

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Extração de características robustas de superfícies NURBS

Amanda Carrijo Viana Figur

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-C²MC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Amanda Carrijo Viana Figur

Extração de características robustas de superfícies NURBS

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestra em Ciências – Ciências de Computação e Matemática Computacional. *EXEMPLAR DE DEFESA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Antonio Castelo Filho

Coorientador: Prof. Dr. Farid Tari

USP – São Carlos
Novembro de 2022

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

C316e Carrijo Viana Figur, Amanda
Extração de características robustas de superfícies
NURBS / Amanda Carrijo Viana Figur; orientador
Antonio Castelo Filho; coorientador Farid Tari. --
São Carlos, 2022.
67 p.

Dissertação (Mestrado - Programa de Pós-Graduação
em Matemática) -- Instituto de Ciências Matemáticas
e de Computação, Universidade de São Paulo, 2022.

1. Características Robustas. 2. Curva Ridge. 3.
Geometria Diferencial. 4. NURBS. 5. Preditor-
Corretor. I. Castelo Filho, Antonio, orient. II.
Tari, Farid, coorient. III. Título.

Amanda Carrijo Viana Figur

Extraction of Robust Features from NURBS surfaces

Master dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP, in partial fulfillment of the requirements for the degree of the Master Program in Computer Science and Computational Mathematics. *EXAMINATION BOARD PRESENTATION COPY*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Antonio Castelo Filho

Co-advisor: Prof. Dr. Farid Tari

USP – São Carlos
November 2022

Este trabalho é dedicado a todos que esperançam por uma Universidade Popular e em especial àqueles que lutam por um mundo sem exploração da Natureza.

Aos que não se lembram: nós somos natureza.

Em memória de Sebastião Divino Viana, Maria do Carmo Viana e aos mais de 600 mil mortos pela Covid-19 no Brasil.

*A morte dos meus avós era inevitável.
A morte dessas mais de 600 mil pessoas era evitável.*

AGRADECIMENTOS

Os agradecimentos principais são direcionados à Leila Viana, Adélio Darcy Figur, Nix Lopes de Araujo, Oliver Zancul Prado e Paula de Godoi Vianna, sem os quais eu não conseguiria estar aqui. Agradecimentos notáveis à Maira Andretta, Marina Andretta, Leandro Fiorini Aurichi, Carlos Henrique Grossi Ferreira, Thaís Jordão, Márcia Cristina Anderson Braz Federson, Éder Ritis Aragão Costa, Eduardo Tengan, Antonio Castelo Filho, Farid Tari e Maíra Duran Baldissera professores que foram essenciais no meu caminho até aqui e me ensinaram muito mais do que conceitos matemáticos. Agradecimento especial aos meus amigos Giovana Sachett Maia, Gabriel Nogueira Malta e Lucas Siqueira Cheim, que estiveram comigo em todos os momentos (bons e ruins) dessa jornada acadêmica, da graduação ao mestrado. Agradeço ainda ao meu amigo Thiago Rodrigo Ramos que manteve o meu interesse por matemática vivo nos tempos difíceis da graduação, ao meu amigo Iago Israel, que me apresentou simbolismos muito importantes para a minha vida, à minha amiga e vizinha Caroline Santos Corrêia, uma ótima companhia nos tempos de pandemia, à minha amiga Livia Maria Vieira Pereira, uma grande companheira dos estudos fora da matemática. Agradeço aos amigos Gustavo Nicolau Gonçalves e Heloisa dos Santos Muniz que conheci na reta final do mestrado e que têm sido companhias incríveis. Agradeço também Ezra Zinete Gurgel, nossos momentos juntos são raros, mas todos foram sempre muito impactantes. Agradeço aos amigos que fiz recentemente, aos amigos que (por diversas razões) não estiveram presentes em todos os momentos e às amigas que vieram e já se foram: vocês também são parte constituinte de toda a minha essência. Não posso me esquecer de agradecer também ao Jean Carlos de Aguiar Lelis, amigo de ensino médio que nutriu a sementinha do meu interesse por matemática, ao tio Edson Lucas Viana, que me deu meu primeiro computador quando era criança e plantou uma das sementinhas que originou este trabalho, à tia Ruth Aparecida Viana da Silva, que revisou meus trabalhos inúmeras vezes e em conjunto com a tia Múria Carrijo Viana, sempre nutriram meu amor à leitura e estudos. Agradecimentos aos tios Rosevert, Phillips, Lincoln e Celso Kaleb e às tias Vana, Euterp e Verena que eu sei que sempre poderei contar com o apoio. Ao meu primeiro orientador Roni Ederson Krause de Oliveira, que me introduziu no mundo da pesquisa ainda durante o Ensino Médio e estudou o conceito de vida feliz em Sêneca junto comigo; e, ao meu melhor professor de física Fabiano Caetano de Souza, que me estimulou a vir estudar em São Carlos.

Agradecimentos especiais são direcionados ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), que financiou meu trabalho no mestrado; à

estrutura da Universidade de São Paulo (USP) e seus funcionários, que me acolheram durante a graduação e o mestrado mesmo nos tempos mais difíceis; Aos organizadores e participantes do SIGGRAPH 2018, que me levaram a querer estudar computação gráfica; à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) que, em conjunto com o CNPq, fornece uma estrutura robusta (e invejável) à organização da pesquisa brasileira; e à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), que financia o grupo de pesquisa ao qual este trabalho está associado.

Agradecimento especial à minha gestora, na empresa que trabalho atualmente, Jacqueline Andrade Oliveira. Ela é uma pessoa incrível, que é muito boa em reconhecer o que tem de melhor nas pessoas. Obrigado pela paciência comigo, por me dar tantas oportunidades, valorizar o meu trabalho, confiar em mim e por estimular que eu concluísse meu mestrado logo.

Queria ressaltar que eu não tenho absolutamente nada a agradecer ao governo golpista em poder desde 2016, especialmente ao governo fascista que esteve governando o país durante todo o período do meu mestrado. Que esse período da política brasileira seja enterrado na lata de lixo da história. Espero jamais vivenciar um novo genocídio de toda a população brasileira e luto para que o genocídio à população negra e originária (iniciado em 1500) acabe o mais breve possível.

Espero que nós, trabalhadores da pesquisa brasileira, consigamos resistir a esses tempos difíceis e de escassez e que sejamos capazes de lutar para construir uma pesquisa (e sociedade) baseada em abundância. Que sejamos capazes de nos unir mesmo em meio às diferenças.

Agradeço por último à mim mesmo! Uma pessoa não-binária, bissexual, branca e da classe trabalhadora. Eu aceito ser tratade por qualquer pronome e gosto que me chamem de Figura!

*“[...] Uma esperança sincera
Cresceu no seu coração
E dentro da tarde mansa
Agigantou-se a razão
De um pobre homem esquecido
Razão porém que fizera
Em operário construído
O operário em construção.”
(Vinicius de Moraes)*

RESUMO

FIGUR, A. **Extração de características robustas de superfícies NURBS**. 2022. 67 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2022.

As características robustas são pontos e curvas com propriedades geométricas importantes de uma superfície. No caso de rostos humanos, o *nose ridge* já é uma característica conhecida e utilizada em métodos de reconhecimento facial. Esperamos que outras características robustas avancem e melhorem as ferramentas e métodos para identificação e reconstrução de rostos humanos. O objetivo deste projeto é implementar um algoritmo de extração de curvas ridge e pontos umbílicos sobre superfícies NURBS. Esperamos que este trabalho abra caminho para que tenhamos um melhor entendimento da geometria do rosto humano no futuro.

Palavras-chave: Características Robustas, Curva Ridge, Superfícies parametrizadas por B-Splines, Reconhecimento Facial, Geometria Diferencial.

ABSTRACT

FIGUR, A. **Extraction of Robust Features from NURBS surfaces**. 2022. 67 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2022.

Robust features are points and curves which encode important geometric properties of a surface. Regarding human faces, the nose ridge is an already known robust feature and is used in facial recognition methods. We expect other robust features to advance and improve tools and methods for human faces identification and reconstruction. The aim of this project is to implement an extraction algorithm of ridge curves and umbilic points on NURBS surfaces. We expect this work opens the way for us to have a better understanding of the geometry of human faces in the future.

Keywords: Robust Features, Ridge Curve, surfaces parameterized by B-Splines, Facial Recognition, Differential Geometry.

SUMÁRIO

1	INTRODUÇÃO	17
1.1	Organização	21
2	CONCEITOS TEÓRICOS	23
2.1	Superfícies parametrizadas	23
2.2	Características robustas	28
2.3	Superfícies parametrizadas por B-Splines	31
2.3.1	<i>As derivadas parciais de uma superfície NURBS</i>	33
2.4	Métodos Numéricos para encontrar raízes	35
2.4.1	<i>O método de Newton-Raphson</i>	35
2.4.2	<i>O método Régula-Falsi generalizado</i>	35
3	PONTOS DE PARTIDA PARA O TRAÇO DE CURVAS RIDGE	37
3.1	Um algoritmo para construir e operar superfícies NURBS	37
3.2	Encontrando raízes de um sistema de equações não-lineares	38
3.3	Obtendo os coeficientes de uma superfície	39
3.4	Localizando pontos umbílicos	40
3.5	Localizando pontos de partida	41
3.5.1	<i>Pontos críticos das curvaturas principais</i>	41
3.5.2	<i>Pontos roxos das curvas ridge</i>	43
3.5.3	<i>Pontos de partida a partir de pontos umbílicos</i>	44
4	TRAÇO DE CURVAS RIDGE POR MÉTODO PREDITOR-CORRETOR	47
4.1	Resumo do algoritmo	47
4.2	Etapa de predição	48
4.2.1	<i>Calculando o gradiente da curva ridge</i>	49
4.2.2	<i>Cuidados adicionais da etapa de predição</i>	50
4.3	Etapa de correção	50
4.4	Condições de parada	51
4.4.1	<i>Etapa de predição</i>	51
4.4.2	<i>Etapa de correção</i>	51

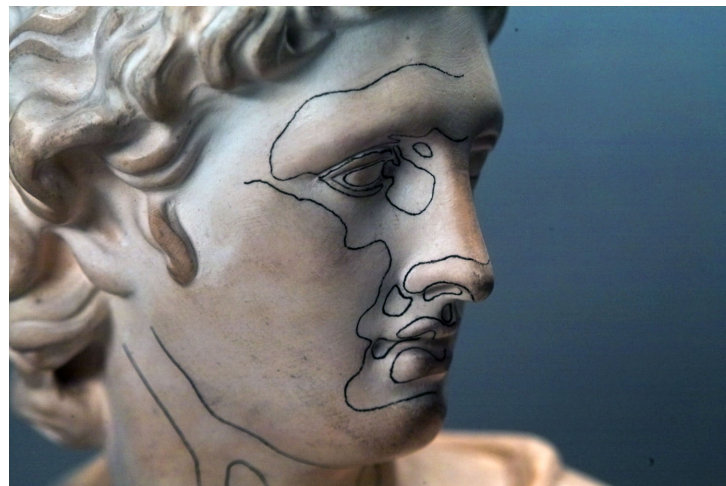
5	RESULTADOS OBTIDOS PELO ALGORITMO	53
5.1	Superfícies NURBS	53
5.1.1	<i>Superfície utilizada para validação</i>	54
5.2	Pontos de partida	54
5.2.1	<i>Pontos críticos das curvaturas principais</i>	54
5.2.2	<i>Pontos roxos das curvaturas principais</i>	55
5.2.3	<i>Pontos umbílicos</i>	56
5.3	Curvas Ridge	57
5.3.1	<i>Curvas ridges traçadas a partir dos pontos críticos</i>	57
5.3.2	<i>Curvas ridge traçadas a partir dos pontos umbílicos</i>	58
5.4	Métodos para verificação do algoritmo	59
6	CONCLUSÃO	61
6.1	Trabalhos futuros	62
6.1.1	<i>Refinamento do algoritmo ao redor de pontos umbílicos</i>	62
6.1.2	<i>Extração de outras características robustas</i>	62
6.1.3	<i>Extração de Regiões Geometricamente Salientes</i>	62
6.1.4	<i>Reconstrução de superfícies</i>	62
6.1.5	<i>Reconhecimento Facial</i>	63
	REFERÊNCIAS	65

INTRODUÇÃO

O termo “características robustas” é utilizado para denominar as características que podem ser acompanhadas quando uma superfície sofre deformações.

A primeira descrição conhecida sobre a extração de características robustas de um rosto humano foi feita por [Hilbert e Cohn-Vossen \(1990, p. 198\)](#). Eles contam que Felix Klein queria testar a hipótese de que a beleza de uma face estava ligada a algumas relações matemáticas. Klein não atingiu seu objetivo, mas no processo desenhou sobre o busto de Apolo Belvedere um tipo de característica robusta, a curva parabólica¹, como podemos ver na [Figura 1](#).

Figura 1 – Curvas desenhadas sobre o busto de Apolo Belvedere por Felix Klein.



Fonte: [Klein \(ano desconhecido\)](#).

O trabalho de [Porteous \(1994\)](#) foi pioneiro na descrição da geometria das características robustas. Até então elas haviam recebido pouquíssima atenção dos livros texto de

¹ Das características robustas conhecidas, a curva parabólica é a que possui os cálculos mais simples de serem diretamente calculados

Geometria Diferencial. Porteous destaca em [Porteous \(1994, p. 287\)](#) que a partir do final do século XX as características robustas passaram a ser de grande interesse na área de Visão Computacional.

Na literatura sobre características robustas nos deparamos com a extração de curvas ridge (um tipo de característica robusta) de bustos como o de Max Planck ou de David de Michelangelo. Essas extrações foram feitas por [Ohtake, Belyaev e Seidel \(2004\)](#) e [Cazals e Pouget \(2005\)](#). Na [Figura 2](#) é possível comparar que em ambos os bustos as curvas ridge aparecem em regiões semelhantes.

Na verdade, como descrito por [Hallinan *et al.* \(2018, p. 203-204\)](#), já se utilizam as curvas ridge que se destacam no nariz para auxiliar alguns algoritmos de reconhecimento de rostos humanos. [Hallinan *et al.* \(2018, p. 139\)](#), inclusive, classifica quais tipos de curvas ridge são mais salientes ao olho nu de um observador.

Figura 2 – Curvas ridge destacadas nos bustos.



(a) Max Planck

Fonte: [Ohtake, Belyaev e Seidel \(2004\)](#).



(b) David de Michelangelo

Fonte: [Cazals e Pouget \(2005\)](#).

Isso levanta um questionamento semelhante ao de Felix Klein. Não no que diz respeito à beleza, mas será os padrões que nós, humanos, percebemos nos rostos de outros humanos poderiam ser codificados pela geometria? Será que os rostos humanos apresentam sempre as mesmas características robustas? Seria possível reconhecer e distinguir faces apenas com as características robustas? Nesse sentido, seria possível reconstruir um rosto humano apenas com informações sobre suas características robustas?

