdot$$

$$(v_o + 2x_o)\}$$
(4.8)

De posse da equação da base e dos coeficientes c_i , podemos definir a expressão da i -ésima subfunção da “ q -Spline” de grau 3 como:

$$s_{q,i}^3(t) = 2^{-1+\frac{2}{-1+q}}[4 - (-1+q)^2t^4]^{\frac{1}{1-q}} \{2c_0 + 2tc_1 - [2 + (-3+q)t^2]c_2 +$$

$$t[2(-4+q) + (-3+q)(-2+q)t^2]c_3\}$$
(4.9)

A equação 4.9 é passível de ser calculada, a despeito dos extensos coeficientes c_i . No entanto, iremos reescrevê-la de outro modo, como produto matricial, a fim de determinar uma matriz característica da q -Spline de Hermite, à semelhança do que foi feito na 2.35, na forma genérica da subfunção da spline como $s_i^n(t) = R_H \cdot M_H \cdot B(t)$, substituindo M_H por M_{qH} . Para isso, iremos escrever os coeficientes c_i em função das restrições P_0 , P_F , V_0 , e V_F , obtendo:

$$c'_0 = \frac{3+q}{3-q}P_0 + 2^{\frac{q-4}{q-1}}(1+q)(3+2q-q^2)^{\frac{1}{q-1}-2}(5 \cdot 2^{\frac{1}{q-1}} + 5 \cdot 2^{\frac{q}{q-1}}q - 3 \cdot 2^{\frac{1}{q-1}}q^2)P_F +$$

$$\frac{4}{q-3}V_0 + \frac{2^{\frac{q-3}{q-1}}(3+2q-q^2)^{\frac{1}{q-1}}}{q-3}V_F$$
(4.10)

$$c'_1 = \frac{-4(q-4)}{6-5q+q^2}P_0 + \frac{2^{\frac{-7}{q-1}}(1+q)(3+2q-q^2)^{\frac{1}{q-1}-2}}{q-2} \cdot$$

$$(2^{4+\frac{5}{q-1}} + 152^{2+\frac{5}{q-1}}q - 2^{\frac{5q}{q-1}}q^2 + 2^{2+\frac{5}{q-1}}q^3)P_F +$$

$$\frac{14-7q+q^2}{6-5q+q^2}V_0 + \frac{2^{\frac{q-3}{q-1}}(q-4)(3+2q-q^2)^{\frac{q}{q-1}}}{(q-3)^2(-2-q+q^2)}V_F \quad (4.11)$$

$$c'_2 = \frac{6}{q-3}P_0 + 2^{\frac{q-3}{q-1}}(1+q)(3+2q-q^2)^{\frac{1}{q-1}-2}(5+10q-3q^2)P_F +$$

$$\frac{4}{q-3}V_0 + \frac{2^{\frac{q-3}{q-1}}(3+2q-q^2)^{\frac{1}{q-1}}}{q-3}V_F \quad (4.12)$$

$$c'_3 = \frac{4}{6-5q+q^2}P_0 + \frac{2^{\frac{2(q-2)}{q-1}}(1+q)(3+2q-q^2)^{\frac{1}{q-1}-2}(1+4q-q^2)}{q-2}P_F +$$

$$\frac{2^{\frac{q-2}{q-1}}(-32^{\frac{1}{q-1}} - 2^{\frac{q}{q-1}}q + 2^{\frac{1}{q-1}}q^2)}{(q-3)^2(q-2)(1+q)}V_0 + \frac{2^{\frac{q-3}{q-1}}(3+2q-q^2)^{\frac{1}{q-1}}}{(q-3)(q-2)}V_F \quad (4.13)$$

Essas novas versões c'_i dos coeficientes nos permitem definir a matriz característica da abordagem de q -Hermite como:

$$M_{q_H} = \begin{pmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & m_{23} \\ m_{30} & m_{31} & m_{32} & m_{33} \end{pmatrix} \quad (4.14)$$

na qual $m_{00} = \frac{3+q}{q-3}$, $m_{01} = \frac{16-4q}{6-5q+q^2}$, $m_{02} = \frac{6}{q-3}$, $m_{03} = \frac{4}{6-5q+q^2}$,

$$m_{10} = \frac{2^{\frac{q-4}{q-1}}(1+q)(3+2q-q^2)^{\frac{1}{q-1}-2}(5 \cdot 2^{\frac{1}{q-1}} + 5 \cdot 2^{\frac{q}{q-1}}q - 3 \cdot 2^{\frac{1}{q-1}}q^2)}{2^{\frac{-7}{q-1}}(1+q)(3+2q-q^2)^{\frac{1}{q-1}-2}(2^{4+\frac{5}{q-1}} + 152^{2+\frac{5}{q-1}}q - 2^{\frac{5q}{q-1}}q^2 + 2^{2+\frac{5}{q-1}}q^3)}$$

$$m_{11} = \frac{q-2}{q-2},$$

$$m_{12} = \frac{2^{\frac{q-3}{q-1}}(1+q)(3+2q-q^2)^{\frac{1}{q-1}-2}(5+10q-3q^2)}{2^{\frac{2q-4}{q-1}}(1+q)(3+2q-q^2)^{\frac{1}{q-1}-2}(1+4q-q^2)},$$

$$m_{13} = \frac{4}{q-3},$$

$$m_{20} = \frac{14-7q+q^2}{6-5q+q^2},$$

$$m_{21} = \frac{4}{q-3},$$

$$m_{22} = \frac{2^{\frac{q-2}{q-1}}(-3 \cdot 2^{\frac{1}{q-1}} - 2^{\frac{q}{q-1}}q + 2^{\frac{1}{q-1}}q^2)}{(q-3)^2(q-2)(1+q)},$$

$$m_{23} = \frac{2^{\frac{q-3}{q-1}}(3+2q-q^2)^{\frac{1}{q-1}}}{q-3},$$

$$m_{30} = \frac{2^{\frac{q-3}{q-1}}(q-4)(3+2q-q^2)^{\frac{q}{q-1}}}{(q-3)^2(-2-q+q^2)},$$

$$m_{31} = \frac{2^{\frac{q-3}{q-1}}(3+2q-q^2)^{\frac{1}{q-1}}}{q-3},$$

$$m_{32} = \frac{2^{\frac{q-3}{q-1}}(3+2q-q^2)^{\frac{1}{q-1}}}{q-3} \text{ e } m_{33} = \frac{2^{\frac{q-3}{q-1}}(3+2q-q^2)^{\frac{1}{q-1}}}{(q-3)(q-2)}.$$

As equações evidenciam pontos de singularidade ($q=1, 2$), e a convergência numérica é para $q \in]-1; 3[$. Desse modo, as subfunções da q-Spline unidimensional podem também ser representadas como $s_{q,i}^n(t) = R_H \cdot M_{q_H} \cdot B_q^3(t)$. Obviamente, essa equação já nos define a “q-Spline”, já que esta assume o valor das suas subfunções para cada intervalo, ou seja, $S_{q,i}^3(t) = s_{q,i}^3(t)$. Essa notação é mais genérica do que a representação simplesmente em termos dos coeficientes c_i , e além de simplificar a representação da spline em termos de produtos matriciais, será útil na etapa bidimensional.

A seguir, apresentamos um exemplo da q-Spline na Figura 4.17, exemplificando seu comportamento para diferentes valores de q .

Para o exemplo da Figura 4.17, os pontos utilizados, com suas respectivas inclinações, foram $P_1 = (0, 8; 1)$, $V_1 = (2, 5; 1)$, $P_2 = (2, 3; 3, 6)$, $V_2 = (0, 5; -1)$, $P_3 = (4; 2)$, $V_3 = (0, 9; 7)$, $P_4 = (1; 2)$, $V_4 = (-1; -1)$, adotando P_i como a coordenada (x, y) do i -ésimo ponto, e V_i sua derivada em relação à curva *spline*. Os exemplos dessa figura nos dão uma mostra de como a influência da super aditividade na base pode afetar a curva final, característica ainda a ser explorada nas diversas áreas que utilizam a *spline*.

Figura 4.17: Exemplos de “q-Spline” com valores de q entre $-0, 98$ a $2, 9999$.

4.3.2 Segunda etapa: Testes unidimensionais

Após a formulação da q-Spline, passamos a uma fase de testes unidimensionais com funções fractais simuladas, especificamente da família de **funções de Weierstrass**⁹, $W_\alpha(x)$, visualizadas na Figura 4.18. À medida que aumenta-se o parâmetro α , a morfologia dessa curva se suaviza, não deixando porém de perder sua característica fractal. Essas curvas foram utilizadas para verificar o desempenho das q-Splines na representação de objetos com essa natureza.

Figura 4.18: Curvas de Weierstrass.

Testes iniciais buscando o entendimento das principais características da q-Spline nos renderam interessantes comportamentos, visualizados na Figura 4.19. No item (a), vemos as bases de interpolação expressando “super-aditividade”. Nessas condições, à medida que o valor interpolado se afasta dos extremos do sub-intervalo, a curva resultante exige uma super-soma, resultando em valores maiores que os obtidos por uma curva clássica. No item (b), o efeito é análogo, porém inverso, obtendo-se uma sub-soma, e um valor inferior ao da curva clássica.

⁹Tais funções podem ser representadas por $W_\alpha(x) = \sum_{k=0}^{\infty} \frac{\text{sen}(\pi k^\alpha x)}{\pi k^\alpha}$.

Figura 4.19: Exemplo das características não aditivas da q -Spline.

Os resultados dessa seção são, no mínimo, intrigantes. É possível que o bom desempenho da q -Spline esteja ligado a uma combinação de fatores que resultou, em todos os nossos experimentos, em melhorias em relação à interpolação clássica (entre eles as características únicas dessa spline, vislumbradas na Figura 4.19).

N_p	qE_{min}	RMSE em q_{min} (E_q)	RMSE em $q = 1$ (E_c)	Relação E_c/E_q	Diferença (%)
4	2,87	0,027591	0,0335893	1,2174	+21,74
5	-0,249	0,0235825	0,0251108	1,06481	+6,481
6	2,89	0,00511694	0,0175184	3,42361	+242,361
7	1,77	0,00996744	0,0100272	1,00599	+0,599
8	2,21	0,0108127	0,0109164	1,00959	+0,959
8	0,32	0,00548224	0,00550489	1,00413	+0,413
9	2,33	0,00936063	0,00946726	1,01139	+1,139
10	2,25	0,0065325	0,00670895	1,02701	+2,701

Tabela 4.7: Erro médio quadrático para diferentes amostragens sobre a curva $W_2(x)$ ($\alpha = 2$).

Uma pergunta suscitada durante nossas investigações foi acerca da conexão entre a característica fractal da curva (com sua auto-similaridade) e os resultados obtidos com o uso da q -Spline. Aparentemente, os elevados ganhos obtidos poderiam ser associados a dois fa-

tores primordiais: a existência de certa periodicidade inerente à curva de Weierstrass, e o ajuste do fator de aditividade por meio da variação do valor de q . Convencionou-se como $q_{E_{min}}$ o valor de q para o qual se obtém o mínimo erro quadrático. Nas tabelas denominamos por N_p o número de amostras tomada sobre a curva fractal para se realizar a interpolação.

Podemos notar, nas tabelas 4.7 a 4.10 que, dependendo da curva, certo número de amostras de interpolação maximiza o desempenho da q-Spline, observada em todos os ensaios. As tabelas estão acompanhada pelas respectivas visualizações de dados nas figuras de 4.20 a 4.23, e trazem o resultado numérico visualizado nos gráficos. Por exemplo, para uma amostra de tamanho 6 (terceira linha da tabela 4.7) temos a cor azul no gráfico da Figura 4.20.