Como já ressaltado, o livro escrito por [Hallinan *et al.* \(2018\)](#) mostra que o *nose ridge*² já é conhecido e utilizado em algoritmos de reconhecimento facial. Apesar disso, de todos os artigos que pesquisamos sobre extração das Curvas Ridge em superfícies

² Ridge do nariz, numa tradução livre

parametrizadas até o momento, principalmente os que a extraíram de rostos humanos, nenhum seguiu pelo caminho de responder as perguntas que apresentamos.

Para responder essas perguntas que apresentamos, uma etapa que precisamos realizar é extrair essas curvas dos rostos humanos. Mas como? Podemos representar rostos humanos utilizando superfícies. Ou seja, uma etapa anterior à extração dessas características robustas de rostos humanos consiste em extrair essas características robustas de superfícies. Mas quais superfícies? Utilizamos a literatura da área de extração de características robustas para tomar essa decisão. Destacamos aqui o trabalho feito por [Mari, Hetroy-Wheeler e Subsol \(2020\)](#), que fizeram um levantamento bibliográfico dos métodos conhecidos até 2019 para extração de curvas ridge de superfícies parametrizadas e, também, de superfícies formadas por malhas e nuvens de pontos.

Pela ligação direta entre superfícies parametrizadas e a teoria da Geometria Diferencial, na qual é possível definir e estudar as propriedades das características robustas, optamos por trabalhar com superfícies parametrizadas. Dentre as superfícies parametrizadas, optamos por trabalhar com superfícies NURBS, pelo abrangente referencial teórico disponível sobre sua representação computacional e por serem há décadas utilizadas em modelagem computacional, em programas de CAD.

A nossa principal referência foi um método de continuação para traço de curvas ridge implementado por [Musuvathy et al. \(2011\)](#). Esse artigo utiliza-se de métodos simbólicos para implementar o método de continuação que resulta na curva ridge. Nos baseamos nesse algoritmo para implementar um método de continuação numérica, feito em Matlab. A seguir, apresentamos uma breve descrição do método realizado por [Musuvathy et al. \(2011\)](#).

A ideia principal do algoritmo para traçar curvas ridges do artigo de [Musuvathy et al. \(2011\)](#) é utilizar as direções principais e o plano tangente em um ponto na superfície.

O algoritmo de continuação apresentado no artigo necessita de pontos de partida. A partir deles, a estratégia de busca dos pontos da curva se aproveita do fato de que os ridges de uma curvatura principal específica intersectam sua direção principal correspondente transversalmente, exceto nos pontos de virada ([Definição 20](#)), que são pontos isolados nos ridges. O artigo garante que a sequência de etapas realizada converge para um ponto ridge. Veja a seguir a estrutura do algoritmo de [Musuvathy et al. \(2011\)](#):

1. **Pontos de Partida**³: obtém-se como pontos de partida para o traço das curvas os pontos críticos das curvaturas principais e pontos umbílicos da superfície. Utiliza-se um método de subdivisão do domínio para obter os pontos com o nível de precisão desejado.

³ *Seed points*, no artigo.

2. **Avançar**⁴: partindo de um ponto ridge azul ou vermelho, obtém-se um ponto no plano tangente na direção principal oposta à desse ponto ridge. Esse ponto no plano tangente é, então, projetado na superfície.
3. **Esgueirar**⁵: Partindo de um ponto na superfície, obtém-se um ponto no plano tangente na mesma direção principal da curva ridge a ser buscada. Esse ponto no plano tangente é, então, projetado na superfície. Esta etapa é iterada até se encontrar um ponto na superfície que satisfaça a condição de ridge (Equações 2.17 e 2.18).

O algoritmo de traço das curvas consiste em executar os dois métodos apresentados iterativamente até atingir uma condição de parada retorna uma sequência de pontos que forma uma aproximação polinomial da curva ridge a qual o ponto de partida pertence. Mesmo sendo um artigo publicado em 2011, ainda é um trabalho que se encontra no estado da arte de extração de ridges, como mostra o levantamento bibliográfico realizado por [Mari, Hetroy-Wheeler e Subsol \(2020\)](#).

Nos utilizamos desses mesmos conceitos apresentados por [Musuvathy et al. \(2011\)](#) para construir nosso algoritmo. Detalhamos esses conceitos no [Capítulo 2](#).

Vale ressaltar aqui que a extração de curvas ridges em superfícies não é uma tarefa computacionalmente simples. As equações que definem essas curvas dependem das derivadas de terceira ordem de uma superfície, como pode ser visto na [Definição 18](#) da [Seção 2.2](#). O trabalho realizado por [Cazals et al. \(2006\)](#) aponta as dificuldades de se calcular ridges em torno dos pontos umbílicos, pontos roxos e dos pontos de virada. O artigo apresenta o primeiro trabalho que se tem notícia capaz de extrair as curvas ridge com precisão em torno desses pontos importantes. Isso é feito utilizando ferramentas de cálculos simbólicos, como o *Maple*, e é um trabalho importante até os dias atuais. Utilizamos esse trabalho para validação dos resultados dessa dissertação.

Escolhemos trabalhar com superfícies parametrizadas, pois nos permite trabalhar com superfícies NURBS. A teoria das curvas NURBS é bastante antiga na história da computação e até hoje é padrão em diversos softwares de CAD⁶. É também o mesmo tipo de superfície utilizada no trabalho de [Musuvathy et al. \(2011\)](#). É possível encontrar na literatura uma grande quantidade de métodos para trabalhar com elas, como podemos ver nos trabalhos de [Piegl e Tiller \(1997\)](#) e [Rogers \(2000\)](#). Essas referências foram utilizadas para construir as funções capazes de definir e operar sobre as superfícies que extraímos as curvas ridge, como apresentamos na [Seção 3.1](#).

As três etapas do método de continuação desenvolvido por [Musuvathy et al. \(2011\)](#) que enumeramos acima são muito parecidas com a dos métodos de continuação numérica

⁴ *Advance step*, no artigo.

⁵ *Slide step*, no artigo.

⁶ *Computer Aided Design*.

do tipo preditor-corretor, já bem estabelecidos na literatura. Com base na teoria apresentada por [Allgower e Georg \(2003\)](#), é possível notar que com uma superfícies NURBS de grau suficientemente grande (≥ 5), as curvas ridge dessa superfície satisfazem as condições necessárias para alguns dos métodos preditor-corretor do livro.

Nesta dissertação, apresentamos uma implementação do método preditor-corretor de continuação numérica baseado no Método de Euler-Newton com Adaptação do Tamanho do Passo via Estimativas Assintóticas apresentado por [Allgower e Georg \(2003\)](#). A estrutura do nosso algoritmo é muito parecida com a estrutura do algoritmo de [Musuvathy et al. \(2011\)](#) que apresentamos: no [Capítulo 3](#) apresentamos como obtemos pontos de partida para, no [Capítulo 4](#), implementar o método preditor-corretor citado.

Todas as etapas necessárias para a construção do algoritmo de extração de ridges de superfícies NURBS foram desenvolvidas utilizando métodos numéricos em MATLAB com a versão R2015a do programa/linguagem.

1.1 Organização

As perguntas que apresentamos são bastante ambiciosas. Vinculado ao Projeto Temático da FAPESP “Teoria de Singularidades e aplicações à geometria diferencial e visão computacional” (processo N^o 2019/07316-0), este trabalho não tem a intenção de responder a todas elas, mas pretende abrir o caminho para isso. A base teórica, desenvolvimento, resultados e discussões estão distribuídas nesta dissertação da seguinte forma:

Capítulo 2 Apresentamos uma fundamentação sobre curvas e superfícies na linguagem da Geometria Diferencial para, no mesmo capítulo, definir características robustas. Também apresentamos os conceitos numéricos necessários para o desenvolvimento do trabalho, como superfícies NURBS e métodos para solução de equações.

Capítulo 3 Este capítulo apresenta a primeira parte do algoritmo desenvolvido durante este projeto de mestrado. Nele estão os cálculos necessários e a descrição do algoritmo para obter pontos umbílicos (que são características robustas) de uma superfície e pontos de partida para traçar curvas ridge (outra característica robusta).

Capítulo 4 Este capítulo apresenta a segunda parte do algoritmo desenvolvido durante este projeto de mestrado. Nele estão os cálculos necessários e a descrição de um algoritmo preditor-corretor utilizado para obter o traço de curvas ridge numa superfície NURBS.

Capítulo 5

Neste capítulo apresentamos os resultados obtidos a partir do algoritmo apresentados nos capítulos 3 e 4. Também apresentamos o trabalho que desenvolvemos para validar os resultados do algoritmo.

Capítulo 6

Apresentamos as limitações da nossa abordagem, bem como os possíveis caminhos que outros trabalhos poderão seguir a partir dela.

CONCEITOS TEÓRICOS

Neste Capítulo apresentamos os conceitos teóricos necessários para fundamentar os algoritmos e resultados apresentados nesta dissertação. Outra finalidade deste capítulo é estabelecer as notações utilizadas ao longo do presente texto.

Na [Seção 2.1](#), apresentamos alguns conceitos da Geometria Diferencial sobre superfícies parametrizadas. Estabelecemos apenas as notações e resultados necessários para as ferramentas que serão utilizadas neste trabalho. O autor [do Carmo \(2016\)](#) aprofunda a teoria de Geometria Diferencial apresentada aqui.

Na [Seção 2.2](#) apresentamos a definição de ridges e de outras características robustas. Os conceitos são apresentados sob o ponto de vista da Geometria Diferencial, mas as características robustas são também estudadas utilizando a Teoria de Singularidades. Os autores [Izumiya et al. \(2016\)](#) aprofundam esse assunto.

Na [Seção 2.3](#) apresentamos a definição de superfícies NURBS (acrônimo para *Non Uniform Rational B-Splines*) e damos algumas de suas particularidades úteis a este trabalho. Um aprofundamento da teoria apresentada aqui foi elaborado pelos autores [Piegl e Tiller \(1997\)](#) e [Rogers \(2000\)](#).

Na [Seção 2.4](#) apresentamos dois métodos numéricos utilizados para encontrar soluções de aplicações lineares. Cada um dos métodos apresenta condições iniciais diferentes, que terão aplicações específicas ao longo deste trabalho.

2.1 Superfícies parametrizadas

As definições apresentadas nesta seção têm o fim de permitir o cálculo das curvaturas principais de uma superfície parametrizada. Elas são importantes para definir e calcular diversas curvas robustas.

Definição 1 (Aplicação diferenciável). Seja $\Phi : U \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ uma aplicação dada por:

$$\Phi(x_1, \dots, x_n) = (f_1(x_1, \dots, x_n), \dots, f_m(x_1, \dots, x_n))^T,$$

onde cada f_i é uma função de $U \subset \mathbb{R}^n$ em \mathbb{R} . Dizemos que Φ é diferenciável no ponto $q \in U$ se as derivadas parciais de cada f_i existem e são contínuas no ponto q . Dizemos que a aplicação Φ é diferenciável se ela for diferenciável em todo ponto de U . A função Φ é dita de classe C^n se todas as derivadas parciais de ordem n de cada f_i existem e são contínuas.

Definição 2 (Derivada num ponto). Seja $\Phi : U \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ uma aplicação diferenciável e $q \in U$. Definimos a derivada de Φ no ponto q como a matriz $m \times n$

$$d\Phi_q = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(q) & \dots & \frac{\partial f_1}{\partial x_n}(q) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1}(q) & \dots & \frac{\partial f_m}{\partial x_n}(q) \end{bmatrix}. \quad (2.1)$$

Essa matriz está expressa nas bases canônicas de \mathbb{R}^n e \mathbb{R}^m . Ela também é chamada de Matriz Jacobiana.

Definição 3 (Superfície regular). Um subconjunto $S \subset \mathbb{R}^3$ é uma superfície regular se, para cada ponto $p \in S$ existe uma vizinhança $V \subset \mathbb{R}^3$ e uma aplicação sobrejetora $\Phi : U \subset \mathbb{R}^2 \rightarrow V \cap S$ (U é aberto) que satisfaça as seguintes condições:

- i) A aplicação Φ é diferenciável;
- ii) A aplicação Φ é um homeomorfismo, ou seja, possui inversa Φ^{-1} contínua;
- iii) (Condição de regularidade) Para todo ponto $q \in U$ a derivada $d\Phi_q : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ é injetora, ou seja, as colunas da Matriz Jacobiana $d\Phi_q$ são linearmente independentes.

A aplicação Φ é chamada de parametrização local de S .

Definição 4. (Curva regular em uma superfície) Seja S uma superfície e $\Phi : U \subset \mathbb{R}^2 \rightarrow S \subset \mathbb{R}^3$ uma parametrização local de S . Seja $\alpha : I \subset \mathbb{R} \rightarrow U \subset \mathbb{R}^2$ uma curva regular (ou seja, α é diferenciável e $\alpha'(t) \neq 0$ para todo $t \in I$). Dizemos que a curva $\beta : I \subset \mathbb{R} \rightarrow S \subset \mathbb{R}^3$ dada por $\beta(t) = \Phi \circ \alpha(t)$ é uma curva regular na superfície $\Phi(U) \subset S$.

Definição 5 (Derivada de uma curva na Superfície). Seja $\beta : I \subset \mathbb{R} \rightarrow S \subset \mathbb{R}^3$ uma curva regular da forma $\beta(t) = \Phi \circ \alpha(t)$ na superfície S , parametrizada localmente por $\Phi : U \subset \mathbb{R}^2 \rightarrow S \subset \mathbb{R}^3$. Então β é diferenciável e sua derivada em $t \in I$ é dada por:

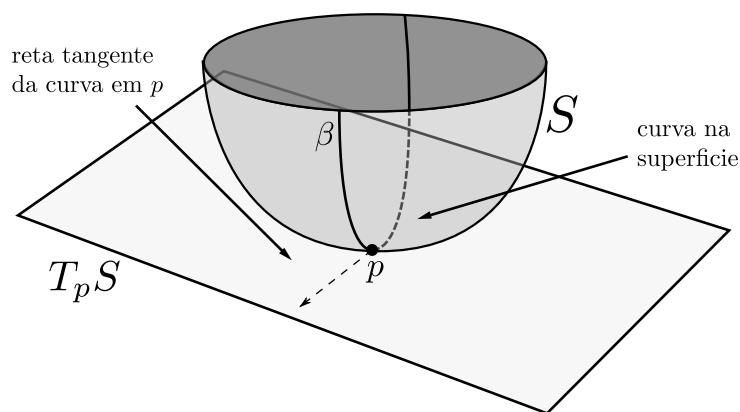
$$\beta'(t) = d\Phi(\alpha(t)) \alpha'(t). \quad (2.2)$$

Chamamos esse vetor $\beta'(t) \in \mathbb{R}^3$ de vetor tangente à curva β no ponto t .