Figura 4.20: Visualização do erro médio quadrático em função de q para $W_2(x)$. Os pontos destacados correspondem aos $q_{E_{min}}$ de cada curva.

Um fato curioso observado nessas tabelas é a consistente melhoria em relação à diminuição do erro proporcionada pela q-Spline. Não se sabe ainda se tal observação se manteria para qualquer curva fractal, ou mesmo para uma curva não-fractal, e certamente merece destaque a melhoria na redução do erro de cerca de 920% em um dos ensaios da Tabela 4.7. A princípio, tal resultado parece estar associado como maior poder de representação da curva proporcionado pela existência da super/sub aditividade. Tal característica, por si só, não é

Figura 4.21: Visualização do erro médio quadrático em função de q para $W_3(x)$. Os pontos destacados correspondem aos $q_{E_{min}}$ de cada curva.

capaz de modelar um fractal, porém pode ser um de seus constituintes, no sentido de que as interações aditivas intra-sistema estão entre os tipos mais simples de interações que, associadas a outras, poderia levar à criação de fractais. Somente pelo experimento, não foi possível definir se esse resultado se deve à forma específica da curva (que apresenta certa periodicidade), se está associado a uma característica de auto-similaridade da curva (o que poderia levar ao mesmo fenômeno de periodicidade), ou ambos.

Esses testes nos forneceram indícios favoráveis ao uso das q -

N_p	$q_{E_{min}}$	RMSE em q_{min} (E_q)	RMSE em $q = 1$ (E_c)	Relação E_c/E_q	Diferença (%)
4	2,43	0,0119026	0,0128284	1,07778	+7,778
5	2,57	0,00300427	0,00539502	1,79578	+79,578
6	2,6	0,0103947	0,0110156	1,05974	+5,974
7	2,4	0,0070602	0,00736263	1,04284	+4,284
8	2,41	0,00663933	0,00695778	1,04797	+4,797
9	2,16	0,00644355	0,00657967	1,02112	+2,112
10	2,21	0,00370079	0,00397873	1,0751	+7,51
11	2,09	0,00348461	0,00363973	1,04452	+4,452

Tabela 4.8: Erro médio quadrático para diferentes amostragens sobre a curva $W_3(x)$ ($\alpha = 3$).

N_P	$q_{E_{min}}$	RMSE em q_{min} (E_q)	RMSE em $q = 1$ (E_c)	Relação E_c/E_q	Diferença (%)
4	2,77	0,00221622	0,0100603	4,53936	+353,936
5	2,56	0,00102723	0,00455476	4,43404	+343,404
6	2,4	0,00227773	0,00359999	1,58052	+58,052
7	2,16	0,00442828	0,00475946	1,07479	+7,479
8	1,84	0,00476734	0,00489428	1,02663	+2,663
9	0,64	0,0000225684	0,000230327	10,2057	+920,57
10	1,16	0,00438781	0,00448917	1,0231	+2,31
11	1,22	0,00416121	0,00423011	1,01656	+1,656

Tabela 4.9: Erro médio quadrático para diferentes amostragens sobre a curva $W_4(x)$ ($\alpha = 4$).

Figura 4.22: Visualização do erro médio quadrático em função de q para $W_4(x)$. Os pontos destacados correspondem aos $q_{E_{min}}$ de cada curva.

Splines, demonstrando que elas são generalizações da B-Spline tradicional, com ampliado poder de representação e podem apresentar maior precisão em diversas situações como as avaliadas neste trabalho.

Figura 4.23: Visualização do erro médio quadrático em função de q para $W_5(x)$. Os pontos destacados correspondem aos $q_{E_{min}}$ de cada curva.

N_p	$q_{E_{min}}$	RMSE em q_{min} (E_q)	RMSE em $q = 1$ (E_c)	Relação E_c/E_q	Diferença (%)
4	2,65	0,00539133	0,00852444	1,58114	+58,114
5	2,56	0,00102418	0,00449543	4,38931	+338,931
6	2,31	0,00289251	0,00380107	1,31411	+31,411
7	2,35	0,00188766	0,00282908	1,49872	+49,872
8	2,21	0,000505213	0,00165875	3,28326	+228,326
9	2,09	0,000285847	0,00129168	4,51876	+351,876
10	1,99	0,000328426	0,00111037	3,38088	+238,088
11	1,88	0,000727188	0,00116641	1,60401	+60,401

Tabela 4.10: Erro médio quadrático para diferentes amostragens sobre a curva $W_5(x)$ ($\alpha = 5$).

4.3.3 Terceira etapa: Formulação bidimensional

4.3.3.1 Formulação e testes analíticos

Para a formulação bidimensional, partimos da q-definição unidimensional ($s_i^n(t) = R_H \cdot M_{q_H} \cdot B_q^3(t)$) e, utilizando os desenvolvimentos da seção 2.4.4, especialmente o resultado da equação 2.43 ($\sigma^n(t, u) = B_{can}^T(t) \cdot M_H^T \cdot R_{2D} \cdot M_H \cdot B_{can}(u)$), expressamos a versão bidimensional da q-Spline¹⁰ como

$$\sigma_q^n(t, u) = B_q^3(t)^T \cdot M_{q_H}^T \cdot R_{2D} \cdot M_{q_H} \cdot B_q^3(u) \quad (4.15)$$

Há pontos de singularidade ($q=1, 2$), e a convergência numérica é para $q \in [-0, 5; 3[$. Um dos testes dessa formulação, utilizou

$$R_{2D} = \begin{pmatrix} 0 & 0.5 & 1 & -0.2 \\ 1 & 0.4 & 1 & 1 \\ 2 & -1 & -9 & 7 \\ -1 & -1 & 1 & 0.8 \end{pmatrix} \quad (4.16)$$

resultando na superfície da Figura 4.24.

Figura 4.24: Visualização da superfície q-Spline para $q=2.9$.

¹⁰A menos do uso da base canônica dos polinômios, a Equação 2.43 é equivalente à que está sendo introduzida para a “q-Spline”.

4.3.3.2 Testes computacionais

Foram realizados testes com 4 níveis de imagens reamostradas com resoluções inferiores, respectivamente 2, 3, 4 e 5 vezes menores na resolução espacial¹¹, que denominamos resumidamente neste trabalho como “níveis de zoom”. A redução do nível da amostragem ocasionou maior erro médio quadrático (RMSE), como esperado, e portanto o nível 5 apresentou o maior erro.

Foram tomadas 2 imagens de tórax, 2 de abdome e 2 de cérebro. Não houve tempo hábil para finalizar uma implementação em GPU, portanto, somente um conjunto reduzido de imagens foi utilizado. O algoritmo foi validado por meio de um ensaio comparando seus resultados (implementados em C++) aos resultados obtidos no *Mathematica*, com aproximadamente 200 pontos de comparação. Os erros, ou a diferença dos resultados, foi da ordem de 10^{-15} . Investigada a razão desse valor, comprovou-se ser este o “zero da máquina” (ou precisão numérica máxima efetiva).

As imagens I00005 e I00006 são de tórax (T2), as imagens I00013 e I00014 são de abdome (T2) e as imagens I00032 e I00033 são cerebrais (T2), e os arquivos produzidos de dados possuem o nome das imagens com extensão “.xls”. Quanto à dimensão fractal, a imagem I00005 tem dimensão 1,925, a imagem I00006 tem dimensão 1,923, a imagem I00013 apresenta dimensão 1,920, a imagem I00014 apresenta dimensão 1,927, a imagem I00032 tem dimensão 1,942, e a imagem I00033 tem dimensão 1,942.

O algoritmo gravou todas as saídas numéricas como texto formatado, no formato “.csv”, a fim de que se pudesse trabalhar os dados posteriormente, realizar análises e construir os gráficos. As imagens produzidas durante o processamento foram salvas em formato “jpeg” para comparação e verificação posteriores.

Na Figura 4.25 pode se notar o aumento da granularidade da ima-

¹¹Por exemplo, se o tamanho da imagem original é 500x500, e ela é reamostrada no nível ‘2’, o tamanho da imagem reamostrada será 250x250, e assim por diante.

Figura 4.25: Imagem cerebral (I00032) com diferentes níveis de interpolação, para $q=1.38$.

gem à medida que o fator de subamostragem aumenta. O fator 2 indica que a imagem foi reamostrada sobre a imagem original para assumir um tamanho duas vezes menor antes de se aplicar a interpolação, o fator 3, três vezes, e assim por diante. Esse procedimento foi realizado a fim de se simular uma ampliação, e permite a comparação do resultado com a imagem original, a fim de se calcular o erro adicionado pela interpolação. Nas Figuras 4.26, 4.28 e 4.30, são apresentados os gráficos de erro, bem como algumas estatísticas nas respectivas tabelas e exemplos do resultado visual das interpolações.

Ressaltamos que, embora possamos ter encontrado um valor mínimo para o erro em todos os testes, este difere muito pouco de outros

(a) Imagem T2 de tórax I00005.

(b) Imagem T2 de tórax I00006.

Figura 4.26: Visualização do erro médio quadrático em função de q para imagens de tórax. Os pontos destacados correspondem aos $q_{E_{min}}$ de cada curva.

"Zoom"	$q_{E_{min}}$	RMSE em q_{min} (E_q)	RMSE em $q = 1$ (E_c)	Diferença (%)
2	1,54	44,30705204	44,44358153	+0,3081
3	1,59	72,21465738	72,31096627	+0,1333
4	1,34	92,48770694	92,51235782	+0,0266
5	0,87	109,2878431	109,3008338	+0,0118

Tabela 4.11: Erro médio quadrático para diferentes amostragens para a imagem I00005 (tórax).

(a) Imagem interpolada classicamente ($q=1$).(b) Imagem q -interpolada ($q=1,54$).

(c) Diferença (b-a).

Figura 4.27: Diferença entre os diferentes tipos de interpolação (imagem I00005, tórax).

valores dentro de uma região, ou platô de erro mínimo. Esse fenômeno foi observado em todas as imagens interpoladas. Nota-se também que o erro apresentado das q -interpolações para as imagens de tórax foi significativamente maior que o erro observado nas demais imagens. Houve um ganho melhor que o observado nas outras imagens, e, para níveis baixos de subamostragem, os ganhos foram insignificantes quando comparados ao ganho do filtro clássico. É possível que o aumento no erro das imagens de abdome esteja vinculado a características intrínsecas da imagem (como dimensão fractal e distribuição de pixels), porém nada se pode afirmar no momento. Um outro ponto relevante é que, mesmo em imagens do mesmo tipo, esse erro sofre variações, bem como os valores de índice entrópico para os quais a interpolação é ótima (ou apresenta menor erro quadrático médio).

(a) Imagem T2 de abdome I00013.

(b) Imagem T2 de abdome I00014.

Figura 4.28: Visualização do erro médio quadrático em função de q para imagens de abdome. Os pontos destacados correspondem aos $q_{E_{min}}$ de cada curva.

Nos casos bidimensionais, os ganhos obtidos pela simples substituição da B -Spline clássica pela q -Spline apresentaram-se mais discretos que no caso unidimensional. A abordagem deste trabalho não fez uso de técnicas mais avançadas de vetorização de imagens, pois isso poderia criar um viés nos resultados.

O intuito nesse trabalho foi uma comparação básica entre diferentes abordagens spline, a fim de obter as características mais gerais

"Zoom"	$q_{E_{min}}$	RMSE em q_{min} (E_q)	RMSE em $q = 1$ (E_c)	Diferença (%)
2	1,20	10,48111770	10,49889022	+0,1695
3	1,35	19,87666448	19,90386680	+0,1368
4	1,52	27,44765414	27,48696382	+0,1432
5	1,39	35,11665570	35,13550711	+0,0536

Tabela 4.12: Erro médio quadrático para diferentes amostragens sobre a curva para a imagem I00013 (abdome).

(a) Imagem interpolada classicamente ($q=1$). (b) Imagem q-interpolada ($q=1,54$). (c) Diferença (b-a).

Figura 4.29: Diferença entre os diferentes tipos de interpolação (imagem I00005, abdome).

possíveis da diferença entre as abordagens. Os resultados sugerem que a q-Spline parece preservar melhor as bordas que a abordagem clássica (quando $q=1$), além de, em todos os casos, produzir interpolações sensivelmente melhores para interpolações não clássicas.

Os resultados bidimensionais apresentados até o momento, mostraram-se bastante similares visualmente à interpolação clássica, mas gostaríamos de chamar a atenção para uma peculiaridade da q-interpolação, destacada na Figura 4.32. A q-interpolação produz uma ampla gama de resultados, por tratar-se da manipulação de infinitas representantes de bases, cada qual definida por um valor diferente de q . Quando esse valor é distante do valor clássico (1), a superfície

(a) Imagem T2 de cérebro I00032.

(b) Imagem T2 de cérebro I00033.

Figura 4.30: Visualização do erro médio quadrático em função de q para imagens de cérebro. Os pontos destacados correspondem aos $q_{E_{min}}$ de cada curva.

"Zoom"	$q_{E_{min}}$	RMSE em q_{min} (E_q)	RMSE em $q = 1$ (E_c)	Diferença (%)
2	1,38	18,04259706	18,09497477	+0,2903
3	1,46	31,78654431	31,84116485	+0,1718
4	1,38	42,29663361	42,32276564	+0,0617
5	1,63	51,17891966	51,25894146	+0,1563

Tabela 4.13: Erro médio quadrático para diferentes amostragens sobre a curva para a imagem I00032 (cérebro).

gerada não é exatamente "suave" globalmente, fato que, surpreendentemente, pode melhorar algumas características da imagem (como melhor definição nas 'bordas' das formas representadas pela imagem) mas também pode gerar alguns efeitos colaterais, sobretudo para valores muito distantes¹² de 1, como visualizado na Figura 4.32. Quanto à possível conexão com a dimensão fractal, também nada foi evidenciado, nenhuma característica marcante que distinguísse as imagens de dimensões diferentes (ressalvando que o número de imagens utilizado não permite nenhuma conclusão segura a respeito).

¹²A q -Spline bidimensional (Equação 4.15) converge numericamente para $q \in [-0.5, 3[$, exceto $q = \{1; 2\}$ (pontos singulares).

(a) Imagem interpolada classicamente ($q=1$). (b) Imagem q-interpolada ($q=1,38$). (c) Diferença (b-a).

Figura 4.31: Diferença entre os diferentes tipos de interpolação (imagem I00032, cérebro).

(a) Detalhe da q-Interpolação com $q=-0,48$.

(b) Detalhe da q-Interpolação com $q=2,90$.

Figura 4.32: Visualização da influência da super/sub-aditividade na interpolação.

4.4 Conclusões

O plano de pesquisa adotado durante o trabalho possibilitou a análise progressiva das nuances da q -estatística, bem como a exploração do índice entrópico e seu papel nos diversos cenários propostos. A princípio, como muitas bibliografias associam esta teoria a uma nova gama de modelos, em muitos casos lidando com fractais, houve um forte interesse em investigar a relação dessa teoria com a dimensão fractal das imagens estudadas.

Os resultados com os q -filtros, e a própria análise da q -Gaussiana se mostraram promissores, e ofereceram pistas sobre o melhor caminho a seguir. Aos poucos, tornou-se claro que sua diferença primordial em relação à abordagem clássica ficava por conta da mudança na aditividade, controlada pelo índice entrópico q , o qual facilita o controle morfológico das funções nas quais ele participa. Comparar as abordagens clássicas isoladamente com uma abordagem no contexto da q -Estatística chega a ser uma comparação desleal, pois esta engloba de fato a abordagem clássica. Enquanto abordagens clássicas conseguem representar modelos para apenas *um valor* de q , *infinitos outros* podem ser obtidos com a variação deste parâmetro e, muito frequentemente, um desses modelos se ajusta melhor ao problema. Ou seja, a abordagem clássica é um membro de uma classe maior de funções, as quais envolvem o índice entrópico em sua definição.

Esse fenômeno foi observado na análise das curvas de ganho dos q -filtros, e muitos resultados de melhor ganho foram encontrados para ($q \neq 1$) numericamente, fato fortemente reforçado pela posterior ampliação dos testes. Vários representantes dos q -filtros preservaram em maior grau bordas e contraste, apesar de também atenuar o ruído em relação à imagem ruidosa original. Obviamente, houve variações, bem como diversos q -filtros apresentaram efeitos de borramento superior ao do filtro gaussiano (e portanto, desempenho inferior), porém, mesmo diante da massiva quantidade de q -filtros disponíveis (cada qual para

um q diferente), quase sempre foi possível encontrar q -filtros superiores ao gaussiano clássico.

Na segunda etapa, após a migração do código Java para código binário com aceleração CUDA, e a ampliação massiva dos testes, quase todas as possibilidades de uso vislumbradas puderam ser testadas numericamente. Essa ampliação e a utilização de mais quatro tipos de ruídos, mantendo-se ainda a realização de 7 ensaios por ponto, ampliou o entendimento dos filtros e tornou os resultados mais confiáveis. Inclui-se nessa ampliação o teste tanto de q -filtros isotrópicos quanto q -filtros anisotrópicos, ficando evidente que, dependendo do tipo de ruído e da anisotropia do filtro, os ganhos efetivos poderiam variar de 1,32% a 1.341,51%. A classificação, em ordem decrescente de ganhos foi, de modo geral, para imagens com ruído marrom, rosa, violeta, azul e branco.

Outro ponto que ficou bastante claro foi a dependência entre o ganho resultante da filtragem e a dimensão fractal das imagens filtradas, para todos os casos, inclusive o clássico. Para alguns tipos de imagens, quanto maior a dimensão fractal, maior era o ganho resultante da filtragem, fato muitas vezes observado em todo o espectro de ganho (para todos os valores de q utilizados). Não foi possível, no entanto, obter uma explicação para esse comportamento.

Paralelamente às implementações e trabalhos no filtro, pesquisas matemáticas foram realizadas no intuito de utilizar a q -gaussiana na elaboração de uma base viável para interpolação. O desafio foi idealizar uma formulação interpolante que preservasse as características observadas na q -Estatística, nesse caso, a super/sub aditividade, a qual constitui a diferença fundamental entre a q -estatística e a estatística clássica. Uma vez definida, a q -base de Hermite se mostrou bastante flexível, variando tanto com o índice entrópico q , que chega a tornar-se irreconhecível para valores extremos do parâmetro.

Os testes unidimensionais evidenciaram um interessante potencial de uso, com significativas reduções no erro médio quadrático em

relação à abordagem clássica, chegando a 1.300% de redução em alguns casos, instigando maiores investigações.

Graficamente, observou-se que as bases spline cúbicas tinham seus graus ampliados, de modo que o grau de um componente da base de grau 3 da spline clássica poderia corresponder a uma q -base de grau 5, possibilitando à spline uma maior capacidade de “oscilar”, ou inverter seu sentido em torno do eixo horizontal.

O desenvolvimento analítico bidimensional exemplifica a extração do método para dimensões superiores, preparando o caminho para interpolações n -dimensionais por meio de produtos tensoriais. Ambas as representações materializam o foco principal do trabalho, que foi o desenvolvimento das bases criadas.

O destaque na versão bidimensional da interpolação recai sobre as bordas da imagem, a despeito do resultado discreto. E ainda assim, o resultado é visível na imagem da diferença, levantando a possibilidade de que haja algum efeito de compensação no erro médio. Na maioria dos testes, a q -interpolação parece realçar o contraste em níveis variáveis, dependendo do tipo de sinal e demais parâmetros, sobretudo o valor de q . Nesse contexto, a análise do histograma revela uma maior distinção entre as diferentes regiões da imagem, o que denota esse efeito. Ainda assim, uma desvantagem da abordagem adotada é que ela não explora a capacidade da spline de resumir dados através da vetorização da informação, o que sem dúvida alguma alteraria o resultado.

Quanto à dimensão fractal, na interpolação, apesar dos resultados interessantes, não ficou evidenciada uma conexão entre a dimensão fractal dos elementos analisados e as q -Splines, embora mais estudos devam ser conduzidos, a fim de obter uma eventual relação entre ambas.

Os problemas da formulação consistem nos pontos singulares, bem como na grande sensibilidade das soluções para valores de q próximos aos valores extremos do intervalo de validade, regiões nas quais

as curvas divergem muito rapidamente.

Em síntese, as principais contribuições desse trabalho foram:

- Elaboração das splines 1D e 2D com base na q -estatística (q -Splines);
- Testes e validação analíticos das splines criadas;
- Testes numéricos em curvas sintéticas e imagens médicas das q -interpolações propostas;
- Elaboração de uma infra-estrutura de software para testes automatizados de interpolação de imagens médicas;
- Elaboração de um algoritmo para a criação de q -filtros genéricos;
- Desenvolvimento de um protótipo para a criação de q -filtros genéricos;
- Avaliação numérica do desempenho de q -filtros em função do índice entrópico q .

Esses ensaios representam os primeiros passos em se tratando de q -interpolação, um conceito até então inexistente. Problemas diferentes exigem olhares diferentes, e novas soluções trazem novos questionamentos. O estudo dessa família de q -filtros e q -Splines se mostra promissor, e os potenciais de uso vislumbrados até o momento englobam a melhoria do contraste de imagens, utilização artística dos q -filtros para efeitos especiais diversos, utilização em sistemas dinâmicos nos quais é observada a interação dinâmica construtiva ou destrutiva entre vários componentes de uma solução/formulação, usos em estatística, na modelagem de comportamentos e sistemas super/sub aditivos e aplicações a situações não solúveis pelos métodos clássicos, entre outros usos possíveis.

4.5 Trabalhos futuros

Os estudos realizados representam ensaios preliminares em assuntos bastante amplos, e certamente há possibilidade de desdobramentos, dentre os quais, destacamos:

- Melhoria da abordagem por meio de uma vetorização mais adequada a superfícies spline: foi utilizada uma abordagem simplificada e uniforme para toda a imagem, que poderia ser melhorada por meio de uma abordagem vetorizada, de tal modo que a imagem fosse representada de modo otimizado por meio de splines;
- Uso das Q-Splines para realce de imagens (sobretudo bordas): as splines desenvolvidas poderiam ser utilizadas para melhoria de contraste das imagens, com efeito marcante sobre o realce de bordas;
- Ampliação dos estudos com filtros Q-Gaussianos: a análise poderia ser expandida englobando os espectros de frequência dos filtros, bem como caracterizações dos principais grupos q-Gaussianos;
- Estudo do uso das Q-Gaussianas entre variáveis aleatórias q-relacionadas.