Definição 6 (Plano tangente). Seja S uma superfície parametrizada localmente por $\Phi : U \subset \mathbb{R}^2 \rightarrow S \subset \mathbb{R}^3$ e seja $p = \Phi(q)$ um ponto em S . Definimos o plano tangente de S no ponto p como o subespaço linear $T_p S \subset \mathbb{R}^3$ gerado pelos vetores $\frac{\partial \Phi}{\partial x}(q)$ e $\frac{\partial \Phi}{\partial y}(q)$. Equivalentemente: $T_p S = d\Phi_q(\mathbb{R}^2)$.

A Figura 3 dá uma ilustração de uma curva regular β em uma superfície S . Note que a derivada da curva no ponto p pertence ao plano tangente da Superfície nesse mesmo ponto.

Figura 3 – Uma curva numa superfície e a reta tangente em um de seus pontos. Representamos $T_p S$ como um plano afim que contém o ponto p .



Definição 7 (Curvas transversais). Duas curvas regulares em uma superfície S se intersectam transversalmente em um ponto $p \in S$ (ou são transversais em p) se seus vetores tangentes em p não são paralelos. Equivalentemente, se seus vetores tangentes geram o plano tangente $T_p S$.

Definição 8 (Primeira Forma Fundamental). Seja S uma superfície parametrizada localmente por $\Phi : U \subset \mathbb{R}^2 \rightarrow S \subset \mathbb{R}^3$. No plano tangente a um ponto $p = \Phi(q)$ podemos induzir um produto interno a partir do produto interno canônico de \mathbb{R}^3 . Esse produto interno induzido corresponde a uma forma quadrática $I_p : T_p S \rightarrow \mathbb{R}$ dada por $I_p(u) = \langle u, u \rangle_p$. Essa forma quadrática é chamada de Primeira Forma Fundamental. As funções:

$$E = \left\langle \frac{\partial \Phi}{\partial x}, \frac{\partial \Phi}{\partial x} \right\rangle, \quad F = \left\langle \frac{\partial \Phi}{\partial x}, \frac{\partial \Phi}{\partial y} \right\rangle, \quad G = \left\langle \frac{\partial \Phi}{\partial y}, \frac{\partial \Phi}{\partial y} \right\rangle.$$

são chamadas coeficientes da primeira forma fundamental. Temos $I_p(u) = v^T I_p v$, onde $u = d\Phi_q(v)$ e

$$I_p = \begin{bmatrix} E & F \\ F & G \end{bmatrix}. \quad (2.3)$$

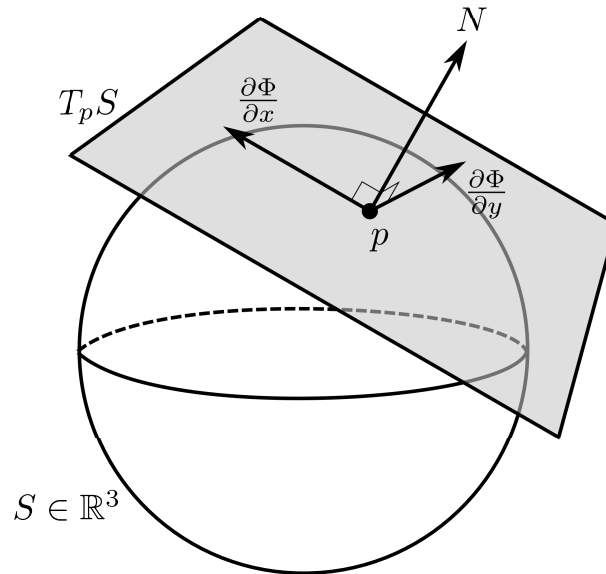
É possível definir um vetor normal à superfície em qualquer ponto dela a partir dessa definição do plano tangente. Uma ilustração dessa construção pode ser vista na Figura 4. A definição formal desse vetor normal pode ser vista a seguir:

Definição 9 (Aplicação de Gauss). Seja S uma superfície parametrizada localmente por $\Phi : U \subset \mathbb{R}^2 \rightarrow S \subset \mathbb{R}^3$. Definimos um vetor normal unitário no ponto $p = \Phi(q)$ pelo produto vetorial normalizado das derivadas parciais de Φ :

$$N(p) = \frac{\frac{\partial \Phi}{\partial x} \times \frac{\partial \Phi}{\partial y}}{\left\| \frac{\partial \Phi}{\partial x} \times \frac{\partial \Phi}{\partial y} \right\|}(q). \quad (2.4)$$

A aplicação N é chamada aplicação de Gauss.

Figura 4 – Um ponto numa esfera com seu respectivo plano tangente e vetor normal.



Definição 10 (Segunda Forma Fundamental). Seja S uma superfície parametrizada localmente por $\Phi : U \subset \mathbb{R}^2 \rightarrow S \subset \mathbb{R}^3$ de classe C^2 . Definimos a Segunda Forma Fundamental pela forma quadrática $\Pi_p : T_p S \rightarrow \mathbb{R}$ dada por $\Pi_p(u) = \langle u, dN_p(u) \rangle$. As funções:

$$e = \left\langle N, \frac{\partial^2 \Phi}{\partial x^2} \right\rangle, \quad f = \left\langle N, \frac{\partial^2 \Phi}{\partial x \partial y} \right\rangle, \quad g = \left\langle N, \frac{\partial^2 \Phi}{\partial y^2} \right\rangle.$$

são chamadas coeficientes da segunda forma fundamental. Temos $\Pi_p(u) = v^T \Pi_p v$, onde $u = d\Phi_q(v)$ e

$$\Pi_p = \begin{bmatrix} e & f \\ f & g \end{bmatrix}. \quad (2.5)$$

A Primeira Forma Fundamental é utilizada para construir uma métrica sobre a superfície S . Já a Segunda Forma Fundamental indica a “taxa de variação” dos planos tangentes em cada ponto da superfície. Essas intuições são úteis para entender os conceitos que definiremos a seguir.

Definição 11 (Aplicação de Weingarten). Seja S uma superfície de classe C^2 parametrizada localmente por $\Phi : U \subset \mathbb{R}^2 \rightarrow S \subset \mathbb{R}^3$. A Aplicação de Gauss $N : S \rightarrow \mathbb{S}^2$ é

diferenciável. Sua derivada num ponto $p = \Phi(q)$ é a função $dN_p : T_pS \rightarrow T_pS$ que pode ser expressa por uma multiplicação de matrizes utilizando a Primeira e a Segunda Forma Fundamental:

$$dN_p = -I_p^{-1}\Pi_p = -W \quad (2.6)$$

A aplicação W é conhecida como Aplicação de Weingarten e pode ser representada pelos Coeficientes de Weingarten derivados da Primeira e Segunda Forma Fundamental:

$$w_{1,1} = \frac{eG - fF}{EG - F^2}, \quad w_{1,2} = \frac{fG - gG}{EG - F^2}, \quad w_{2,1} = \frac{fE - eF}{EG - F^2}, \quad w_{2,2} = \frac{gE - fF}{EG - F^2}.$$

Para $u = d\Phi_q(v)$, temos $dN_p(d\Phi_q(v)) = -W(v)$, onde

$$W = \begin{bmatrix} w_{1,1} & w_{1,2} \\ w_{2,1} & w_{2,2} \end{bmatrix}. \quad (2.7)$$

Definição 12 (Curvaturas e direções principais). Seja S uma superfície parametrizada localmente por $\Phi : U \subset \mathbb{R}^2 \rightarrow S \subset \mathbb{R}^3$. Denotamos por d_1 e d_2 os autovetores da aplicação de Weingarten, que são chamados de direções principais. Seus autovalores associados κ_1 e κ_2 são chamados de curvaturas principais da superfície S no ponto p . Estas últimas podem ser calculadas através dos coeficientes da primeira e segunda forma fundamental da seguinte maneira:

$$\kappa_1 = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \quad \text{e} \quad \kappa_2 = \frac{-B - \sqrt{B^2 - 4AC}}{2A}, \quad \text{onde:} \quad (2.8)$$

$$A = EG - F^2, \quad B = 2fF - eG - gE, \quad C = eg - f^2.$$

Temos que $\kappa_1 \geq \kappa_2$ e, a partir disso, podemos obter as direções principais da seguinte maneira:

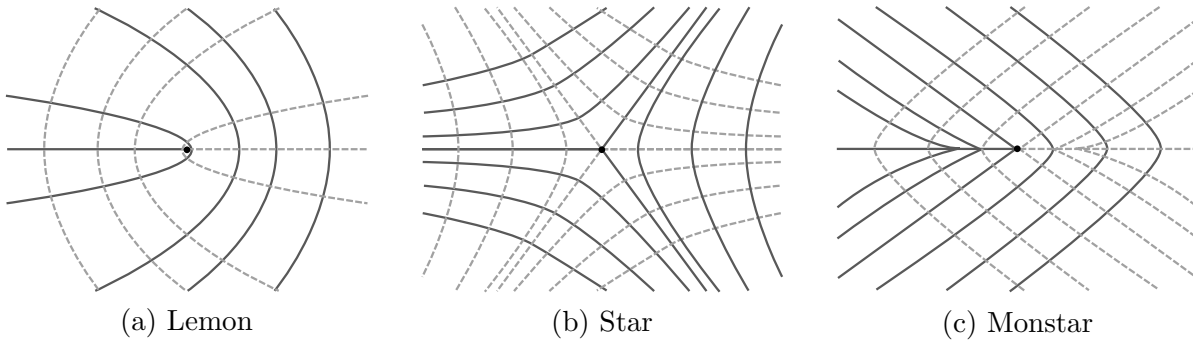
$$d_1 = \begin{bmatrix} \kappa_1 F - f \\ e - \kappa_1 E \end{bmatrix} \quad \text{e} \quad d_2 = \begin{bmatrix} \kappa_2 F - f \\ e - \kappa_2 E \end{bmatrix}. \quad (2.9)$$

Observação 1. É importante ressaltar que as curvaturas principais são funções $\kappa_i : U \subset S \rightarrow \mathbb{R}$, com $i = 1, 2$.

Definição 13 (Ponto umbílico). Seja S uma superfície regular parametrizada localmente por $\Phi : U \subset \mathbb{R}^2 \rightarrow S \subset \mathbb{R}^3$. Um ponto $p = \Phi(q)$ é chamado ponto umbílico se $\kappa_1(q) = \kappa_2(q)$.

Uma curva em S tangente em todos os seus pontos a uma direção principal é chamada de **linha de curvatura** de S . Em cada ponto não umbílico da superfície passam duas linhas de curvatura ortogonais. Como descrito por [Sotomayor e Gutierrez \(1982\)](#) e [Bruce e Fidal \(1989\)](#), nos pontos umbílicos as linhas de curvatura de comportam genericamente de três maneiras distintas, como indicado na [Figura 5](#).

Figura 5 – Comportamento genérico das linhas de curvatura associadas (linhas contínuas e pontilhadas) no domínio de uma superfície na vizinhança de um ponto umbílico (Sotomayor e Gutierrez (1982), BRUCE (1984), Bruce e Fidal (1989)).



Definição 14 (Curvatura gaussiana). Seja S uma superfície parametrizada localmente por $\Phi : U \subset \mathbb{R}^2 \rightarrow S \subset \mathbb{R}^3$. O determinante da derivada da aplicação de Gauss dN_p é chamado de curvatura gaussiana da superfície S no ponto $p = \Phi(q)$. Podemos expressá-la em termos das curvaturas principais como:

$$K(q) = \kappa_1(q)\kappa_2(q), \quad (2.10)$$

e em termos dos coeficientes de Weingarten como:

$$K(q) = \frac{eg - f^2}{EG - F^2}(q). \quad (2.11)$$

Definição 15 (Curvatura média). Seja S uma superfície parametrizada por $\Phi : U \subset \mathbb{R}^2 \rightarrow S \subset \mathbb{R}^3$. A metade do traço da aplicação de Weingarten é chamado de curvatura média da superfície S no ponto $p = \Phi(q)$. Podemos expressá-la em termos das curvaturas principais como:

$$H(q) = \frac{\kappa_1(q) + \kappa_2(q)}{2}, \quad (2.12)$$

e em termos dos coeficientes de Weingarten como:

$$H(q) = \frac{1}{2} \frac{eG - 2fF + gE}{EG - F^2}(q). \quad (2.13)$$

2.2 Características robustas

Nesta seção utilizaremos as definições dos vários tipos de curvaturas que apresentamos na Seção 2.1 para definir algumas características robustas.

O trabalho de Porteous (1994) destaca as características robustas¹ como características de uma superfície que podem ser acompanhadas quando deformadas. Porteous iniciou esse trabalho, mas diversos outros autores fizeram trabalhos fundamentais sobre esse assunto. Apresentaremos algumas características robustas a seguir.

¹ *Robust features.*

Como fixado na [Definição 12](#), em todo ponto não umbílico de uma superfície denotamos as curvaturas principais como κ_1 e κ_2 , escolhidas de modo que $\kappa_1 \geq \kappa_2$.

Definição 16 (Curva parabólica). Seja S uma superfície de classe C^2 parametrizada localmente por $\Phi : U \subset \mathbb{R}^2 \rightarrow S \subset \mathbb{R}^3$. O conjunto parabólico P de S é o conjunto de todos os pontos $p = \Phi(q)$ em que a curvatura gaussiana é $K(q) = 0$. Quando P é uma curva, a chamamos de curva parabólica. Mais especificamente², para $q = (u, v)$, o ponto pertence à curva parabólica se, e somente se

$$\kappa_1(u, v) = 0 \quad \text{ou} \quad \kappa_2(u, v) = 0, \quad (2.14)$$

equivalentemente,

$$(eg - f^2)(u, v) = 0. \quad (2.15)$$

Definição 17 (Curva sub-parabólica). Seja S uma superfície de classe C^3 parametrizada localmente por $\Phi : U \subset \mathbb{R}^2 \rightarrow S \subset \mathbb{R}^3$. Um ponto $p = \Phi(q)$ não umbílico é chamado ponto sub-parabólico se

$$\langle \nabla \kappa_2, d_1 \rangle = \nabla_{d_1} \kappa_2 = 0 \quad \text{ou} \quad \langle \nabla \kappa_1, d_2 \rangle = \nabla_{d_2} \kappa_1 = 0 \quad (2.16)$$

em $q = (u, v)$. O conjunto de todos os pontos sub-parabólicos de S é chamado de curva sub-parabólica.

Definição 18 (Curva ridge). Seja S uma superfície de classe C^3 parametrizada localmente por $\Phi : U \subset \mathbb{R}^2 \rightarrow S \subset \mathbb{R}^3$. Um ponto $p = \Phi(q)$ não umbílico é chamado:

- um ponto ridge azul (ou apenas ponto azul) se a derivada direcional da curvatura principal κ_1 na sua respectiva direção principal d_1 é nula, ou seja, em $q = (u, v)$ temos que

$$\langle \nabla \kappa_1, d_1 \rangle = \nabla_{d_1} \kappa_1 = 0. \quad (2.17)$$

- um ponto ridge vermelho (ou apenas ponto vermelho) se a derivada direcional da curvatura principal κ_2 na sua respectiva direção principal d_2 é nula, ou seja, em $q = (u, v)$ temos que

$$\langle \nabla \kappa_2, d_2 \rangle = \nabla_{d_2} \kappa_2 = 0. \quad (2.18)$$

O conjunto formado por todos os pontos ridges de uma superfície é chamado de curva ridge ou simplesmente de ridges. O conjunto de todos os ridges azuis forma um ridge maximal (κ_1 -ridge) e o conjunto de todos os ridges vermelhos forma um ridge minimal (κ_2 -ridge).

² pela [Equação 2.10](#).

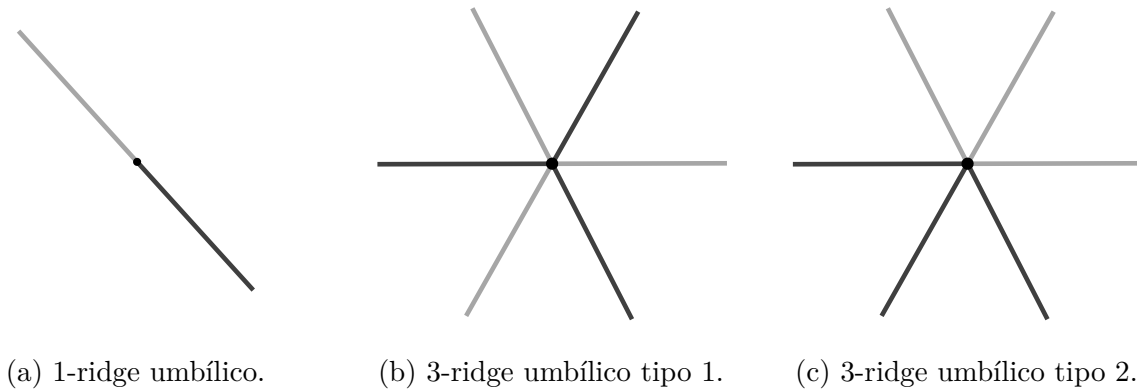
Os ridges são os pontos onde uma das curvaturas principais atinge um ponto crítico em sua respectiva direção principal. Os pontos sub-parabólicos são os pontos onde uma das curvaturas principais atinge um ponto crítico em relação à outra direção principal. Como os ridges são os protagonistas deste trabalho, vamos detalhar um pouco mais da teoria por trás dessas curvas.

O comportamento dos ridges numa vizinhança de pontos umbílicos genéricos foi estudado por BRUCE (1984). Temos três tipos genéricos de ridges (ilustrados na Figura 6):

1-ridge umbílico: é um ponto umbílico onde chega apenas 1 ridge maximal e 1 ridge minimal. Ver Figura 6a.

3-ridge umbílico: é um ponto umbílico onde chegam 3 ridges maximais e 3 ridges minimais. Temos duas configurações, como indicado na Figura 6b e na Figura 6c.

Figura 6 – Comportamento dos ridges nas proximidades de pontos umbílicos. Nas figuras (a), (b) e (c) os ridges mudam de cor ao passar pelo ponto umbílico (preto).



(a) 1-ridge umbílico.

(b) 3-ridge umbílico tipo 1.

(c) 3-ridge umbílico tipo 2.

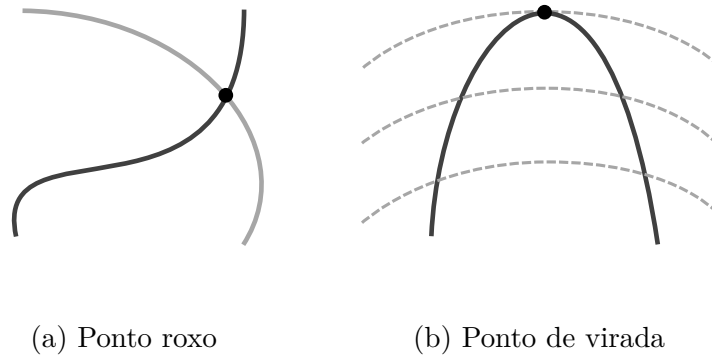
Existem pontos não umbílicos que também são importantes para entender o comportamento dos ridges. Um desses pontos que merece destaque é quando duas curvas ridge diferentes se intersectam.

Definição 19 (Ponto roxo). Uma curva ridge maximal (formada por pontos azuis, um κ_1 -ridge) pode interceptar uma curva ridge minimal (formada por pontos vermelhos, um κ_2 -ridge) em um ponto não umbílico, chamado de ponto roxo.

Outro ponto não umbílico que merece destaque diz respeito ao comportamento dos ridges quando cruzam suas respectivas linhas de curvatura. Exceto em alguns pontos isolados, os ridges intersectam transversalmente sua linha de curvatura correspondente. Os pontos onde isso não acontece são chamados pontos de virada. Uma ilustração desses pontos pode ser vista na Figura 7.

Definição 20 (Ponto de virada). Quando um ridge intersecta tangencialmente a sua linha de curvatura correspondente esse ponto de intersecção recebe o nome de ponto de virada.

Figura 7 – Na figura (a) há duas curvas ridge maximal e minimal se intersectando num ponto roxo (preto). Na figura (b) há uma curva ridge (escuro) se intersectando com algumas das suas linhas de curvatura correspondentes (tracejado) e o ponto de virada (preto).



Os ridges maximais e minimais não umbílicos podem ser classificados³ nos seguintes subtipos:

- Elíptico:** Um ridge maximal ($\nabla_{d_1} \kappa_1 = 0$) é chamado ridge elíptico quando⁴ $d_1^T H_{\kappa_1} d_1 < 0$; Análogamente um ridge minimal ($\nabla_{d_2} \kappa_2 = 0$) é um ridge elíptico quando $d_2^T H_{\kappa_2} d_2 > 0$.
- Hiperbólico:** Um ridge maximal ($\nabla_{d_1} \kappa_1 = 0$) é chamado ridge hiperbólico quando $d_1^T H_{\kappa_1} d_1 > 0$; Análogamente um ridge minimal ($\nabla_{d_2} \kappa_2 = 0$) é um ridge hiperbólico quando $d_2^T H_{\kappa_2} d_2 < 0$.
- Crest:** Um ridge maximal elíptico é chamado crest quando $|\kappa_1| > |\kappa_2|$; Um ridge minimal elíptico é chamado de crest (ou κ_2 -crest) quando $|\kappa_1| < |\kappa_2|$.

2.3 Superfícies parametrizadas por B-Splines

Todas as definições apresentadas na [Seção 2.1](#) tornam-se mais fáceis de serem computacionalmente calculadas se trabalharmos apenas com polinômios. As definições desta seção são dedicadas a permitir essa tarefa.

³ Essa é a classificação utilizada pelos autores [Musuvathy et al. \(2011\)](#).

⁴ A matriz H_{κ_i} que aparece nessas definições é a matriz hessiana da função κ_i das curvaturas principais.

Definição 21 (Espaço projetivo). Definimos o espaço projetivo \mathbb{P}^3 como o quociente $\mathbb{R}^4/\{0\}$ obtido pela relação de equivalência: $x \sim y$ se, e somente se, existe $\lambda \in \mathbb{R}$ tal que $x = \lambda y$. Se (x, y, z, w) é um representante da classe de equivalência de um ponto $p \in \mathbb{P}^3$, dizemos que $(x : y : z : w)$ são as coordenadas homogêneas de p .

Definição 22 (Aplicação perspectiva). Seja \mathbb{P}^3 o espaço projetivo 3 dimensional. A aplicação perspectiva $H^3 : \mathbb{P}^3 \rightarrow \mathbb{R}^3$ é definida como:

$$H^3(x : y : z : w) = \begin{cases} \left(\frac{x}{w}, \frac{y}{w}, \frac{z}{w}\right) & \text{se } w \neq 0 \\ (x, y, z) & \text{se } w = 0 \end{cases} \quad (2.19)$$

Essa aplicação é essencial para definir uma superfície parametrizada por funções racionais. Essas superfícies podem ser definidas utilizando um tipo específico de polinômios como base, a serem definidos adiante. Um exemplo de tal superfície pode ser visto na [Figura 9](#).

Definição 23 (Vetor de nós). Seja $(U)_m = (u_0, u_1, \dots, u_m)$ uma seqüência não decrescente de números reais, isto é, $u_i \leq u_{i+1}$ para todo $i \in \{0, \dots, m-1\}$. Cada um dos elementos u_i é chamado de nó e $(U)_m$ é chamado vetor de nós.

Definição 24 (B-Spline). Considere um vetor de nós $(U)_m = (u_0, \dots, u_m)$ e um número natural $k < m$. Dado um nó u_i , com $i \in \{0, \dots, m-k-1\}$, a i -ésima função B-Spline de grau k é definida recursivamente como $N_{i,k} : \mathbb{R} \rightarrow \mathbb{R}$ tal que:

$$N_{i,0}(u) = \begin{cases} 1 & \text{se } u_i \leq u < u_{i+1} \\ 0 & \text{caso contrário} \end{cases} \quad (2.20)$$

$$N_{i,k}(u) = \frac{u - u_i}{u_{i+k} - u_i} N_{i,k-1}(u) + \frac{u_{i+k+1} - u}{u_{i+k+1} - u_{i+1}} N_{i+1,k-1}(u) \quad (2.21)$$

A [Figura 8](#) ilustra algumas funções B-Spline.

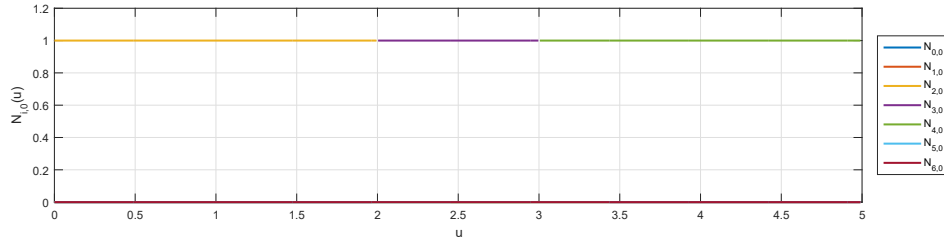
Definição 25 (Superfície NURBS). Definimos uma Superfície NURBS⁵ de grau $k \times \ell$ como uma aplicação $S : [u_0, u_m] \times [v_0, v_\ell] \rightarrow \mathbb{R}^3$ dada pela composição $H^3 \circ S_w$, onde $S_w : [u_0, u_m] \times [v_0, v_\ell] \rightarrow \mathbb{R}^4$ é da forma:

$$S_w(u, v) = \sum_{i=0}^{m-k-1} \sum_{j=0}^{n-\ell-1} N_{i,k}(u) N_{j,\ell}(v) P_{i,j}^w, \quad \text{onde} \quad (2.22)$$

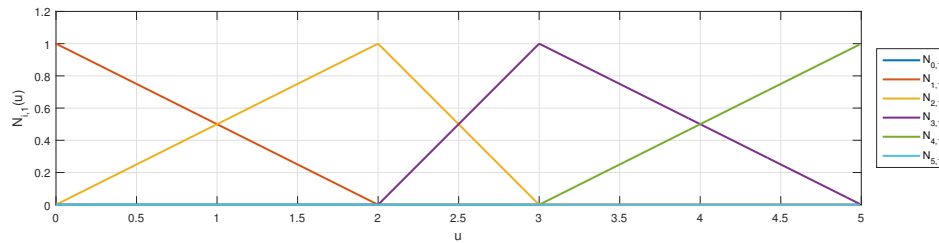
cada um dos polinômios $N_{i,k}$ e $N_{j,\ell}$ são B-Splines de grau k e ℓ e definidos pelos seguintes vetores de nós, respectivamente:

$$U = \underbrace{\{u_0, \dots, u_k\}}_{u_0 = \dots = u_k} \dots \underbrace{\{u_{m-k}, \dots, u_m\}}_{u_{m-k} = \dots = u_m} \quad \text{e} \quad V = \underbrace{\{v_0, \dots, v_\ell\}}_{v_0 = \dots = v_\ell} \dots \underbrace{\{v_{n-\ell}, \dots, v_n\}}_{v_{n-\ell} = \dots = v_n}.$$

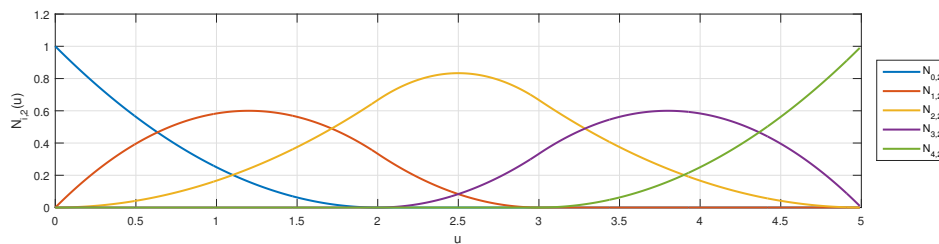
⁵ Acrônimo para *Non Uniform Rational B-Splines*. Numa tradução livre: B-Splines Racionais Não Uniformes.

Figura 8 – B-Splines geradas pelo vetor de nós $(U)_7 = \{0, 0, 0, 2, 3, 5, 5, 5\}$.

(a) B-Splines de grau 0.



(b) B-Splines de grau 1.



(c) B-Splines de grau 2.

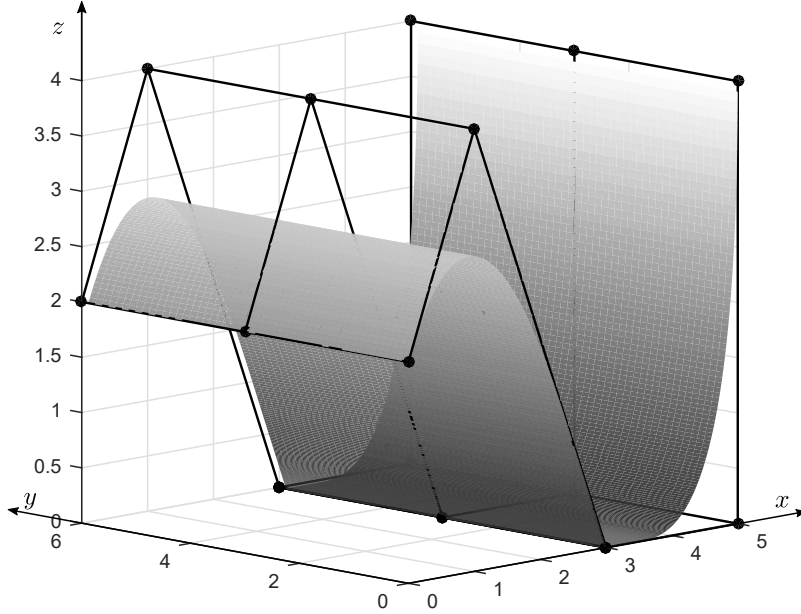
que cada $P_{i,j}^w = (w_{i,j}x : w_{i,j}y : w_{i,j}z : w_{i,j})$ é uma representação em coordenadas homogêneas de um ponto de controle $P_{i,j} = (x, y, z) \in \mathbb{R}^3$ da superfície utilizando os pesos $w_{i,j} \in \mathbb{R}$ com $i \in \{0, \dots, m - k - 1\}$ e $j \in \{0, \dots, n - \ell - 1\}$. A matriz de pontos $P_{i,j}$ forma o que chamamos de polígono de controle da superfície.