Esses desdobramentos ilustram, mas não esgotam as possibilidades de investigação, que poderia ainda seguir pelo veio do uso artístico da ferramenta, como a modelagem de objetos 3D e sua implementação em alguma ferramenta gráfica, como o Blender e outros programas do gênero.

Referências Bibliográficas*

REFERÊNCIAS BIBLIOGRÁFICAS *

- [1] MEDICINA, C. F. de. *Resolução CFM Nº 1.973/2011*. Conselho Federal de Medicina, acesso em 12/01/2013. Website. Disponível em: <http://www.portalmedico.org.br/resolucoes/CFM/2011/1973_2011.htm>.
- [2] MENEZES, M. R. de. Intervenção orientada por imagem. *Radiologia Brasileira*, v. 39, n. 4, 2006. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-39842006000400001&lng=pt>.
- [3] TSALLIS, C. Possible generalization of boltzmann-gibbs statistics. *Journal of Statistical Physics*, v. 52, p. 479–487, 1988.
- [4] GARROD, C. *Statistical Mechanics and Thermodynamics*. [S.l.: s.n.], 1995.
- [5] TSALLIS, C. *Currículo Lattes*. CNPq, acesso em 12/01/2013. Website. Disponível em: <<http://buscatextual.cnpq.br/buscatextual/visualizacv.do?metodo=apresentar&id=K4797532Z4>>.
- [6] WIKIPEDIA. *Constantino Tsallis*. Wikipedia, acesso em 12/01/2013. Website. Disponível em: <http://pt.wikipedia.org/wiki/Constantino_Tsallis>.
- [7] TSALLIS, C. *Introduction to nonextensive statistical mechanics: approaching a complex world*. [S.l.: s.n.], 2009.

*De acordo com a Associação Brasileira de Normas Técnicas, NBR 6023.

- [8] TSALLIS, C.; ABE, S.; OKAMOTO, Y. Nonextensive statistical mechanics and thermodynamics: Historical background and present status. *Nonextensive Statistical Mechanics and its Applications*, v. 560, p. 3–98, 2001.
- [9] BROER, H.; STRIEN, F. D. S. van; TAKENS, F. Structures in dynamics – finite dimensional deterministic studies. *Studies in Mathematical Physics*, p. 25–52, 1991.
- [10] HILEY, B. J.; JOICE, G. S. The ising model with long-range interactions. *Proceedings of the Physical Society*, v. 85, p. 493–507, 1965.
- [11] TISZA, L. *Generalized Thermodynamics*. [S.l.: s.n.], 1966.
- [12] GRAD, H. Asymptotic theory of the boltzmann equation. *Proceedings of 3rd International Symposium on Rarefied Gas Dynamics*, I, p. 26–59, 1963.
- [13] FISHER, M. E.; LEBOWITZ, J. L. Asymptotic free energy of a system with periodic boundary conditions. *Communications on Mathematical Physics*, v. 19, p. 251–272, 1970.
- [14] LANDSBERG, P. T. *Thermodynamics and Statistical Mechanics*. [S.l.: s.n.], 1990.
- [15] THIRRING, W. Stability of matter. *Foundations of Physics*, v. 20, p. 1103–1110, 1990.
- [16] KAMPEN, N. G. van. *Stochastic Processes in Physics and Chemistry*. [S.l.: s.n.], 1981.
- [17] LANDSBERG, P. T. Is equilibrium always an entropy maximum? *Journal of Statistical Physics*, v. 35, p. 159–169, 1984.
- [18] HUANG, X.-P.; DRISCOLL, C. F. Relaxation of 2d turbulence to a metaequilibrium near the minimum enstrophy state. *Physical Review Letter*, v. 72, p. 2187–2190, 1994.

- [19] TAGUCHI, Y.-H.; TAKAYASU, H. Power law velocity fluctuations due to inelastic collisions in numerically simulated vibrated bed of powder. *Europhysics Letter*, v. 30, n. 8, p. 499–504, 1995.
- [20] CLAYTON, D. C. Maxwellian relative energies and solar neutrinos. *Nature*, v. 249, p. 131, 1974.
- [21] MADDIX, J. When entropy does not seem extensive. *Nature*, v. 365, p. 103, 1993.
- [22] COLLABORATION, T. C. *Transverse momentum and pseudorapidity distributions of charged hadrons in pp collisions at $\sqrt{s} = 0.9$ and 2.36 TeV*. Cornell University Library, acesso em 12/01/2013. Website. Disponível em: <<http://arxiv.org/abs/1002.0621>>.
- [23] COLLABORATION, T. C. *LHC confirma teoria de físico brasileiro*. Inovação tecnológica, acesso em 12/01/2013. Website. Disponível em: <<http://www.inovacaotecnologica.com.br/noticias/noticia.php?artigo=lhc-confirma-teoria-fisico-brasileiro>>.
- [24] CURADO, E. M. F.; TSALLIS, C. Generalized statistical mechanics: Connection with thermodynamics. *Journal of Physics A*, v. 24, p. L69–L72, 1991.
- [25] CURADO, E. M. F.; TSALLIS, C. (corrigenda) generalized statistical mechanics: Connection with thermodynamics. *Journal of Physics A*, v. 24, p. 3187, 1991.
- [26] CURADO, E. M. F.; TSALLIS, C. (corrigenda) generalized statistical mechanics: Connection with thermodynamics. *Journal of Physics A*, v. 25, p. 1019, 1992.
- [27] UMAROV, S.; TSALLIS, C.; STEINBERG, S. A generalization of the central limit theorem consistent with nonextensive statistical mechanics. *Condensed Matter*, v. 04, 2006.

- [28] BORGES, E. P. A possible deformed algebra and calculus inspired in nonextensive thermostatistics. *Physica A*, Elsevier, n. 340, p. 95–101, 2004.
- [29] WIKIPEDIA, S.; BOOKS, L. *Algebraic Structures: Group, Boolean Algebra, Field, Ring, Ideal, Monoid, Semigroup, Algebraic Structure, Kleene Algebra, Magma, Lattice*. General Books LLC, 2010. ISBN 9781157620518. Disponível em: <<http://books.google.com.br/books?id=1axJSgAACAAJ>>.
- [30] EVARISTO, J.; PERDIGÃO, E. *Introdução à Álgebra Abstrata*. UFAL, 2013. Disponível em: <<http://professor.ic.ufal.br/jaime/livros/Algebra.htm>>.
- [31] KHURI, A. I. *Advanced Calculus with Applications in Statistics*. New Jersey: John Wiley and Sons, 2003. 673 p.
- [32] BOLDRINI, J. L. et al. *Álgebra Linear*. São Paulo: Harbra, 1986. 411 p.
- [33] CARVALHO, A. B. de. A filosofia da educação moderna: Bacon e descartes. *Objetos Educacionais: Textos*, Acervo Digital da Unesp, <http://www.acervodigital.unesp.br/handle/123456789/127>, acesso em 16/09/2012, 2010.
- [34] PAPINEAU, D. O falsificacionismo de karl popper. A arte de pensar, http://www.aartedepensar.com/leit_falsificacionismo.html, Acesso em 16/09/2012.
- [35] LIAN, Y.; JUNCHENG, L.; GUOHUA, C. A class of algebraic trigonometric interpolation splines and applications. In: *Computational and Information Sciences (ICCIS), 2010 International Conference on*. [S.l.: s.n.], 2010. p. 1174 –1177.
- [36] WEISSTEIN, E. W. *Weierstrass Function*. MathWorld–A Wolfram Web Resource, acesso em 2012. Website. Disponível em: <<http://mathworld.wolfram.com/WeierstrassFunction.html>>.

- [37] KIRK, D. B.; HWU, W. mei W. *Programming massively parallel processors: a hands on approach*. Burlington: Elsevier, 2012. 258 p.
- [38] NVIDIA. *Cuda C Programming Guide*. Online: NVIDIA, 2010. 183 p.
- [39] NEMA. *DICOM Homepage*. NEMA - National Electric Manufacturers Association of EUA, acesso em 2012. Website. Disponível em: <<http://medical.nema.org>>.
- [40] WIKIPEDIA. *DICOM*. Wikipedia, acesso em 2012. Website. Disponível em: <<http://pt.wikipedia.org/wiki/DICOM>>.
- [41] WANG, Z.; SHEIKH, H. R. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, IEEE Press Piscataway, v. 14, n. 4, p. 14, 2004.
- [42] UNSER, M.; EDEN, M. B-spline signal processing, part i - theory. *IEEE Transactions on Signal Processing*, v. 41, n. 3, p. 821–833, 1993.
- [43] WIKIPEDIA. *Prova de Conceito*. Wikipedia, acesso em 12/01/2013. Website. Disponível em: <http://pt.wikipedia.org/wiki/Prova_de_conceito>.
- [44] MEIJERING, E. A chronology of interpolation: from ancient astronomy to modern signal and image processing. In: *Proceedings of IEEE*. [S.l.: s.n.], 2002. v. 90, p. 319–342.
- [45] NEUGEBAUER, O. *Astronomical Cuneiform Texts. Babylonian Ephemerides of the Seleucid Period for the Motion of the Sun, the Moon, and the Planets*. London: Lund Humphries, 1955.
- [46] NEUGEBAUER, O. *A History of Ancient Mathematical Astronomy*. Berlin: Springer-Verlag, 1975.
- [47] TOOMER, G. S. 'Hipparchus', in *Dictionary of Scientific Biography*, C. C. Gillispie and F. L. Holmes, vol. XV. New York: Charles Scribner's Sons, 1978.

- [48] BRUMMELEN, G. van. Lunar and planetary interpolation tables in ptolemy's almagest. *Journal for the History of Astronomy*, Charles Scribner's Sons, v. 25, n. 4, p. 297–311, 1999.
- [49] YAN, L.; LUN), D. S. E. translation by J. N. C. . A. W.-C. *Chinese Mathematics: A Concise History*. Oxford: Clarendon Press, 1987.
- [50] GUPTA, R. C. Second order interpolation in indian mathematics up to the fifteenth century. *Indian Journal of History of Scienc*, v. 4, n. 1 and 2, p. 86–98, 1969.
- [51] KENNEDY, E. S. The chinese-uighur calendar as described in the islamic sources. *Science and Technology in East Asia*, Science History Publications, New York, p. 191–199, 1977.
- [52] GREGORY, J. Letter to j. collins (23 november 1670). *James Gregory Tercentenary Memorial*, G. Bells and Sons, p. 118–137, 1939.
- [53] NEWTON, I. Letter to j. smith (8 may 1675). *The Correspondence of Isaac Newton*, Cambridge University Press, Cambridge, p. 342–345, 1959.
- [54] NEWTON, I. Methodus differentialis. *Analysis per Quantitates, Series, Fluxiones, ac Differentias: cum Enumeratione Linearum Tertii Ordinis*, W. Jones, London, p. 93–101, 1711.
- [55] FRASER, D. C. Newton and interpolation. *Isaac Newton 1642-1727: A Memorial Volume*, G. Bell and Sons, London, p. 45–69, 1927.
- [56] FRASER, D. C. Newton's interpolation formulas. *Journal of the Institute of Actuaries*, C. & E. Layton, London, 1927.
- [57] NEWTON, E. t. b. F. C. I. *Philosophiae Naturalis Principia Mathematica*. Berkeley: University of California Press, 1960.
- [58] WARING, E. Problems concerning interpolations. *Philosophical Transactions of the Royal Society of London*, London, v. 69, p. 59–67, 1779.