2.3.1 As derivadas parciais de uma superfície NURBS

Considere um vetor de nós $(U)_m = (u_0, \dots, u_m)$ e suas B-Splines $N_{i,k}(u)$ associadas. Podemos calcular, como mostrado por Piegl e Tiller (1997), a derivada de ordem $c \leq k$ de uma B-Spline com a fórmula:

$$N_{i,k}^{(c)}(u) = \frac{k!}{(k-c)!} \sum_{j=0}^c a_{c,j} N_{i+j,k-c}(u), \quad (2.23)$$

Figura 9 – Uma Superfície NURBS e seu polígono de controle.



onde os coeficientes $a_{c,j}$ que multiplicam as funções B-Splines $N_{i,k-c}, \dots, N_{i+c,k-c}$ são definidos como:

$$\begin{aligned}
 a_{0,0} &= 1, \\
 a_{c,0} &= \frac{a_{c-1,0}}{u_{i+k-c+1} - u_i}, \\
 a_{c,j} &= \frac{a_{c-1,j} - a_{c-1,j-1}}{u_{i+k+j-c+1} - u_{i+j}}, \text{ para } j \in \{1, \dots, c-1\}, \\
 a_{c,c} &= \frac{-a_{c-1,c-1}}{u_{i+c+1} - u_{i+c}}.
 \end{aligned} \tag{2.24}$$

As derivadas parciais de ordem $c \leq k$ e $d \leq \ell$ de uma superfície NURBS de grau $k \times \ell$ podem ser calculadas de maneira recursiva da seguinte forma:

$$\begin{aligned}
 S^{(c,d)}(u,v) &= \frac{\partial^{c+d}}{\partial^c u \partial^d v} S(u,v) = \frac{1}{w(u,v)} \left(A^{(c,d)}(u,v) - \sum_{i=1}^c \binom{c}{i} w^{(i,0)}(u,v) S^{(c-1,d)}(u,v) \right. \\
 &\quad - \sum_{j=1}^d \binom{d}{j} w^{(0,j)}(u,v) S^{(c,d-j)}(u,v) \\
 &\quad \left. - \sum_{i=1}^c \binom{c}{i} \sum_{j=1}^d \binom{d}{j} w^{(i,j)}(u,v) S^{(c-i,d-j)}(u,v) \right),
 \end{aligned}$$

onde as funções $A^{(i,j)}$ e $w^{(i,j)}$ são, respectivamente, as três primeiras coordenadas e a última coordenada da função $S_w^{(i,j)}$. Vale ressaltar que $S^{(0,0)} = S_w$.

2.4 Métodos Numéricos para encontrar raízes

Nesta seção apresentamos brevemente dois métodos utilizados para encontrar raízes de sistemas de aplicações reais (de múltiplas variáveis). Na literatura usual esses métodos costumam ser apresentados apenas em sua versão para funções reais (de uma única variável). A apresentação dos métodos neste contexto tem como pretensão facilitar o entendimento da construção dos algoritmos apresentados posteriormente.

2.4.1 O método de Newton-Raphson

Seja $\Psi : U \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$ uma aplicação diferenciável e $\hat{q} \in U$ tal que $\Psi(\hat{q}) = \mathbf{0}$ cuja derivada $d\Psi(\hat{q})$ seja invertível. O teorema da função inversa afirma que existe uma bola aberta $B \subset U$ centrada em \hat{q} tal que a restrição $\Psi|_B$ é um difeomorfismo (isto é, $d\Psi|_B$ tem inversa diferenciável). A partir disso, é possível definir o método de Newton-Raphson⁶ Generalizado⁷.

Dado um chute inicial $q^{(0)} \in B$, calcular um termo da seguinte sequência:

$$q^{(k+1)} = q^{(k)} - [d\Psi]^{-1}(q^{(k)})\Psi(q^{(k)}), \text{ com } k = 0, 1, 2, \dots, \quad (2.25)$$

consiste em executar uma iteração do método de Newton-Raphson.

Temos que

$$\lim_{k \rightarrow \infty} q^{(k)} = \hat{q}.$$

Mais detalhes para as condições de convergência desse método podem ser vistas no trabalho de [Ryaben'kii e Tsynkov \(2006\)](#). Aqui cabe destacar que, com um chute inicial $q^{(0)}$ suficientemente próximo de \hat{q} , o método de Newton-Raphson obtém com uma precisão suficientemente grande a raiz \hat{q} desejada.

A condição de parada do algoritmo pode ser baseada no erro⁸ estimado ε_{k+1} a cada iteração do algoritmo:

$$\varepsilon_{k+1} = \left\| \frac{q^{(k+1)} - q^{(k)}}{q^{(k+1)}} \right\|. \quad (2.26)$$

Quando ε_{k+1} atinge uma tolerância pré determinada (próxima de zero), o algoritmo se encerra com solução $q^{(k+1)} \simeq \hat{q}$.

2.4.2 O método Régula-Falsi generalizado

Seja $D \subset \mathbb{R}^n$ um subconjunto convexo e $\Psi : D \rightarrow \mathbb{R}^m$ uma aplicação contínua. Considere $q_0, q_1, \dots, q_n \in D$ tais que os valores $\Psi_0 = \Psi(q_0), \Psi_1 = \Psi(q_1), \dots, \Psi_n = \Psi(q_n)$ sejam os vérti-

⁶ O método de Newton-Raphson também é conhecido apenas como método de Newton.

⁷ Definição análoga para o método de Newton-Raphson para funções reais apresentada por [Quarteroni, Sacco e Saleri \(2007\)](#).

⁸ Para detalhes sobre a escolha do erro ver as discussões feitas por [Chapra \(2011\)](#) sobre erro aproximado no capítulo 3 do livro.

ces de um n -simplexo que contenha a origem o , isto é, se a solução do sistema $T\lambda = v$ seja tal que $\lambda \in \mathbb{R}_{\geq 0}^{n+1}$ (isto é, λ é um vetor cujas coordenadas são não-negativas). Considere:

$$T = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \Psi_0^T & \Psi_1^T & \dots & \Psi_n^T \end{bmatrix} \in \mathbb{R}^{(n+1) \times (n+1)} \quad \text{e} \quad v^T = \begin{bmatrix} 1 & o \end{bmatrix} \in \mathbb{R}^{n+1}. \quad (2.27)$$

Esse sistema calcula as coordenadas baricêntricas de o em relação aos vértices $\Psi_0, \Psi_1, \dots, \Psi_n$ do n -simplexo. Quando a solução λ é composta apenas por coordenadas positivas, isso significa que o pertence ao interior do n -simplexo; no caso de λ composto por coordenadas não-negativas, o pode pertencer à fronteira do n -simplexo; quando alguma coordenada de λ é negativa, temos que o está no exterior do n -simplexo.

O método Regula-Falsi generalizado consiste em, considerando que $\det(T) \neq 0$, encontrar a solução $\hat{q} \in \mathbb{R}^n$ tal que $\Psi(\hat{q}) = 0$. No método, $q_0, q_1, \dots, q_n \in D$ são considerados como as aproximações iniciais. Melhores aproximações são obtidas a cada iteração. Essas aproximações são obtidas a partir do seguinte algoritmo extraído de [Bittner \(1976\)](#)⁹:

- i) Encontre a solução $\lambda = (\lambda_0, \lambda_1, \dots, \lambda_n) \geq 0$ para a origem o em relação ao n -simplexo de vértices $\Psi(q_0), \Psi(q_1), \dots, \Psi(q_n)$.
- ii) Calcule $q = \sum_{i=0}^n \lambda_i q_i$ e $\Psi(q)$
 - Caso $\Psi(q) = 0$, temos que q é a solução \hat{q} desejada e o algoritmo se encerra aqui.
 - Caso contrário, continuamos com os procedimentos:
- iii) Calcule as coordenadas baricêntricas $\varkappa = (\varkappa_0, \varkappa_1, \dots, \varkappa_n)$ do ponto $\Psi(q)$ em relação ao n -simplexo de vértices $\Psi(q_0), \Psi(q_1), \dots, \Psi(q_n)$.
- iv) Calcule o índice σ que satisfaça a condição:

$$\frac{\lambda_\sigma}{\varkappa_\sigma} = \min \left\{ \frac{\lambda_i}{\varkappa_i} : \varkappa_i > 0 \right\}.$$

- v) Substitua $q_\sigma = q$ e mantenha $q_i = q_i$ quando $i \neq \sigma$.
- vi) Repita o algoritmo desde a etapa (i) com os novos valores de q_i .

⁹ Para mais detalhes do algoritmo e detalhes sobre sua convergência, consultar o artigo citado.

PONTOS DE PARTIDA PARA O TRAÇO DE CURVAS RIDGE

Neste capítulo começamos a descrever o algoritmo para extrair curvas ridges de superfícies NURBS. Optamos por dividir a apresentação do algoritmo em dois capítulos devido ao volume de informação e cálculos necessário em cada um. Aqui, em resumo, apresentamos como encontrar sobre uma superfície os pontos de partida necessários para poder realizar o traço de uma curva ridge.

Na [Seção 3.1](#) apresentamos brevemente a estrutura de operações com superfícies NURBS necessária para construir o algoritmo. Na [Seção 3.2](#) apresentamos o algoritmo que desenvolvemos para calcular raízes de equações não-lineares, que é utilizado para resolver os sistemas apresentados nas demais seções deste capítulo. Na [Seção 3.3](#) apresentamos as equações para calcular numericamente diversos coeficientes da superfície, necessários para qualquer etapa do algoritmo e que constituem o cerne deste trabalho. Na [Seção 3.4](#) apresentamos como obter uma das características robustas de uma superfície, a saber, os pontos umbílicos. Na [Seção 3.5](#) condensamos o que apresentamos nas seções anteriores para apresentar como calcular os pontos de partida necessários para executar o algoritmo de traço das curvas ridge, que será apresentado no [Capítulo 4](#).

3.1 Um algoritmo para construir e operar superfícies NURBS

Para extrair e exibir ridges de superfícies NURBS, é necessário realizar operações sobre tais superfícies. Para isso, foi preciso implementar algoritmos para localizar pontos sobre uma superfície NURBS, bem como obter suas características intrínsecas e extrínsecas definidas na [Seção 2.1](#).

As operações básicas para as superfícies NURBS, descritas na [Seção 2.3](#), foram implementadas utilizando como principal referências os algoritmos de [Piegl e Tiller \(1997\)](#).

As características intrínsecas e extrínsecas das superfícies parametrizadas, descritas na [Seção 2.1](#), foram obtidas através de operações utilizando esses algoritmos básicos das NURBS. Em resumo, listamos a seguir os principais métodos implementados que compõem o algoritmo para operar sobre superfícies NURBS. Aqui estamos assumindo que estamos trabalhando com uma superfície NURBS $S: \mathbb{R}^2 \rightarrow \mathbb{R}^3$ de classe C^3 a partir do seu grau, vetores de nós, pontos de controle e pesos.

1. Um método que recebe um ponto do domínio da parametrização e retorna o respectivo ponto avaliado na superfície;
2. Um método que recebe um ponto do domínio da parametrização S e retorna a derivada parcial de ordem desejada dessa parametrização naquele ponto;
3. Um método que recebe um ponto (a, b) do domínio da parametrização S e retorna o vetor normal a $S(a, b)$;
4. Um método que recebe um ponto (a, b) do domínio da parametrização S e retorna os coeficientes da primeira forma fundamental daquele ponto;
5. Um método que recebe um ponto (a, b) do domínio da parametrização S e retorna os coeficientes da segunda forma fundamental daquele ponto.

Esses métodos compõem toda a base para construirmos o algoritmo de extração dos ridges das superfícies, pois são utilizados para definir todas as funções que estamos calculando.

3.2 Encontrando raízes de um sistema de equações não-lineares

Para construir um algoritmo para encontrar as raízes de um sistema de equações não-lineares Ψ , compomos os dois métodos numéricos apresentados na [Seção 2.4](#). Isso foi feito da seguinte forma:

- i) Divide-se o domínio \mathcal{D} (limitado) em que o sistema de equações Ψ está definido em vários subconjuntos convexos D .
- ii) Subdivide-se um subconjunto convexo D em n -simplexos d_i e verificamos se algum dos n -simplexos $\Psi(d_i)$ contém a origem (utilizando coordenadas baricêntricas). Caso haja algum n -simplexo que contenha a origem, satisfazem-se as condições para o método Régula-Falsi generalizado (ver [Subseção 2.4.2](#)) e faz-se o seguinte:

- Realizam-se cinco iterações do método Régula-Falsi generalizado para obter aproximações dos zeros das equações que compõe o sistema. Testes empíricos mostraram ser suficiente executar cinco iterações do método para obter uma aproximação que satisfaça as condições do método de Newton-Raphson.
- Utiliza-se as aproximações como pontos iniciais para o método de Newton-Raphson generalizado (ver [Subseção 2.4.1](#)), que obtém o resultado dos sistemas com a precisão desejada.
- Executa-se o passo ii) para o próximo subconjunto convexo.

O método de Newton-Raphson converge rapidamente, mas depende de boas aproximações para sua convergência. O método Régula-Falsi não converge tão rápido, mas tem condições iniciais que se satisfazem com mais facilidade. Com isso em mente, a ideia desse algoritmo é utilizar o método Régula-Falsi a partir de uma subdivisão do domínio para encontrar uma boa condição inicial para o método de Newton-Raphson.

Esse algoritmo foi utilizado para encontrar as soluções de todos os sistemas de equações não-lineares das próximas seções. Cabe aqui dizer que as implementações feitas estão definidas (domínio e imagem) em \mathbb{R}^2 .

3.3 Obtendo os coeficientes de uma superfície

Como visto na [Seção 2.1](#) e na [Seção 2.2](#), encontrar as características robustas passa pelo cálculo de alguns coeficientes, como os coeficientes da primeira e da segunda forma fundamental. Com os métodos para operar superfícies NURBS apresentados na [Seção 3.1](#), somos capazes de obtê-los.

Os trabalhos de [Musuvathy *et al.* \(2011\)](#) e [Maekawa e Patrikalakis \(1994\)](#) apresentam os cálculos dos coeficientes da primeira e da segunda forma fundamental na sua forma racional. A ideia principal gira em torno de calcular os coeficientes que apresentamos na [Seção 2.1](#) a partir de $\hat{N} = \Phi_u \times \Phi_v$. Temos as seguintes funções racionais por partes:

$$\hat{N}_u = (\Phi_{uu} \times \Phi_v) + (\Phi_u \times \Phi_{uv}), \quad \hat{N}_v = (\Phi_{uv} \times \Phi_v) + (\Phi_u \times \Phi_{vv}). \quad (3.1)$$

Por definição, as superfícies NURBS são obtidas a partir de funções racionais por partes. Quando calculamos os coeficientes na [Seção 2.1](#), temos que os da primeira forma fundamental são racionais por definição. Quando se trata da segunda forma fundamental, os coeficientes contém o termo irracional $\|\Phi_u \times \Phi_v\| = \sqrt{A}$ (ver [Equação 2.4](#) e [Equação 2.8](#)). Utilizamos as contas feitas por [Musuvathy *et al.* \(2011\)](#) e [Maekawa e Patrikalakis \(1994\)](#) para implementar o cálculo dos coeficientes de uma superfície.

Com isso, apresentamos a seguir os coeficientes calculados a partir da [Equação 3.1](#) compostos apenas por coeficientes racionais por partes:

Coefficientes da primeira forma fundamental

$$\begin{aligned} E &= \langle \Phi_u, \Phi_u \rangle, & E_u &= 2\langle \Phi_u, \Phi_{uu} \rangle, & E_v &= 2\langle \Phi_u, \Phi_{uv} \rangle, \\ F &= \langle \Phi_u, \Phi_v \rangle, & F_u &= \langle \Phi_v, \Phi_{uu} \rangle + \langle \Phi_u, \Phi_{uv} \rangle, & F_v &= \langle \Phi_v, \Phi_{uv} \rangle + \langle \Phi_u, \Phi_{vv} \rangle, \\ G &= \langle \Phi_v, \Phi_v \rangle, & G_u &= 2\langle \Phi_v, \Phi_{uv} \rangle, & G_v &= 2\langle \Phi_v, \Phi_{vv} \rangle. \end{aligned}$$

Coefficientes racionais da segunda forma fundamental

$$\begin{aligned} \hat{e} &= \langle \hat{N}, \Phi_{uu} \rangle, & \hat{e}_u &= \langle \hat{N}_u, \Phi_{uu} \rangle + \langle \hat{N}, \Phi_{uuu} \rangle, & \hat{e}_v &= \langle \hat{N}_v, \Phi_{uu} \rangle + \langle \hat{N}, \Phi_{uuv} \rangle, \\ \hat{f} &= \langle \hat{N}, \Phi_{uv} \rangle, & \hat{f}_u &= \langle \hat{N}_u, \Phi_{uv} \rangle + \langle \hat{N}, \Phi_{uuv} \rangle, & \hat{f}_v &= \langle \hat{N}_v, \Phi_{uv} \rangle + \langle \hat{N}, \Phi_{uvv} \rangle, \\ \hat{g} &= \langle \hat{N}, \Phi_{vv} \rangle, & \hat{g}_u &= \langle \hat{N}_u, \Phi_{vv} \rangle + \langle \hat{N}, \Phi_{uvv} \rangle, & \hat{g}_v &= \langle \hat{N}_v, \Phi_{vv} \rangle + \langle \hat{N}, \Phi_{vvv} \rangle. \end{aligned}$$

Coefficientes racionais das curvaturas principais

$$\begin{aligned} A &= EG - F^2, \\ A_u &= E_u G + G_u E - 2FF_u, \\ A_v &= E_v G + G_v E - 2FF_v, \\ \hat{B} &= 2\hat{f}F - \hat{e}G - \hat{g}E, \\ \hat{B}_u &= 2(F\hat{f}_u + F_u\hat{f}) - (E_u\hat{g} + E\hat{g}_u) - (G_u\hat{e} + G\hat{e}_u), \\ \hat{B}_v &= 2(F\hat{f}_v + F_v\hat{f}) - (E_v\hat{g} + E\hat{g}_v) - (G_v\hat{e} + G\hat{e}_v), \\ \hat{C} &= \hat{e}\hat{g} - \hat{f}^2, \\ \hat{C}_u &= \hat{e}_u\hat{g} + \hat{e}\hat{g}_u - 2\hat{f}\hat{f}_u, \\ \hat{C}_v &= \hat{e}_v\hat{g} + \hat{e}\hat{g}_v - 2\hat{f}\hat{f}_v. \end{aligned}$$

3.4 Localizando pontos umbílicos

Baseados na estratégia de [Musuvathy et al. \(2011\)](#), utilizamos os pontos umbílicos de uma superfície como pontos de referências para as ridges. Como vimos na [Seção 2.2](#), é sabido o comportamento das curvas ridges em torno de pontos umbílicos genéricos. Por isso, apresentamos nesta seção a estratégia que utilizamos para localizá-los.

Como visto na [Definição 13](#), nos pontos umbílicos temos que $\kappa_1 = \kappa_2$. [Musuvathy et al. \(2011\)](#) reescreveram as equações das curvaturas principais utilizando coeficientes racionais como uma equação só:

$$\kappa_i(u, v) = \left(\frac{-\hat{B} \pm \sqrt{\hat{B}^2 - 4A\hat{C}}}{2A^{3/2}} \right) (u, v) \quad (3.2)$$

onde é possível notar (baseado nas definições da [Equação 3.7](#)) que $\kappa_1 = \kappa_2$ se, e somente se:

$$Q(u, v) = \left(\hat{B}^2 - 4A\hat{C} \right) (u, v) = 0. \quad (3.3)$$

Como observado por [Musuvathy et al. \(2011\)](#), temos que $Q(u, v) \geq 0$, portanto os pontos umbílicos são os pontos mínimos de $Q(u, v)$. Com isso os pontos umbílicos são raízes de Q_u e Q_v . Tais raízes são obtidas resolvendo o sistema de equações racionais por partes:

$$Pu : \begin{cases} 2\widehat{B}\widehat{B}_u - 4A_u\widehat{C} - 4A\widehat{C}_u = 0 \\ 2\widehat{B}\widehat{B}_v - 4A_v\widehat{C} - 4A\widehat{C}_v = 0. \end{cases} \quad (3.4)$$

Para implementar o método de Newton-Rhapson é necessário calcular a matriz jacobiana das funções que compõe o sistema de equações 3.4, então faz-se necessário utilizar as equações:

$$\begin{cases} Pu_u^1 = 2(\widehat{B}_u^2 + \widehat{B}\widehat{B}_{uu}) - 4(A_{uu}\widehat{C} + A_u\widehat{C}_u + A_u\widehat{C}_u + A\widehat{C}_{uu}) \\ Pu_v^1 = 2(\widehat{B}_u\widehat{B}_v + \widehat{B}\widehat{B}_{uv}) - 4(A_{uv}\widehat{C} + A_u\widehat{C}_v + A_v\widehat{C}_u + A\widehat{C}_{uv}) \\ Pu_u^2 = 2(\widehat{B}_u\widehat{B}_v + \widehat{B}\widehat{B}_{uv}) - 4(A_{uv}\widehat{C} + A_u\widehat{C}_v + A_v\widehat{C}_u + A\widehat{C}_{uv}) \\ Pu_v^2 = 2(\widehat{B}_v^2 + \widehat{B}\widehat{B}_{vv}) - 4(A_{vv}\widehat{C} + A_v\widehat{C}_v + A_v\widehat{C}_v + A\widehat{C}_{vv}) \end{cases} \quad (3.5)$$

para o cálculo da matriz jacobiana:

$$dPu_{(u,v)} = \begin{bmatrix} Pu_u^1(u, v) & Pu_v^1(u, v) \\ Pu_u^2(u, v) & Pu_v^2(u, v) \end{bmatrix}$$

3.5 Localizando pontos de partida

No [Capítulo 4](#) apresentamos o nosso método para traçar curvas ridges. O algoritmo depende de pontos de partida. Nos baseamos na estratégia utilizada por [Musuvathy et al. \(2011\)](#) para encontrar esses pontos.

Essa estratégia passa por encontrar esses pontos de partida. Nesta seção apresentamos os sistemas de equações que serão resolvidos para obtermos esses pontos de partida. Todas as equações descritas aqui podem ser obtidas a partir dos coeficientes que apresentamos anteriormente.

3.5.1 Pontos críticos das curvaturas principais

Uma curvatura principal da superfície atinge um ponto crítico quando seu gradiente é nulo. Para encontrar esses pontos críticos precisamos, dada uma curvatura principal κ_i (com $i = 1, 2$), encontrar as soluções (u, v) do sistema:

$$\begin{cases} \kappa_{iu}(u, v) = 0 \\ \kappa_{iv}(u, v) = 0. \end{cases} \quad (3.6)$$

Para executar o algoritmo que encontra as raízes de tal sistema, nos baseamos na estratégia adotada por [Musuvathy et al. \(2011\)](#). Para isso, eles reescreveram a [Equação 3.6](#) utilizando os seguintes coeficientes:

$$\begin{aligned}
P^{(u)} &= \frac{1}{2} \left(-A^{-3/2} \widehat{B}_u + \frac{3}{2} A^{-5/2} A_u \widehat{B} \right), \\
P^{(v)} &= \frac{1}{2} \left(-A^{-3/2} \widehat{B}_v + \frac{3}{2} A^{-5/2} A_v \widehat{B} \right), \\
R^{(u)} &= \frac{1}{2} \left(A^{-3/2} \widehat{B}_u \widehat{B} - 2A^{-1/2} \widehat{C}_u + 4A^{-3/2} A_u \widehat{C} - \frac{3}{2} A^{-5/2} A_u \widehat{B}^2 \right), \\
R^{(v)} &= \frac{1}{2} \left(A^{-3/2} \widehat{B}_v \widehat{B} - 2A^{-1/2} \widehat{C}_v + 4A^{-3/2} A_v \widehat{C} - \frac{3}{2} A^{-5/2} A_v \widehat{B}^2 \right), \\
Q &= \widehat{B}^2 - 4A\widehat{C}.
\end{aligned} \tag{3.7}$$

onde as curvaturas principais são reescritas da seguinte forma:

$$\begin{aligned}
\kappa_{1u} &= P^{(u)} + \frac{R^{(u)}}{\sqrt{Q}}, & \kappa_{1v} &= P^{(v)} + \frac{R^{(v)}}{\sqrt{Q}}, \\
\kappa_{2u} &= P^{(u)} - \frac{R^{(u)}}{\sqrt{Q}}, & \kappa_{2v} &= P^{(v)} - \frac{R^{(v)}}{\sqrt{Q}},
\end{aligned} \tag{3.8}$$

Considerando a [Equação 3.8](#) e o fato de que estamos buscando onde as derivadas parciais dos κ_i se anulam, podemos fazer $\kappa_{1u} = \kappa_{2u}$ e $\kappa_{1v} = \kappa_{2v}$, separar o termo com raiz quadrada de um lado da equação, elevar ao quadrado ambos os lados e obter o novo sistema:

$$P_C : \begin{cases} QP^{(u)2} - R^{(u)2} = 0 \\ QP^{(v)2} - R^{(v)2} = 0 \end{cases} \tag{3.9}$$

cujas soluções (u, v) codificam os pontos críticos de ambas equações κ_1 e κ_2 .

Esse sistema construído por [Musuvathy et al. \(2011\)](#) permite obter os pontos críticos de ambas as curvaturas principais resolvendo apenas um sistema de equações (uma tarefa computacionalmente custosa). Para classificar os pontos críticos como pertencentes às curvatura κ_1 ou κ_2 , basta avaliar os resultados da [Equação 3.9](#) na [Equação 3.8](#).

Para implementar o método de Newton-Rhapson é necessário calcular a matriz jacobiana das funções que compõe o sistema de equações [3.9](#), então faz-se necessário utilizar as equações:

$$\begin{cases} P_{C_u}^1 = Q_u P^{(u)2} + 2QP^{(u)} P_u^{(u)} - 2R^{(u)} R_u^{(u)} \\ P_{C_v}^1 = Q_v P^{(u)2} + 2QP^{(u)} P_v^{(u)} - 2R^{(u)} R_v^{(u)} \\ P_{C_u}^2 = Q_u P^{(v)2} + 2QP^{(v)} P_u^{(v)} - 2R^{(v)} R_u^{(v)} \\ P_{C_v}^2 = Q_v P^{(v)2} + 2QP^{(v)} P_v^{(v)} - 2R^{(v)} R_v^{(v)} \end{cases} \tag{3.10}$$

para o cálculo da matriz jacobiana:

$$dPC_{(u,v)} = \begin{bmatrix} PC_u^1(u,v) & PC_v^1(u,v) \\ PC_u^2(u,v) & PC_v^2(u,v) \end{bmatrix}$$

cujos cálculos das derivadas parciais dos coeficientes $P^{(u)}$ e $P^{(v)}$ é:

$$\begin{aligned} P_u^{(u)} &= \frac{3}{4}A^{-5/2} \left(A_u \widehat{B}_u + A_{uu} \widehat{B} + A_u \widehat{B}_u - \frac{5}{2} A_u A_u \widehat{B} \right) - \frac{1}{2} \left(A^{-3/2} \widehat{B}_{uu} \right), \\ P_v^{(u)} &= \frac{3}{4}A^{-5/2} \left(A_v \widehat{B}_u + A_{uv} \widehat{B} + A_u \widehat{B}_v - \frac{5}{2} A_u A_v \widehat{B} \right) - \frac{1}{2} \left(A^{-3/2} \widehat{B}_{uv} \right), \\ P_u^{(v)} &= \frac{3}{4}A^{-5/2} \left(A_u \widehat{B}_v + A_{uv} \widehat{B} + A_v \widehat{B}_u - \frac{5}{2} A_u A_v \widehat{B} \right) - \frac{1}{2} \left(A^{-3/2} \widehat{B}_{uv} \right), \\ P_v^{(v)} &= \frac{3}{4}A^{-5/2} \left(A_v \widehat{B}_v + A_{vv} \widehat{B} + A_v \widehat{B}_v - \frac{5}{2} A_v A_v \widehat{B} \right) - \frac{1}{2} \left(A^{-3/2} \widehat{B}_{vv} \right), \end{aligned} \quad (3.11)$$

e dos coeficientes $R^{(u)}$ e $R^{(v)}$ é:

$$\begin{aligned} R_u^{(u)} &= \frac{1}{2}A^{-3/2} \left(\widehat{B}\widehat{B}_{uu} + \widehat{B}_u \widehat{B}_u + A_u \widehat{C}_u + 4A_{uu} \widehat{C} + 4A_u \widehat{C}_u \right) \\ &= -\frac{3}{4}A^{-5/2} \left(A_u \widehat{B}\widehat{B}_u + 4A_u A_u \widehat{C} + A_u u \widehat{B}\widehat{B} + 2A_u \widehat{B}\widehat{B}_u \right) \\ &= +\frac{15}{8}A^{-7/2} \left(A_u A_u \widehat{B}\widehat{B} \right) - A^{-1/2} \widehat{C}_{uu}, \\ R_v^{(u)} &= \frac{1}{2}A^{-3/2} \left(\widehat{B}\widehat{B}_{uv} + \widehat{B}_u \widehat{B}_v + A_v \widehat{C}_u + 4A_{uv} \widehat{C} + 4A_u \widehat{C}_v \right) \\ &= -\frac{3}{4}A^{-5/2} \left(A_v \widehat{B}\widehat{B}_u + 4A_u A_v \widehat{C} + A_u v \widehat{B}\widehat{B} + 2A_u \widehat{B}\widehat{B}_v \right) \\ &= +\frac{15}{8}A^{-7/2} \left(A_u A_v \widehat{B}\widehat{B} \right) - A^{-1/2} \widehat{C}_{uv}, \\ R_u^{(v)} &= \frac{1}{2}A^{-3/2} \left(\widehat{B}\widehat{B}_{uv} + \widehat{B}_u \widehat{B}_v + A_u \widehat{C}_v + 4A_{uv} \widehat{C} + 4A_v \widehat{C}_u \right) \\ &= -\frac{3}{4}A^{-5/2} \left(A_u \widehat{B}\widehat{B}_v + 4A_u A_v \widehat{C} + A_u v \widehat{B}\widehat{B} + 2A_v \widehat{B}\widehat{B}_u \right) \\ &= +\frac{15}{8}A^{-7/2} \left(A_u A_v \widehat{B}\widehat{B} \right) - A^{-1/2} \widehat{C}_{uv}, \\ R_v^{(v)} &= \frac{1}{2}A^{-3/2} \left(\widehat{B}\widehat{B}_{vv} + \widehat{B}_v \widehat{B}_v + A_v \widehat{C}_v + 4A_{vv} \widehat{C} + 4A_v \widehat{C}_v \right) \\ &= -\frac{3}{4}A^{-5/2} \left(A_v \widehat{B}\widehat{B}_v + 4A_v A_v \widehat{C} + A_v v \widehat{B}\widehat{B} + 2A_v \widehat{B}\widehat{B}_v \right) \\ &= +\frac{15}{8}A^{-7/2} \left(A_v A_v \widehat{B}\widehat{B} \right) - A^{-1/2} \widehat{C}_{vv}. \end{aligned} \quad (3.12)$$