-
- [59] LAGRANGE, J. L. Leçons elementaires sur les mathematiques donnees a l'école normale. *Euvres de Lagrange*, J.-A. Serret, Gauthier-Villars, Paris, v. 7, p. 183–287, 1795.
- [60] FABER, G. Über die interpolatorische darstellung stetiger functionen. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, Mathematisch-Physikalische Klasse, v. 23, p. 192–210, 1914.
- [61] FEJÉR, L. Ueber interpolation. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen*, Mathematisch-Physikalische Klasse, 1916.

Glossário

Absorção

A propriedade de absorção entre operações estabelece que o resultado de um elemento operado com o resultado da outra operação envolvendo o mesmo operando e um outro qualquer resulta no operando inicial. De um certo modo, é como, como se um operando ‘absorvesse’ o outro, por isso essa nomenclatura. Formalmente, tomando duas operações binárias ‘\$’ e ‘%’, a lei de absorção diz que, se o conjunto atende a essa lei, então $\forall a, b \in G, a$(a\%b) = a\%(a$b) = a$. Um exemplo de conjuntos que seguem essa lei são conjuntos booleanos.

Aditividade

Aditividade é uma propriedade dos sistemas com entropias extensivas (sistemas clássicos, com entropia de Boltzmann-Gibbs), nos quais a soma da entropia do sistema resultante da união de ambos é igual à soma das entropias iniciais de cada sistema.

Ajuste de curva

Processo de se ajustar uma curva contínua a um conjunto discreto de dados, que pode ser feito por meio de interpolação (a curva passa através dos pontos) ou aproximação (a curva ‘passa próximo’ aos pontos).

Álgebra booleana

Uma álgebra booleana é um reticulado limitado, distributivo e complementado.

Anel

Anel é um conjunto com duas operações binárias, uma operação de grupo abeliano definida como adição, e uma operação de semigrupo definida como a multiplicação, que satisfaça a distributividade em relação à soma.

Aproximação

Processo de se criar uma função f que *passe próximo* dos pontos de um conjunto dado.

Associatividade

Uma operação $*$ é dita *associativa* quando $\forall a, b, c \in G, (a*b)*c = a*(b*c)$.

Balanceamento de carga

Balanceamento de carga é a distribuição equitativa de processamento entre as unidades de processamento.

Bibliotecas CUDA

Bibliotecas CUDA são conjuntos de ferramentas que utilizam a arquitetura CUDA a fim de otimizar a realização de processos e cálculos científicos. Dentre as bibliotecas CUDA disponibilizadas por diversas organizações e grupos, destacam-se: NPP, CUBLAS, THRUST, MATLAB, MATHEMATICA, OpenCV, CUFFT, e outras.

Campo

Um campo é um anel comutativo no qual cada elemento não nulo possui um inverso multiplicativo.

Comutatividade

Uma operação $*$ é dita *comutativa* quando a ordem dos argumentos da função $*$ não altera o resultado, ou seja, $\forall a, b \in G, a*b = b*a$.

Constantino Tsallis

Constantino Tsallis nasceu em Atenas na Grécia, emigrou para a

Argentina (Mendoza), onde completou o ensino fundamental e médio. Concluiu graduação e Mestrado em Física pelo Instituto Balseiro de Bariloche da Universidad Nacional de Cuyo (1961-1965). Imigrou para o Brasil em 1975, e atualmente é professor titular do Centro Brasileiro de Pesquisas Físicas e membro da Academia Brasileira de Ciências e da Academia Brasileira de Ciências Econômicas, Políticas e Sociais [5, 6].

Coordenatização

Processo de se aplicar um sistema de coordenadas a um sistema ou situação real.

Corpo

Um corpo é um anel no qual os elementos não-zero formam um grupo abeliano sob multiplicação.

CUDA

CUDA é uma plataforma de desenvolvimento paralelo destinada a tirar proveito da grande capacidade de processamento presente nas placas de vídeo NVidia (GPUs).

CUDA GDB

CUDA GDB é o depurador CUDA fornecido pelo fabricante, juntamente com o CUDA Toolkit.

CUDA Toolkit

CUDA Toolkit é o kit de ferramentas de desenvolvimento disponibilizado pela NVIDIA voltado a CUDA.

Desvio padrão

O desvio padrão é uma medida estatística da dispersão de um conjunto de dados, frequentemente denotado pelo símbolo σ e sua expressão é $\sigma = \sqrt{E(x^2) - E^2(x)}$, em que $E(X)$ é o valor esperado da variável aleatória.

Diagnóstico por imagem

Diagnóstico por imagem é uma especialidade médica voltada ao uso das tecnologias de aquisição de imagens para realização de diagnósticos. No Brasil, essa especialidade é reconhecida pelo Conselho Federal de Medicina sob o nome de "Radiologia e diagnóstico por imagem" [1].

DICOM

DICOM é abreviação de *Digital Imaging and Communications in Medicine* (imagens e comunicação digital em Medicina), e consiste em um conjunto de normas para tratamento, armazenamento e transmissão de informação médica num formato eletrônico, sobretudo imagens e informações agregadas a elas, estruturando um protocolo. Foi criado com a finalidade de padronizar a formatação das imagens diagnósticas e compatibilizar seu uso e manipulação entre diversos aparelhos, processos e unidades de saúde.

Distributividade

Considerando duas operações, uma operação é dita *distributiva* em relação à outra quando for distributiva à esquerda e à direita em relação à outra operação.

Distributividade à direita

Considerando duas operações, uma operação é dita *distributiva à direita* em relação à outra quando pode ser 'distribuída' em relação à outra à direita sem alterar o resultado final. Como exemplo, citamos a distributividade da multiplicação em relação à soma, na qual temos que $\forall a, b, c \in G, (a + b) * c = (a * c) + (b * c)$.

Distributividade à esquerda

Considerando duas operações, uma operação é dita *distributiva à esquerda* em relação à outra quando ser 'distribuída' em relação à outra à esquerda, sem alterar o resultado final. A distinção à esquerda e à direita é importante, pois nem sempre as operações

consideradas são comutativas. Como exemplo disso, citamos o caso da distributividade à esquerda da multiplicação em relação à soma, temos que $\forall a, b, c \in G, a*(b + c) = (a*b) + (a*c)$.

Entropia extensiva

Entropia extensiva é aquela na qual o índice entrópico q é igual a 1, e equivale à entropia de Boltzmann-Gibbs.

Entropia subextensiva

Entropia subextensiva é aquela na qual o índice entrópico q é maior que 1, conferindo um comportamento novo ao sistema, como a sub-aditividade. Essa representação parece adequada a representar sistemas nos quais há interações intra-sistema quando da união dos dois sistemas, resultando na atenuação ou redução de determinadas características do sistema, no caso, a entropia, que se refletiria em propriedades macroscópicas do sistema resultante.

Entropia superextensiva

Entropia superextensiva é aquela na qual o índice entrópico q é menor que 1, conferindo um comportamento novo ao sistema, como a super-aditividade. Essa representação parece adequada a representar sistemas nos quais há interações intra-sistema quando da união dos dois sistemas, resultando na potencialização de determinadas características do sistema, no caso, a entropia, que se refletiria em propriedades macroscópicas do sistema resultante.

Enupla

Uma sequência finita e ordenada de objetos, cujas n posições podem representar um objeto em um espaço n -dimensional.

Erro quadrático médio (RMSE)

É a média do erro quadrático médio de duas funções ao longo

de uma série de pontos amostrados, ou seja, $RMSE = \frac{1}{n} \sum_i^n (a_i - b_i)^2$, considerando-se duas sequencias a e b .

Bimódulo

Um bimódulo é um módulo que é um R -módulo à esquerda e à direita.

Espaço vetorial

Um espaço vetorial é um R -**módulo** no qual o anel R é um campo.

R -módulo

O mesmo que módulo.

Estatística de Tsallis

O mesmo que Mecânica estatística não-aditiva.

Função de corda

Dado um triângulo inscrito em um círculo de raio R , com um dos vértices no centro do círculo e os outros sobre o círculo, a função corda associa a cada possível ângulo central A o valor da medida do lado oposto a ele, ou seja, a corda, pela equação: $corda(A) = 2 \cdot R \cdot \text{sen}(A/2)$.

Função de Weierstrass

Essa função, definida como $W_a(x) = \sum_{k=0}^{\infty} \frac{\text{sen}(\pi k^a x)}{\pi k^a}$, inicialmente para $a = 2$ e k natural, apesar de ser referenciada por muitos como ‘função de Weierstrass’, foi na verdade proposta por Riemann em 1861 em suas palestras, que afirmou que ela era contínua em todos os pontos, porém não diferenciável em um conjunto denso nos reais. Ela possui derivada para os valores de x pertencentes ao conjunto dos números ímpares. Posteriormente, provou-se que ela não é diferenciável para qualquer valor irracional [36].

Função holomórfica

Uma função holomórfica é uma função complexa de uma ou mais

variáveis complexas cuja derivada existe na vizinhança de cada ponto do seu domínio.

Função meromórfica

Uma função meromórfica em um subespaço aberto E do plano complexo é uma função que é holomórfica em todo E , exceto num conjunto de pontos isolados, chamados de *pólos* da função.

Gargalo

Um gargalo em computação é uma restrição em um módulo ou componente do sistema que impede que o pleno desempenho da solução como um todo. Geralmente essa expressão também subentende que os demais componentes do sistema poderiam operar num desempenho superior, que é restringido pelo componente que gera o gargalo.

GPU

GPU (Graphic Processing Unit) é o nome dado ao conjunto de processadores das placas de vídeo. Tais processadores evoluíram do que se chamava *pixel shaders*, e eram destinados a realizar cálculos específicos de iluminação, sombra e outros procedimentos atribuídos aos *pixels* em tempo de renderização da imagem.

Grupo abeliano

Um grupo abeliano é um grupo no qual a operação é comutativa.

Grupo

Um grupo é um monóide no qual a operação é inversível (possuem elementos inversos, o que implica também em *divisibilidade*, e vice-versa).

Idempotência

Uma operação é dita *idempotente* quando apresenta o mesmo resultado mesmo após ser aplicada várias vezes sobre o mesmo valor, ou seja, $\forall a \in G, a * a = a$. Se isso ocorrer somente para um

dado elemento a , então esse elemento é dito elemento idempotente da operação $*$, embora, nesse caso, a operação não o seja.

Identidade

Uma operação $*$ possui a propriedade de identidade quando possui um elemento neutro e para o qual, todo elemento que for operado com ele resulta no próprio elemento, ou seja $\forall a \in G, a * e = a$.

Índice entrópico

O índice entrópico q é um parâmetro introduzido por Tsallis na expressão da entropia a fim de possibilitar uma versão mais ampla da entropia de Boltzmann-Gibbs.

Integração em tempo de compilação

A integração em tempo de compilação é a possibilidade de integração de um programa escrito em uma linguagem compilada com qualquer linguagem via ligação de bibliotecas estáticas e dinâmicas.

Interpolação

Processo de se criar uma função f que *interpole* um dado conjunto de pontos dados.

Interpolação linear

Processo de se encontrar um conjunto de segmentos de reta que passe através de um conjunto de pontos consecutivos, de modo que cada segmento ligue dois pontos consecutivos.

Interpolação osculatória

Método que interpola não somente os pontos dados, mas também as derivadas de ordens maiores da função nos pontos dados.

Interpolar

Dizemos que uma função f *interpola* um conjunto de pontos P quando, para cada ponto p_i do conjunto P , o elemento $f(p_i)$ pertence à imagem de f , ou seja, $\forall p_i \in P \rightarrow f(p_i) \in Im(f)$.

Invertibilidade

Uma operação $*$ é dita *inversível* quando ela é reversível (seus efeitos são reversíveis), ou seja, dado um elemento na imagem da operação, o argumento original pode ser recuperado a partir da operação inversa. Isso pressupõe então que, dado um elemento do conjunto, existe um elemento inverso associado. O conceito de inversão varia de acordo com a operação e a estrutura algébrica em questão. Como exemplo, a subtração em relação à adição, e de inverso (multiplicativo) em relação à multiplicação. Nos espaços de funções, a inversão seria a transformação que leva elementos da imagem de uma aplicação em seus correspondentes no domínio da função.

Invólucros CUDA

Invólucros CUDA são interfaces que permitem a utilização de CUDA com outras linguagens que não foram implementadas nativamente, como Java e Python e possivelmente outras, com a recente abertura do código-fonte pela NVIDIA.

Laço

Um laço é um quase-grupo com elemento identidade.

Magma

Um magma é um par $(G, *)$, no qual G é um conjunto, munido de uma única operação binária $*$: $G \times G \rightarrow G$, ou seja, $\forall a, b \in G : a * b \in G$, a qual é fechada por definição, porém, sem que qualquer outro axioma seja imposto a tal operação.

Mecânica estatística

A Mecânica Estatística é um ramo da ciência que se utiliza da Teoria das Probabilidades para calcular propriedades macroscópicas de sistemas físicos constituídos por muitas partículas [4].

Mecânica estatística não-aditiva

A Mecânica estatística não-aditiva representa um novo paradigma,

proposto em 1988 por Constantino Tsallis [3] e fundamenta-se em uma nova definição para a entropia, da qual diversas consequências surgem, entre elas, a não-aditividade da entropia da soma de dois sistemas. A definição de entropia proposta por Tsallis é

$$S_q = k \frac{1 - \sum_{i=1}^W p_i^q}{q - 1} \quad \left(\sum_{i=1}^W p_i = 1, q \in \mathbb{R} \right).$$

Média

A média aritmética é uma medida estatística que resume a posição média de um conjunto de dados, é geralmente representada pelo símbolo μ e é dada pela expressão $\mu = \frac{1}{n} \sum_{i=1}^n x_i$, considerando n valores de dados.

Método científico

Conjunto de regras básicas elaboradas com o uso da lógica, acerca de como se deve proceder no intuito de se produzir conhecimento científico, quer seja um novo conhecimento ou fruto da integração, correção, evolução ou expansão de conhecimentos pré-existentes. Em geral, o método consiste na análise lógica e sistemática de evidências obtidas por meio da análise do resultado de experimentos de campo ou em ambiente controlado, a fim de se verificar ou falsear uma determinada conjectura.

MPI

Message Passing Interface (Interface de transmissão de mensagens) é uma linguagem nascida em 1992 e destinada a aplicações de supercomputação, especialmente para clusters.

Módulo

Um módulo, ou R -módulo (à esquerda) M sobre um anel R é um grupo abeliano M com um mapeamento $\phi : R \times M \rightarrow M$ satisfazendo:

- Identidade para multiplicação;

- Distributividade da multiplicação em relação à soma;
- Associatividade da multiplicação.

Um *módulo* é comutativo à direita quando é definido de modo análogo, porém com os elementos de R multiplicados pela direita ($\phi : M \times R \rightarrow M$).

Um *bimódulo* é um módulo que é um R -módulo à esquerda e à direita, de tal modo que o resultado das multiplicações são os mesmos (embora isso não exija que a multiplicação seja comutativa, como ocorre no caso de espaços matriciais).

Se o anel R é comutativo, então os R -módulos à esquerda são o mesmo que os R -módulos à direita, e são denominados simplesmente de R -módulos.

Momento

Representa o valor esperado de uma variável aleatória, $E[X]$. Para uma variável aleatória X contínua, com distribuição de probabilidade $p_X(x)$, seu n -ésimo momento é dado por $\mu_n = E(x^n) = \int_{-\infty}^{\infty} p_X(x)x^n dx$.

Monóide

Um monóide é um semigrupo com elemento identidade.

Nó de uma spline

Ponto que marca a fronteira entre dois trechos da spline (ou entre duas subfunções).

Nsight

O Nsight é um depurador CUDA para windows (Visual Studio) e para linux (Eclipse).

nvcc

Nvcc é o compilador CUDA prepara código para compilação, e utiliza um compilador nativo para compilar. Dentre suas características, ele:

- Utiliza a API CUDA;
- Suporta linguagens como C, C++, Fortran, OpenCL, OpenMP, entre outras;
- Utiliza C++ com restrições: Polimorfismo, valores padrão, sobrecarga de operadores, namespaces e classes (para GPUs com capacidade de computação 2.x).

É parte integrante do CUDA Toolkit.

OpenCL

Open Computing Language (Linguagem de computação aberta) é uma linguagem desenvolvida pela Apple (detendora dos direitos) e aperfeiçoada com a colaboração das equipes técnicas da AMD, IBM, Intel, e Nvidia. O grupo Khronos Compute Working, formado por representantes de fabricantes de CPU, GPU, microprocessadores e companhias de software lançaram sua primeira versão aberta em 18 de novembro de 2008.

OpenMP

Open Multiprocessing (Multiprocessamento aberto) é uma linguagem de multiprocessamento lançada pela organização não governamental *OpenMP Architecture Review Board - OpenMP ARB*, em outubro de 1997, consórcio representante das empresas AMD, IBM, Intel, Cray, HP, Fujitsu, Nvidia, NEC, Microsoft, Texas Instruments, Oracle Corporation, entre outras.

Operação binária

Uma operação binária $*$ é uma relação entre o **produto cartesiano** de um conjunto consigo mesmo e o próprio conjunto, ou seja, $S \times S \rightarrow S$, que leva um par ordenado $(a, b) \in S \times S$ em $c = a*b \in S$. Por levar elementos de S em S , a operação é dita *fechada*, ou que possui a propriedade de *fechamento*.

Paralelização de laço

Paralelização de laço é o processo de se abrir um laço (como o *while* ou *for*) em diversos comandos separados, de modo que possam ser executados em paralelo, cada um por uma linha de execução diferente.

Parametrização

Parametrização é o processo de generalização de modelos, objetos, métodos ou comportamentos por meio da definição de parâmetros necessários ou relevantes ao contexto, representando-os em termos dos parâmetros definidos. Esse processo mantém a estrutura da solução parametrizada, permitindo que o mesma idéia ou processo se aplique a condições de entrada diferente. Em Matemática, por exemplo, a parametrização de curvas e superfícies resulta no lugar geométrico de todos os pontos que as compõem, e cada um deles pode ser expresso em termos de um determinado conjunto de parâmetros de entrada.

Particionamento de processo

Particionamento de processo é a divisão de um método, processo ou procedimentos de modo a permitir que as unidades resultantes possam ser executadas em paralelo. O procedimento deve possuir algum paralelismo que possa ser explorado usando a GPU.

Polinômios de Hermite

Polinômios de Hermite são polinômios ortogonais propostos pelo matemático Charles Hermite e definidos como: $h_n(x) = (-1)^n e^{x^2/2} \frac{d^n(e^{-x^2/2})}{dx^n}$, $n = 1, 2, 3, \dots$

Produto cartesiano

O produto cartesiano é uma construção por meio da qual se elabora um conjunto a partir de um ou mais conjuntos dados. Cada elemento do conjunto resultante é a seleção ordenada de um elemento de cada conjunto, originando uma tupla, ou uma sequência

ordenada. Como exemplo, a partir de um conjunto S , podemos construir um produto cartesiano $P = S \times S$, e cada elemento de P é um par ordenado do tipo (a, b) com $a \in S$ e $b \in S$. Se usarmos dois conjuntos, temos $Q = R \times S$ de modo que $\forall (a, b) \in Q, a \in R$ e $b \in S$. Esse processo pode se estender o quantos desejarmos ($R \times S \times T \times U \times \dots$), produzindo tuplas (ou enuplas). No caso de um produto cartesiano entre dois conjuntos, a tupla (a, b) é chamada de *par ordenado*.

Prova de conceito

Uma prova de conceito é um termo utilizado para denominar um modelo prático que possa provar a factibilidade ou viabilidade técnica de um conceito teórico. Em termos computacionais, representa uma implementação que, em geral, é resumida ou incompleta, de um método ou de uma ideia, realizada com o propósito de verificar que o conceito ou a teoria em questão é suscetível de ser explorado de uma maneira útil [43].

q-estatística

O mesmo que Mecânica estatística não-aditiva.

Quase-grupo

Um quase-grupo é um magma não vazio no qual a divisão é sempre possível, o que equivale a dizer que para todo elemento pertencente ao magma, há um inverso.

Raio de corte

O raio de corte representa o raio utilizado para limitar a região de amostragem do filtro, considerando um círculo de corte com centro na origem.

Relação sinal-ruído

A relação sinal ruído (ou **SNR** – *Signal Noise Ratio*) é muito utilizada no ramo das telecomunicações, e serve para representar a

relação entre o sinal e o ruído nele presente. É definida como a relação da potência do sinal sobre a potência do ruído, ou seja,

$$SNR = \frac{P_{sinal}}{P_{ruído}}.$$
Reticulado

Um reticulado é um conjunto com duas operações comutativas, associativas e idempotentes, que satisfazem a lei de absorção.

Semi-reticulado

Um semi-reticulado é um semigrupo no qual a operação é comutativa e idempotente.

Semigrupo

Um semigrupo é um magma no qual a operação é associativa.

Similaridade estrutural

A similaridade estrutural é uma métrica de qualidade elaborada sob a premissa de que o sistema perceptual humano é altamente adaptado à extração de características estruturais de uma cena, proposta por Wang e Sheikh em 2004 [41], e implementado no matlab e na forma de um plugin do ImageJ.

Sincronismo

Sincronismo é o processo de sincronização de comandos entre diversos processos ou *threads*, de modo a executar um determinado algoritmo corretamente.

Spline

Tomado um intervalo $[a, b]$ particionado em $m + 2$ trechos, de tal modo que $a = \tau_0 < \tau_1 < \dots < \tau_m < \tau_{m+1} = b$ sejam pontos de partição em $[a, b]$, uma *spline* de grau n , $S^n(x)$, com **nós** nos pontos $\tau_1, \tau_2, \dots, \tau_m$ é uma função constituída por partes que possui as seguintes propriedades:

1. $S^n(x)$ é um polinômio de grau menor ou igual a n em cada subintervalo $[\tau_{i-1}, \tau_i]$, com $1 \leq i \leq m + 1$;

2. $S^n(x)$ possui derivadas contínuas até a ordem $n - 1$ em $[a, b]$.

Obviamente, como a spline é um conjunto de funções, dizer que o grau de $S^n(x)$ é menor ou igual a n é dizer o mesmo sobre o grau de cada subfunção $s_i^n(x)$ que a compõe.

Sub-aditividade

Sub-aditividade é uma propriedade dos sistemas com entropias subextensivas, nos quais a soma da entropia do sistema resultante da união de ambos é inferior à soma das entropias iniciais de cada sistema.

Subfunção

Uma subfunção de uma spline é uma função definida em uma partição do domínio da spline que, juntamente com outras subfunções, constituem a spline.

Super-aditividade

Super-aditividade é uma propriedade dos sistemas com entropias superextensivas, nos quais a soma da entropia do sistema resultante da união de ambos é superior à soma das entropias iniciais de cada sistema.