3.5.2 Pontos roxos das curvas ridge

As curvas ridges de cores diferentes se intersectam nos pontos roxos (ver Definição 19). Num ponto roxo as condições de ridge são satisfeitas para ambas as curvaturas

principais. Isso significa que para obter tais pontos basta encontrar as soluções (u, v) do seguinte sistema de equações não-lineares:

$$\begin{cases} R_1(u, v) = \langle \nabla \kappa_1, d_1 \rangle = 0 \\ R_2(u, v) = \langle \nabla \kappa_2, d_2 \rangle = 0 \end{cases} . \quad (3.13)$$

Esse sistema depende das derivadas parciais de primeira ordem das curvaturas principais, calculadas na [Equação 3.8](#). Para calcular as direções principais utilizamos a [Equação 2.9](#).

Para obter a matriz jacobiana (a fim de executar o método de Newton-Rhapson):

$$\begin{bmatrix} R_{1u}(u, v) & R_{1v}(u, v) \\ R_{2u}(u, v) & R_{2v}(u, v) \end{bmatrix},$$

utilizamos as equações:

$$\begin{cases} R_{iu} = \langle \nabla \kappa_{iu}, d_i \rangle + \langle \nabla \kappa_1, d_{iu} \rangle \\ R_{iv} = \langle \nabla \kappa_{iv}, d_i \rangle + \langle \nabla \kappa_1, d_{iv} \rangle \end{cases} \quad (3.14)$$

Para os cálculos da [Equação 3.14](#) precisamos também das derivadas parciais de segunda ordem das curvaturas principais, que são (a partir da [Equação 3.8](#) e da [Equação 3.7](#)):

$$\begin{aligned} \kappa_{iuu} &= P_u^{(u)} \pm \frac{1}{Q} \left(R_u^{(u)} \sqrt{Q} - \frac{R^{(u)} Q_u}{2\sqrt{Q}} \right) \\ \kappa_{iuv} &= P_v^{(u)} \pm \frac{1}{Q} \left(R_v^{(u)} \sqrt{Q} - \frac{R^{(u)} Q_v}{2\sqrt{Q}} \right) \\ \kappa_{ivu} &= P_u^{(v)} \pm \frac{1}{Q} \left(R_u^{(v)} \sqrt{Q} - \frac{R^{(v)} Q_u}{2\sqrt{Q}} \right) \\ \kappa_{ivv} &= P_v^{(v)} \pm \frac{1}{Q} \left(R_v^{(v)} \sqrt{Q} - \frac{R^{(v)} Q_v}{2\sqrt{Q}} \right) \end{aligned}$$

3.5.3 Pontos de partida a partir de pontos umbílicos

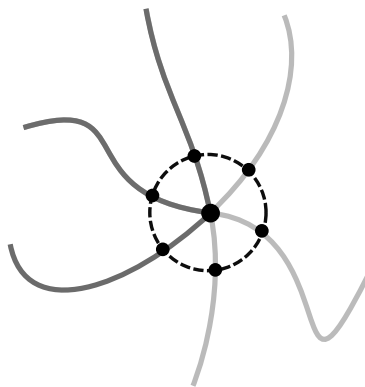
Além dos pontos umbílicos constituírem as características robustas de uma superfície, sabemos classificar o comportamento das curvas ridges numa pequena região ao redor deles (no caso de pontos umbílicos genéricos, como vimos na [Seção 2.2](#)). Na [Seção 3.4](#) mostramos como localizar pontos umbílicos sobre uma superfície NURBS. Nesta seção mostraremos como utilizá-los para encontrar pontos de partida para as curvas ridge.

Inspirados pela técnica utilizada por [Musuvathy et al. \(2011\)](#), construímos um círculo de raio pequeno¹ em torno de cada ponto umbílico. Feito isso, aplicamos o método da

¹ Nos nossos testes o raio utilizado foi $\varepsilon = 0,01$.

bisseção pelos pontos do círculo buscando por pontos que satisfizessem uma das condições de ridge (maximal ou minimal), ou seja, satisfizessem uma das equações no sistema 3.13. A Figura 10 ilustra tais pontos.

Figura 10 – Ilustração dos pontos de partida das curvas ridge encontrados sobre o círculo (traçado) cujo centro é um ponto umbílico.



TRAÇO DE CURVAS RIDGE POR MÉTODO PREDITOR-CORRETOR

Para traçar curvas a partir de pontos de partida, implementamos o método preditor-corretor de continuação numérica baseado no método de Euler-Newton com Adaptação do Tamanho do Passo via Estimativas Assintóticas apresentado por [Allgower e Georg \(2003\)](#).

Para apresentar o método preditor-corretor, primeiro expomos um resumo de seu funcionamento na [Seção 4.1](#) para, nas seções seguintes, detalhar o funcionamento de cada etapa. Na [Seção 4.2](#) apresentamos como funciona a etapa de predição do algoritmo, incluindo os cálculos necessários para se obter a predição, bem como uma heurística utilizada para garantir que o traço tenha um sentido fixo. Complementando o algoritmo, na [Seção 4.3](#) apresentamos a etapa de correção do algoritmo, com ilustrações. Na [Seção 4.4](#) apresentamos, por fim, as condições de parada que utilizamos para a execução e finalização do algoritmo. Aqui vale ressaltar que essa seção é um acréscimo relevante para os trabalhos que utilizamos como referência, uma vez que não encontramos descrição para as condições de paradas nas referências que utilizamos.

4.1 Resumo do algoritmo

O algoritmo consiste em, a partir de um ponto de partida pertencente à curva ridge, buscar o próximo ponto da curva até atingir uma condição de parada.

Todos os métodos desse algoritmo são inteiramente implementado no domínio (contido em \mathbb{R}^2) da superfície. O algoritmo é iterativo e cada uma de suas iterações pode ser dividida em dois métodos:

1. **Predição:** partindo de um ponto p ridge, azul ou vermelho, traça-se a reta tangente

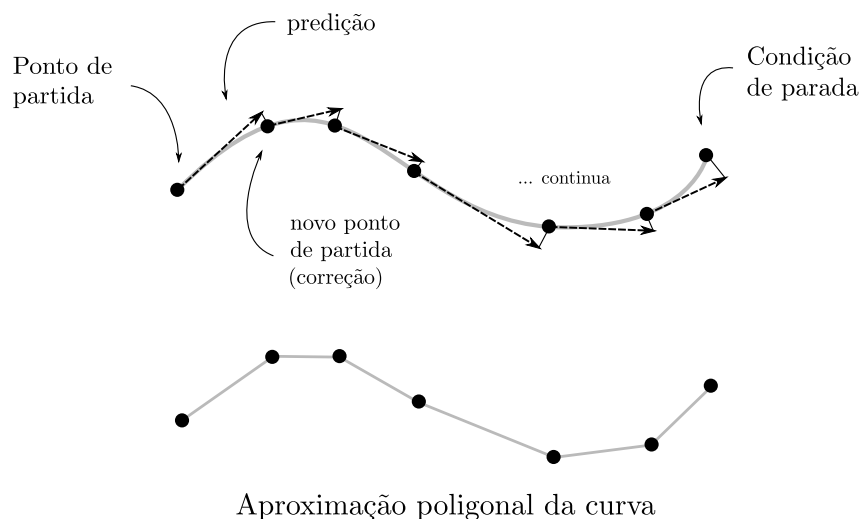
à curva nesse ponto e escolhe-se um ponto q sobre essa reta a uma distância h (ver a [Seção 4.2](#) para mais detalhes sobre a escolha de h) do ponto p .

2. **Correção:** projeta-se o ponto q sobre a curva ridge sendo traçada (verificando a condição de ridge adequada). O ponto obtido na projeção é o próximo ponto da curva, relativo ao ponto da etapa de predição.

Ao executar esse métodos obtemos uma sequência de pontos que, juntos, formam uma aproximação poligonal da curva ridge a qual o ponto de partida inicial pertence. Para obter a curva sobre a superfície NURBS, basta avaliar os pontos obtidos com a parametrização que a define.

Nas próximas seções desse capítulo detalharemos como esse algoritmo funciona. A [Figura 11](#) ilustra algumas iterações do algoritmo.

Figura 11 – Ilustração de curva traçada pelo método predictor-corretor depois de 6 iterações.



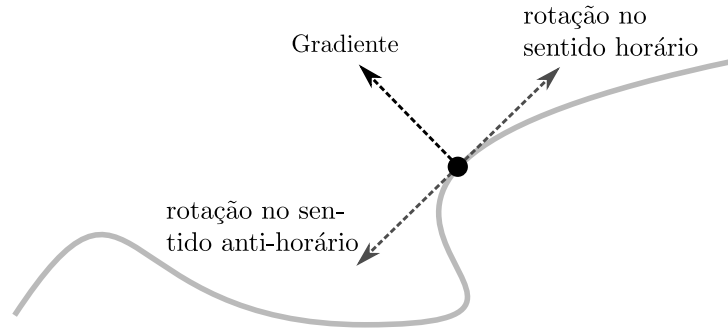
4.2 Etapa de predição

Para a etapa de predição, dado um ponto da curva, queremos obter um novo ponto sobre a reta tangente à curva nesse ponto. Como estamos trabalhando com curvas planas, isso é feito utilizando a direção da reta normal à curva nesse ponto. Essa direção é obtida através do gradiente (o cálculo detalhado encontra-se na [Subseção 4.2.1](#)).

A partir do vetor que representa a direção do gradiente, realiza-se uma rotação de 90° no sentido horário ou anti-horário, a depender do sentido no qual se deseja traçar a curva. Obtém-se, assim, um vetor tangente à curva no ponto dado no início da etapa de predição. A [Figura 12](#) ilustra essa situação.

Para concluir a etapa de predição, escolhemos um ponto da reta tangente diferente do nosso ponto de partida, chamado de ponto predito. Esse ponto é obtido utilizando-se

Figura 12 – Ilustração das direções tangentes obtidas a partir do gradiente da função.



a direção do vetor tangente calculado e uma distância h , chamada de *tamanho do passo*. A cada iteração dessa etapa do algoritmo o tamanho do passo precisa ser decidido. Isso é feito através de uma estimativa assintótica, como descrito no capítulo 6 de [Allgower e Georg \(2003\)](#).

Devidos às particularidades do problema de traçar as curvas ridge e para certificar que o traço se mantenha sempre sobre a mesma curva, alguns cuidados adicionais precisam ser tomados na etapa de predição. Esses cuidados estão descritos na [Subseção 4.2.2](#).

4.2.1 Calculando o gradiente da curva ridge

Como vimos nas equações 2.17 e 2.18, as condições para cada um dos tipos de curvas ridges são as seguintes:

$$\begin{aligned}\Psi_1(u, v) &= \langle \nabla \kappa_1, d_1 \rangle = 0, \\ \Psi_2(u, v) &= \langle \nabla \kappa_2, d_2 \rangle = 0.\end{aligned}\tag{4.1}$$

A partir dessas equações, da [Equação 3.8](#) e da [Equação 2.9](#) calculamos o gradiente $\nabla \Phi_i = (\Psi_{iu}, \Psi_{iv})$ de um ponto da curva ridge onde:

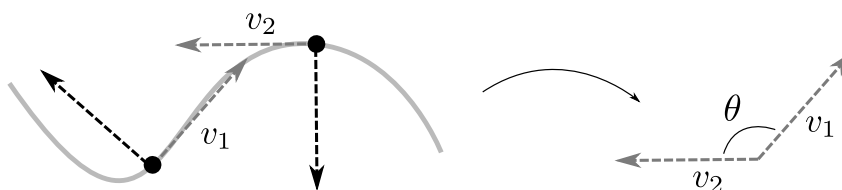
$$\begin{aligned}\Psi_{iu} &= \langle (\nabla \kappa_i)_u, d_i \rangle + \langle \nabla \kappa_i, (d_i)_u \rangle, \\ \Psi_{iv} &= \langle (\nabla \kappa_i)_v, d_i \rangle + \langle \nabla \kappa_i, (d_i)_v \rangle,\end{aligned}\tag{4.2}$$

$$\begin{aligned}\nabla \kappa_i &= (\kappa_{iu}, \kappa_{iv}) = \left(P^{(u)} \pm \frac{R^{(u)}}{\sqrt{Q}}, P^{(v)} \pm \frac{R^{(v)}}{\sqrt{Q}} \right), \\ (\nabla \kappa_i)_u &= (\kappa_{iuu}, \kappa_{iuv}) = \left(P_u^{(u)} \pm \frac{R_u^{(u)} \sqrt{Q} - R^{(u)} \frac{Q_u}{2\sqrt{Q}}}{Q}, P_u^{(v)} \pm \frac{R_u^{(v)} \sqrt{Q} - R^{(v)} \frac{Q_u}{2\sqrt{Q}}}{Q} \right), \\ (\nabla \kappa_i)_v &= (\kappa_{iuv}, \kappa_{ivv}) = \left(P_v^{(u)} \pm \frac{R_v^{(u)} \sqrt{Q} - R^{(u)} \frac{Q_v}{2\sqrt{Q}}}{Q}, P_v^{(v)} \pm \frac{R_v^{(v)} \sqrt{Q} - R^{(v)} \frac{Q_v}{2\sqrt{Q}}}{Q} \right), \\ (d_i)_u &= \begin{bmatrix} \kappa_{iu}F + \kappa_i F_u - f_u \\ e_u - \kappa_{iu}E - \kappa_i E_u \end{bmatrix} \quad \text{e} \quad (d_i)_v = \begin{bmatrix} \kappa_{iv}F + \kappa_i F_v - f_v \\ e_v - \kappa_{iv}E - \kappa_i E_v \end{bmatrix}.\end{aligned}$$

4.2.2 Cuidados adicionais da etapa de predição

Sabemos que a curva ridge a ser traçada é, ao menos, de classe C^1 . Isso significa que suas derivadas são contínuas, ou seja, em pontos suficientemente próximos da curva, a variação de suas derivadas é pequena. Para traçar as curvas, em cada etapa do algoritmo fazemos passos de tamanhos pequenos. Como o algoritmo não calcula a derivada de um ponto da curva diretamente, mas sim um vetor tangente ao ponto, essa pequena variação pode não se verificar. A [Figura 13](#) ilustra a situação.