Tamanho da janela

Esse termo é utilizado em nosso trabalho para denotar o tamanho mínimo (em *pixels*) da matriz de filtros referenciados no texto (medido horizontal ou verticalmente). Como exemplo, uma janela de tamanho 5 representa uma janela 5×5 de *pixels*.

Totalidade

Uma operação $*$ é dita *total* quando pode ser aplicada a todos os elementos de um conjunto G . No caso em que a operação se aplique apenas a um subconjunto de G , a operação $*$ é dita *parcial*.

Tupla

O mesmo que ênupla, ou seja, uma sequencia finita e ordenada de objetos.

Interpolação: notas históricas

A.1 Introdução

Ajuste de curva é um termo que adotaremos para nomear o processo de se representar um conjunto de dados esparsos por meio de uma curva contínua. Esse termo representa bem o processo de *modelagem* que inspirou a busca de funções contínuas na representação de dados discretos.

Essa prática é muito comum no tratamento de dados experimentais e é utilizada em praticamente todas as áreas do conhecimento. Dela nascem teorias e modelos nas mais diversas linhas, sendo também utilizada em áreas relacionadas ao design de peças, objetos e equipamentos por meio do computador, controle de qualidade e em diversas outras áreas. Sendo assim, pode-se dizer que o ajuste de curva é um processo indispensável à vida moderna, e que da sua utilidade advém sua importância.

A teoria matemática envolvendo o ajuste de curvas se desenvolveu enormemente desde suas origens, a fim de abranger diversas situações e propriedades geométricas e físicas dos objetos, como o

Figura A.1: Exemplo de ajustes de curva.

movimento de partículas sobre uma curva final. De um modo geral, a motivação para se aplicar um processo de ajuste de curva está intrinsecamente ligada:

1. Ao formato dos dados que se quer modelar;
2. Às características da curva resultante.

Nesse aspecto, é compreensível que dados em formatos diferentes exijam abordagens diferentes e que atributos diferentes da curva (ou superfície) a ser elaborada devam ser determinadas de modo distinto. Dentre as características desejáveis, podemos citar: precisão, geometria, continuidade de derivadas e minimização de energia de movimento sobre a curva, e muitas outras mais. Dada a multiplicidade de objetivos imaginados pelos estudiosos, o campo do ajuste de curvas recebeu inúmeras contribuições, tornando o assunto vasto e com uma grande variedade de abordagens. Apesar disso, os processos de ajuste de curva podem ser enquadrados basicamente em duas categorias:

- **Interpolação:** a curva a ser obtida passa *através* dos pontos desejados.
- **Aproximação:** a curva a ser obtida passa *próximo* aos pontos desejados (o conceito de ‘proximidade’ depende da abordagem a ser utilizada).

Um método ou uma abordagem específica sempre pertencerá a uma dessas categorias, e seu uso geralmente implicará na imposição restrições, as quais são responsáveis por suas características marcantes, e que lhe permitem cumprir os objetivos estabelecidos para seu uso.

O processo de ajuste de curva é frequentemente mencionado por uma de suas categorias, interpolação ou aproximação, mas, de fato, ambos representam o processo de se encontrar uma curva suave que modele, sob algum ponto de vista, os dados originais.

A.2 Histórico

O processo de ajuste de curvas teve início com coleta de dados da natureza, especificamente dados astronômicos. O homem havia percebido que o clima se alternava em períodos, e que esses períodos alternavam-se entre si. As primeiras tentativas bem sucedidas de se prever a aproximação desses períodos nasceram da observação dos astros nos céus. Os astrólogos¹³ da época observavam os céus, criando longas tabelas de posições dos astros que, ao longo de dado período, se repetiam, assim como as estações do ano. Essas tabelas se mostraram úteis, sobretudo na agricultura, permitindo a elaboração de estratégias de plantio e colheita baseadas em calendários. E mesmo com todo o esforço da época, é claro que tais tabelas continham falhas e ausência de informação, pois nem sempre se podiam obter os dados, devido a condições meteorológicas e ao fato de que certos astros não são visíveis em determinados períodos do ano.

Os registros mais antigos de tabelas astronômicas provém de Uruk e da Babilônia, do período selêucida (cerca de 300 a.C.)[44]. A partir do estudo dessas tabelas, o historiador-matemático Neugebauer concluiu que foram utilizados métodos de interpolação para se preencher suas lacunas [45, 46]. O estudo realizado por Neugebauer revelou ainda que outros métodos de interpolação além do linear foram utilizados, embora a técnica precisa com que foram elaborados não tenha sido preservada.

A interpolação linear, já conhecida desde essa época, também foi utilizada por Hiparco de Rodes [47] no cálculo das **funções de corda**,

Figura A.2: Representação antiga da astrologia.

¹³Ou, mais precisamente os astrônomos da época, pois não havia distinção naquele tempo entre astrologia e astronomia.

para calcular a posição de astros e por Cláudio Ptolomeu com o mesmo objetivo [48]. Este último foi o autor da teoria heliocêntrica, a qual perdurou até o século XVI, e utilizava uma abordagem na qual tabulava as funções nos valores para os quais a variação fosse maior com uma mudança na variável independente.

A primeira pessoa a usar interpolação de segunda ordem para o cálculo das posições do sol e da lua foi o astrônomo chinês Liù Zhuó, a fim de construir o “Calendário Imperial Padrão”, por volta do ano 600 d.C. [49]. Em notação moderna, a fórmula que ele utilizou pode ser escrita como:

$$f(x_0 + \xi T) = f(x_0) + \frac{\xi}{2}(\Delta_1 + \Delta_2) + \xi(\Delta_1 - \Delta_2) - \frac{\xi^2}{2}(\Delta_1 - \Delta_2) \quad (\text{A.17})$$

sendo $0 \leq \xi \leq 1$, $T > 0$, $\Delta_1 = f(x_0 + T) - f(x_0)$ e $\Delta_2 = f(x_0 + 2T) - f(x_0 + T)$ e $f(x_0)$, $f(x_0 + T)$ e $f(x_0 + 2T)$ os resultados observados respectivamente nos tempos x_0 , $x_0 + T$ e $x_0 + 2T$.

Embora tais métodos pudessem oferecer uma primeira aproximação para a posição dos astros, uma precisão maior só poderia ser obtida a partir de um método mais preciso, mais refinado do que uma simples aproximação de segunda ordem. Uma aproximação de terceira ordem foi utilizada por Guo Shoujing e outros no ano de 1280 para a elaboração do “Calendário Trabalhos e Dias” da China. Embora eles não tenham deixado uma formulação explícita, o estudo de seus cálculos demonstra que eles utilizaram esse princípio [44].

Na Índia, o matemático-astrônomo Brahmagupta propôs, no ano de 625, uma interpolação de segunda ordem para a função seno [50]. O original, em sânscrito, continha a seguinte fórmula:

$$f(x_0 + \xi T) = f(x_0) + \frac{\xi}{2} \{ \Delta f(x_0 - T) + \Delta f(x_0) \} + \frac{\xi^2}{2} \{ \Delta f(x_0) - \Delta f(x_0 - T) \} \quad (\text{A.18})$$

na qual $\Delta f(x_0) = f(x_0 + T) - f(x_0)$. Além dessa formulação, várias outras contribuições são devidas aos desenvolvimentos no oriente, como diversos esquemas de interpolação parabólica, encontrados em manuscritos árabes e persas [51].

Independentemente dos avanços obtidos no oriente, o início da teoria da interpolação no ocidente deu-se após uma grande revolução no pensamento científico, impulsionado pelos avanços na Astronomia e Física. A interpolação se fez necessária nos estudos iniciados por Nicolau Copérnico, que chegaram a ser considerados

Figura A.3: *Nicolau Copérnico.*

um contra-senso pelo próprio Copérnico, frente à teoria dominante do Geocentrismo, pois ele se baseou sobretudo em argumentos geométricos para elaborar tais modelos. Posteriormente, Kepler e Galileu prosseguiram seus estudos, aprofundando o abismo filosófico entre o pensamento científico dominante (geocentrismo) e essas novas teorias. Nesse mesmo período, Roger e Francis Bacon iniciaram um questionamento acerca do conhecimento axiomático, e das propostas inquestionáveis da Igreja, posteriormente desenvolvido por René Descartes, em seu “Discurso do Método”, dando força à razão como meio de busca da verdadeira natureza dos fenômenos observáveis. Começou-se a buscar provas concretas e, de modo contundente, Galileu contribuía com esse pensamento em alguns de seus trabalhos, usando a dialética como ferramenta principal. A abordagem racional de Galileu, apoiada no método cartesiano, foi uma de suas principais ferramentas para levar os estudiosos à reflexão e assim, ao menos dar-lhes razões lógicas para questionarem os antigos paradigmas. Galileu escreveu a obra “Diálogo sobre os dois principais sistemas do mundo” em 1630 e publicado em 1632, defendendo o sistema heliocêntrico por meio de um engenhoso diálogo entre três personagens: Salviati (que defende o heliocentrismo), Simplício (personagem que aparenta ser um pouco tonto e que defende o geocentrismo) e Sagredo (um personagem neutro, mas que termina por concordar com Salviati). Buscando apoiar seu ponto de vista, ele usa incorretamente o movimento das marés como prova de sua hipótese, argumento que, apesar de incorreto, denota a busca de indícios experimentais a fim de

corroborar teorias e ampliar o entendimento dos fenômenos naturais. Essa mesma obra o levou frente à Inquisição, perante a qual foi forçado a renegar todo o trabalho de sua vida.

Essa verdadeira revolução no pensamento da época ganhou o reforço de Isaac Newton, cujo entusiasmo produziu uma obra imensamente rica, resultando num legado para a humanidade, do qual fazem parte as bases da teoria clássica de interpolação [44]. Pouco antes de Newton revelar, em uma carta a Oldenburg, que havia obtido uma solução para o problema da interpolação, Gregório, em 1670, propôs a fórmula clássica da interpolação, encontrada numa carta sua a Collins [52]:

$$\begin{aligned}
 f(x_0 + \xi T) = f(x_0) + \xi \Delta f(x_0) &+ \frac{\xi}{2!} (\xi - 1) \Delta^2 f(x_0) \\
 &+ \frac{\xi}{3!} (\xi - 1)(\xi - 2) \Delta^3 f(x_0) \\
 &+ \frac{\xi}{4!} (\xi - 1)(\xi - 2)(\xi - 3) \Delta^4 f(x_0) + \dots
 \end{aligned} \tag{A.19}$$

As contribuições de Newton nesse campo podem ser encontradas em: a) uma carta endereçada a Smith em 1675 [53], b) em um manuscrito intitulado *Methodus Differentialis* publicado em 1711 [54], c) um manuscrito intitulado *Regula Differentiarium*, descoberto no século XX [55, 56], e d) Lemma V da sua célebre obra *Principia*, de 1687 [57]. No *Principia*, Newton apresenta uma fórmula equivalente à apresentada por Gregório (para intervalos igualmente espaçados mas que, aparentemente, foi desenvolvida de modo independente), e uma fórmula mais geral, para intervalos arbitrários.

Posteriormente, Waring publicou, em 1779 [58], uma fórmula alternativa à proposta de Newton, que não requeria o cálculo das dife-

renças divididas ou finitas, e que é expressa por:

$$\begin{aligned}
 f(x) = & f(x_0) \cdot \frac{(x-x_1)(x-x_2)(x-x_3)\dots}{(x_0-x_1)(x_0-x_2)(x_0-x_3)\dots} + \\
 & f(x_1) \cdot \frac{(x-x_0)(x-x_2)(x-x_3)\dots}{(x_1-x_0)(x_1-x_2)(x_1-x_3)\dots} + \\
 & f(x_2) \cdot \frac{(x-x_0)(x-x_1)(x-x_3)\dots}{(x_2-x_0)(x_2-x_1)(x_2-x_3)\dots} + \dots
 \end{aligned} \tag{A.20}$$

ou, em outras palavras,

$$f(x) = \sum_{i=0}^n f(x_i) \cdot \prod_{\substack{j=0 \\ j \neq i}}^n \frac{(x-x_j)}{(x_i-x_j)} \tag{A.21}$$

Essa fórmula é atualmente conhecida como fórmula de Lagrange, que, aparentemente ignorando os trabalhos de Waring, a publicou 16 anos depois [59]. Posteriormente, Gauss notou a conexão lógica entre as representações de Newton, Euler e Waring-Lagrange, embora não os mencione em seus trabalhos póstumos [44]. Ele também formulou uma expressão que ficou conhecida com fórmula de Newton-Gauss:

$$\begin{aligned}
 f(x_0 + \xi T) = f(x_0) + \xi \Delta f(x_0) & + \frac{1}{2!} \xi(\xi-1) \Delta^2 f(x_0 + T) \\
 & + \frac{1}{3!} (\xi+1)\xi(\xi-1) \Delta^3 f(x_0 + T) \\
 & + \frac{1}{4!} (\xi+1)\xi(\xi-1)(\xi-2) \Delta^4 f(x_0 + T) + \dots
 \end{aligned} \tag{A.22}$$

Foram várias as contribuições no campo da interpolação no século XIX, desenvolvidas paralelamente por diversas personalidades, como Bessel, Everett, Cauchy, Hermite e Weierstrass, entre muitos outros. Até esse momento, os modelos de interpolação eram predominantemente polinomiais, sobretudo pelas propriedades dos polinômios, como sua familiaridade, simplicidade e facilidade de manipulação. Durante esse século, os estudiosos consolidaram seu formalismo e uso, justificado por meio do “teorema da aproximação” de Weierstrass, que

estabelece que toda função contínua num intervalo fechado pode ser uniformemente aproximada por um polinômio. Porém, o teorema não apresentava qualquer meio de obter esse polinômio, e em pouco tempo tornou-se claro que ele não necessariamente se aplicava no caso de um polinômio que fosse forçado a passar por pontos pré-estabelecidos num dado intervalo, como no caso da interpolação.

Figura A.4: Augustin Louis Cauchy.

Em 1821, Cauchy encarou o problema da interpolação como sendo a divisão entre dois polinômios, e mostrou que a solução para esse problema é única, sendo a equação de Waring-Lagrange um caso especial, para o qual o segundo polinômio é igual a um. Diversas generalizações para interpolação multivariada começaram a aparecer na segunda metade do século XIX, mas em 1878, Hermite propôs uma solução de natureza diferente para o problema de

interpolação, elaborando um método que permitia obter a continuidade das primeiras derivadas, com a possibilidade de que estas também pudessem assumir valores pré-definidos em certos pontos [44].

A pesquisa na teoria da aproximação continuou intensa, e, no final do século XIX, a busca por métodos capazes de interpolar também as derivadas, motivada por aplicações práticas da área atuarial, deu origem à chamada **interpolação osculatória**, criando uma nova linha de estudo no campo da aproximação. As duas linhas seguiram paralelas, sendo que o conceito de spline originou-se no ramo da interpolação osculatória.

Aprofundando-se na interpolação polinomial, Méray e Runge elaboraram exemplos de funções **meromórficas**¹⁴, para as quais os interpoladores de Waring-Lagrange não convergiam uniformemente [44].

¹⁴Que é **holomórfica** em todo subespaço aberto do plano complexo, exceto nos pólos da função.

Um resultado mais geral foi obtido por Faber [60], que mostrou, em 1914, que para qualquer sistema triangular de pontos de interpolação, o correspondente processo de interpolação de Waring-Lagrange *não converge uniformemente a essa função*, embora tenha sido provado posteriormente que é possível construir polinômios que convergem ao sistema, por exemplo, usando o esquema de interpolação de Hermite proposto por Fejér [61], em 1916. Esses resultados deixaram clara a inflexibilidade do uso de polinômios, e sua limitada aplicabilidade no problema de interpolação.

Em 1915, Whittaker descobriu uma série de propriedades da função *sinc*¹⁵. Ele notou que, para certos valores de uma função f , correspondentes a valores equidistantes de seus argumentos $x_0, x_0 \pm T, x_0 \pm 2T, \dots$, era possível construir uma tabela de diferenças para interpolação, e que existem muitas outras funções para as quais se pode construir a mesma tabela de diferenças. A partir da fórmula de Newton-Gauss, ele obteve uma função que representava os valores dessa tabela, demonstrando que essa era a função cardinal:

$$C(x) = \sum_{k=-\infty}^{+\infty} f(x_0 + kT) \frac{\sin \frac{\pi}{T}(x - x_0 - kT)}{\frac{\pi}{T}(x - x_0 - kT)} \quad (\text{A.23})$$

a qual, além de ser cotabular com a função original desde que $C(x_0 + kT) = f(x_0 + kT)$ para todo $k \in \mathbb{Z}$, não possui singularidades e não possui componentes de período menor que $2T$.

¹⁵A denominação *sinc* é uma abreviatura para ‘sin cardinal’, que significa ‘seno cardinal’.

Índice Remissivo

A

- Ajuste de curvas
 - história, 123
- Anel, 15
- Aproximação, 22, 122

B

- B-Spline
 - Base, 26
 - Definição, 26
 - Pontos de controle, 27
- Bimódulo, 16
- Boltzmann, 7
- Booleana, álgebra, 15

C

- Capacidade de computação*, 47
- Compute capability*, 47
- Campo, 16
- Cauchy, 128
- Conceito, prova de, 79
- Copérnico, Nicolau, 125
- Corpo, 15
- Corte, raio de, 57
- CSV, 62
- CUDA, 37, 52
 - Compute capability*, 47
 - Arquitetura, 39
 - Interface de programação, 44
 - Linguagens, 40
 - Memória, arquitetura, 39
 - Modelo de desenvolvimento, 42
- CUDA, Ferramentas
 - Bibliotecas CUDA, 46
 - CUDA GDB, 45
 - CUDA Toolkit, 45

- Integração estática e dinâmica, 46
- Invólucros CUDA, 46
- Nsight, 45
- nvcc, 44

D

- Desvio padrão, 57
- Diagnóstico por imagem, 1
- DICOM, 50, 53
 - libDicom, 50

E

- Entrópico, índice, 7
- Entropia, 6
 - extensiva, 8
 - sub-extensiva, 8
 - super-extensiva, 8
- Espaço vetorial, 16
- Espectros de Fourier, 58
- Estatística, 51
 - de Tsallis, 51
 - Mecânica, 5
 - não-aditiva, 6, 51
 - não-extensiva, 51
 - q-estatística, 51
- Estrutural, similaridade, 52
- Estruturas algébricas
 - Álgebra booleana, 15
 - Anel, 15
 - Bimódulo, 16
 - Campo, 16
 - Corpo, 15
 - Espaço vetorial, 16
 - Grupo, 14
 - Grupo abeliano, 14
 - Laços, 13

- Módulo, 16
- Magma, 12
- Monóide, 13
- Quase-grupos, 13
- Reticulado, 15
- Semi-reticulado, 13
- Semigrupo, 13
- F**
 - FFT, 58
 - Fourier, 58
 - Fourier, espectro, 58
 - Fractal, 61
 - Dimensão, 61
 - Funções
 - Blending, 19
 - Mistura, 19
- G**
 - Galileu, 125
 - Gaussiana, q-Gaussiana, 51
 - Gibbs, 7
 - GPU, 37
 - Grupo, 14
 - Grupo abeliano, 14
- H**
 - Hermite
 - Polinômios, 19
 - Spline, 29
- I**
 - Índice entrópico, 7, 8, 56
 - Imagem
 - Diagnóstico por, 1
 - Insight Toolkit, 52
 - Interpolação, 22, 122
 - História da, 123
 - linear, 123
 - osculatória, 128
 - Interpolação osculatória, 128
 - Interpolação polinomial, História da, 128
 - Intervenção orientada por imagem, 1
 - ITK, 52
- J**
 - Janela, tamanho da, 57
- K**
 - Kepler, 125
- L**
 - Laço, 13
 - Lagrange
 - Fórmula de, 127
- M**
 - Média, 57
 - Método científico, 125
 - Módulo, 16
 - Magma, 12
 - Magnética
 - Ressonância, 1, 61
 - Mathematica, 52
 - Metodologia, 51
 - Minimamente invasivos, procedimentos, 1
 - Monóide, 13
 - MPI, 40
 - Mudança de base, 20
- N**
 - Não-extensividade, 8
 - Newton, Isaac, 126
 - Nuclear
 - Medicina, 1
 - NVIDIA, 38
- O**
 - OpenCL, 41
 - OpenCV, 48, 52
 - OpenMP, 40
 - Operações, Propriedades
 - Absorção, 15
 - Associatividade, 12
 - Comutatividade, 13
 - Distributividade, 15
 - Distributividade à direita, 15
 - Distributividade à esquerda, 14
 - Idempotência, 13
 - Identidade, 12
 - Invertibilidade, 12
 - Totalidade, 12
 - Orientada por imagem, intervenção, 1

- Ortogonais, polinômios, 21
- Osculatória, interpolação, 128
- P**
 - Paralelização, Métodos, 46
 - Balanceamento de carga, 47
 - Gargalo, 47
 - Paralelização de laços, 46
 - Particionamento de processo, 46
 - Sincronismo, 46
 - Polinômios, 19
 - Polinomiais, espaços vetoriais, 17
 - Probabilidades
 - Teoria, 5
 - Prova de conceito, 79
- Q**
 - q-Base, 51
 - Q-estatística, 6
 - q-Filtros, 54
 - Gráficos, 57, 86, 88, 92
 - Parâmetros, 56
 - Protótipo Java, 56
 - q-Gaussiana, 51, 54
 - Filtros, 54
 - Formulação 1D, 54
 - Formulação 2D, 56
 - q-Polinômios de Hermite, 63
 - q-Spline
 - Exemplo, 64, 98
 - q-Base, 51
 - q-Base 1D de Hermite, 94
 - Quase-grupos, 13
- R**
 - Röntgen, Wilhelm Conrad, 1
 - Raio de corte, 57
 - Raios-X, 1
 - Relação sinal-ruído, 52
 - Ressonância magnética, 61
 - Reticulado, 15
- S**
 - Shader, pixel*, 37
 - Semi-reticulado, 13
 - Semigrupo, 13
 - Similaridade estrutural, 52
 - Sinal-ruído, relação, 52
 - SNR, 61
 - Spline
 - Definição, 23
 - Exemplo, 25
 - Grau, 24
 - Pontos de controle, 31
 - Spline bidimensional
 - Definição, 32
 - Subfunção, 23
- T**
 - Tsallis
 - Constantino, 6
 - Estatística de, 5
- U**
 - Ultrassom, 1
- V**
 - Visão computacional, 48
- W**
 - Weierstrass
 - Gráfico, 99, 101, 102
 - Weierstrass, função, 28, 99