Figura 13 – Ilustração de dois pontos próximos cujo vetores tangentes v_1 e v_2 obtidos pelo algoritmo (ao rotacionar o gradiente, representado em preto pontilhado, no sentido horário) formam um ângulo $\theta > 90^\circ$.



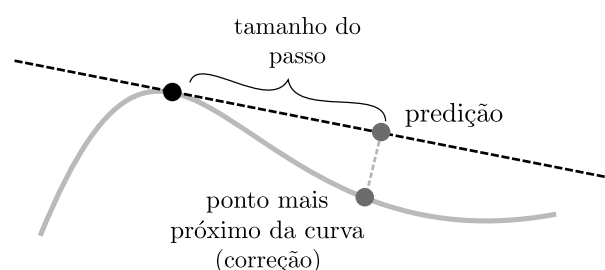
Levando essa propriedade das curvas de classe C^1 em consideração, é possível resolver esse problema através de uma heurística. Ela consiste em impor a seguinte condição na etapa de predição do algoritmo: os vetores tangentes v_1 e v_2 de dois pontos consecutivos do algoritmo de traço da curva devem formar um ângulo θ que seja agudo.

Essa propriedade entre os dois vetores é verificável pelo produto escalar entre eles: assumindo que v_1 e v_2 são unitários, temos que $\langle v_1, v_2 \rangle = \cos \theta$. Com essa informação, sabemos que θ é agudo quando $\langle v_1, v_2 \rangle > 0$. No caso em que $\langle v_1, v_2 \rangle < 0$, como sabemos que v_2 é tangente ao ponto da curva em que foi calculado, passamos a utilizar o vetor $-v_2$ para a etapa de predição do algoritmo.

4.3 Etapa de correção

Na etapa de correção, queremos obter o ponto mais próximo ao ponto predito que pertença à curva ridge que estamos traçando. Esse ponto corrigido é obtido através do método de Newton-Rhapson. A [Figura 14](#) ilustra o que ocorre nesta etapa.

Figura 14 – Etapa de correção do ponto obtido sobre a reta tangente. A “correção” consiste em obter o ponto mais próximo pertencente à curva.



Quando uma etapa de correção se conclui, obtemos o que chamamos de um passo sobre a curva. Esse passo precisa ser validado de acordo com um fator definido na seção 6.1 de [Allgower e Georg \(2003\)](#). Com a validação, obtemos um novo ponto sobre a curva. A partir desse novo ponto podemos executar uma nova etapa de predição, numa nova iteração do algoritmo, ou encerrar o traço da curva se uma condição de parada for satisfeita.

Na etapa de correção o principal cuidado a ser tomado é o que fazer em caso de não convergência do método de Newton-Raphson.

Na implementação determinamos um número `maxit` limite de iterações do método. Caso a convergência do método não ocorra antes dessa quantidade de iterações, fazemos uma diminuição do tamanho do passo feito na etapa de predição e executamos uma nova rodada do algoritmo.

4.4 Condições de parada

Há condições de parada a serem verificadas em cada uma das etapas do algoritmo. Apresentamo-as a seguir.

4.4.1 Etapa de predição

Quando executamos o algoritmo, os cálculos podem ser feitos apenas em regiões onde a superfície e as curvas ridge estejam bem definidas. Para isso, sempre que executa-se uma etapa de predição é necessário verificar duas coisas:

1. se o ponto obtido pertence ao domínio da superfície (ou seja, ao aberto $(0, 1) \times (0, 1)$, pela definição que demos de superfícies NURBS na [Definição 25](#));
2. se está fora de um aberto pequeno de raio $\varepsilon < 10^{-2}$ ao redor de um ponto umbílico (o tamanho de ε foi determinado baseado no algoritmo apresentado por [Musuvathy et al. \(2011\)](#)).

A segunda verificação é necessária pois, caso contrário, a etapa de correção terá problemas quanto à convergência do ponto predito.

4.4.2 Etapa de correção

Podemos considerar que cada execução do método de Newton-Raphson está associada a um ponto da curva: o ponto de partida a partir do qual estamos executando a correção da etapa de predição. Só obtemos um novo ponto da curva se a etapa de correção convergir apropriadamente.

Na maioria dos pontos da curva, o método irá convergir adequadamente na primeira execução. Quando isso não ocorre, determinamos um contador para cada execução do método de Newton-Raphson associada àquele ponto. Quando o método converge, o contador é zerado. Esse contador também tem um valor limite. Quando o valor é ultrapassado, temos uma condição de parada e a execução do algoritmo é encerrada.

RESULTADOS OBTIDOS PELO ALGORITMO

Neste capítulo apresentamos os resultados desta pesquisa. Na [Seção 5.1](#), mostramos algumas superfícies NURBS que nosso algoritmo é capaz de gerar bem como a superfície que aparece tanto nos trabalhos de [Cazals *et al.* \(2006\)](#) e [Musuvathy *et al.* \(2011\)](#). Na [Seção 5.2](#), mostramos os pontos de partida que obtivemos a partir do que descrevemos no [Capítulo 3](#). Na [Seção 5.3](#), mostramos os traços de curvas ridge que obtivemos a partir do algoritmo descrito no [Capítulo 4](#).

Dedicamos uma parte deste capítulo para tratar exclusivamente sobre a validação e verificação dos algoritmos que desenvolvemos. Na [Seção 5.4](#), apresentamos como utilizamos o método desenvolvido por [Cazals *et al.* \(2006\)](#) para verificar e validar os resultados de nossos algoritmos. Citamos também o algoritmo Maple que desenvolvemos para validar os resultados das equações dos capítulos [2](#) e [3](#).

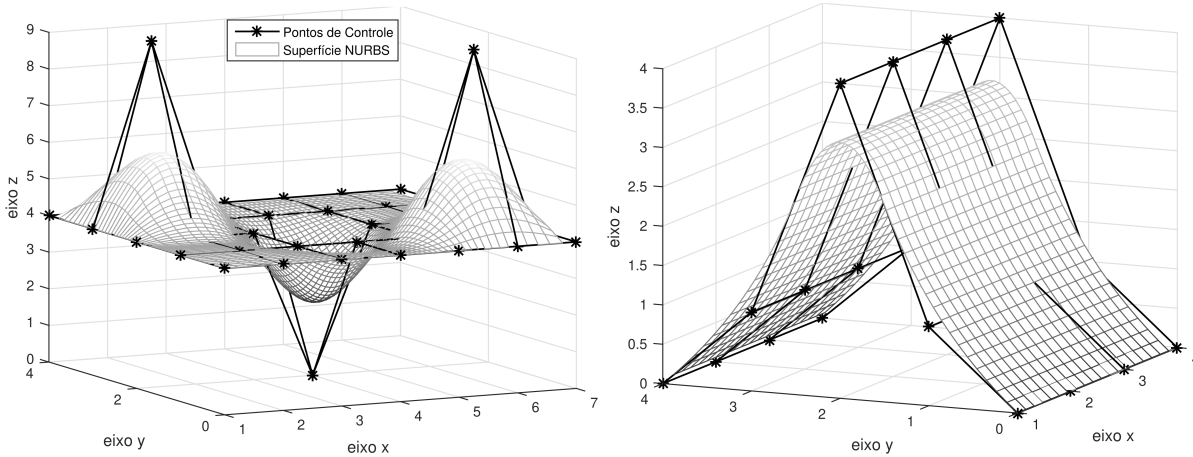
Durante a disciplina de Variedades Computacionais, ministrada por Antonio Castelo Filho no segundo semestre de 2020, escrevi um capítulo de livro introdutório sobre curvas e superfícies NURBS a ser publicado. Em 2021 publiquei um resumo baseado nos resultados preliminares dessa dissertação intitulado “Extraction of robust features on human faces” que pode ser visto em [Figur, Castelo e Tari \(2021\)](#).

5.1 Superfícies NURBS

O algoritmo construído é capaz de gerar uma superfície a partir de pontos de controle e vetores de nós. Para as superfícies exibidas neste trabalho utilizamos vetores de nós uniforme, definidos no intervalo $[0, 1]$. Na [Figura 15](#) podemos ver exemplos de superfícies geradas pelo nosso algoritmo.

Além de obter as parametrizações das superfícies, podemos também calcular as derivadas parciais de tais parametrizações. Como vimos no [Capítulo 2](#), isso permite obter

Figura 15 – Superfícies NURBS juntamente com seus pontos de controle.



as equações das características robustas da superfície.

5.1.1 Superfície utilizada para validação

A matriz em 5.1 representa os pontos de controle que definem a superfície NURBS que utilizamos para executar os algoritmos desta dissertação. Essa superfície também é utilizada nos trabalhos de [Cazals et al. \(2006\)](#), [Cazals et al. \(2005\)](#) e [Musuvathy et al. \(2011\)](#). A [Figura 16](#) é uma representação dessa superfície obtida por nosso algoritmo.

$$\begin{bmatrix} (0, 0, 0) & (1/4, 0, 0) & (2/4, 0, 0) & (3/4, 0, 0) & (1, 0, 0) \\ (0, 1/4, 0) & (1/4, 1/4, 1) & (2/4, 1/4, -1) & (3/4, 1/4, -1) & (1, 1/4, 0) \\ (0, 2/4, 0) & (1/4, 2/4, -1) & (2/4, 2/4, 1) & (3/4, 2/4, 1) & (1, 2/4, 0) \\ (0, 3/4, 0) & (1/4, 3/4, 1) & (2/4, 3/4, -1) & (3/4, 3/4, 1) & (1, 3/4, 0) \\ (0, 1, 0) & (1/4, 1, 0) & (2/4, 1, 0) & (3/4, 1, 0) & (1, 1, 0) \end{bmatrix}. \quad (5.1)$$

Essa mesma superfície NURBS obtida pelos pontos de controle na [Figura 16](#) também pode ser obtida a partir do gráfico da seguinte função $h(u, v)$:

$$\begin{aligned} h_1(u, v) = & 116u^4v^4 - 200u^4v^3 - 312u^3v^4 + 108u^4v^2 + 592u^3v^3 + 252u^2v^4 - 24u^4v - 360u^3v^2 \\ & - 504u^2v^3 - 56uv^4 + 80u^3v + 324u^2v^2 + 112uv^3 - 72u^2v - 72uv^2 + 16uv. \end{aligned} \quad (5.2)$$

5.2 Pontos de partida

5.2.1 Pontos críticos das curvaturas principais

Executando o algoritmo descrito no [Capítulo 3](#) obtemos os pontos críticos das curvaturas principais apresentados na [Figura 17](#).

Figura 16 – Superfície obtida a partir dos pontos de controle na matriz 5.1.

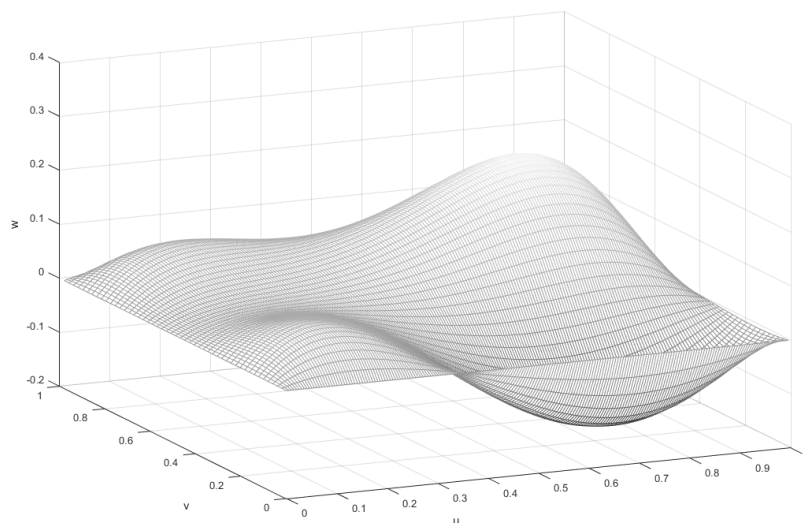
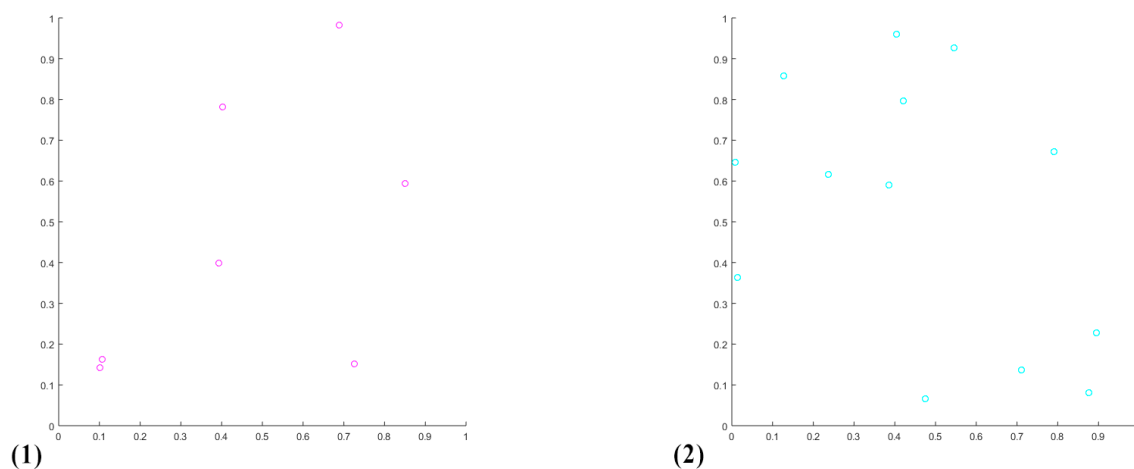


Figura 17 – Pontos críticos das curvaturas principais: (1) pontos críticos vermelhos. (2) pontos críticos azuis.



Considerando as curvas de nível das curvaturas principais na Figura 18 observamos que os pontos críticos obtidos encontram-se em singularidades dessas curvas, como esperado.

5.2.2 Pontos roxos das curvaturas principais

Executando o algoritmo descrito no Capítulo 3 obtivemos os pontos roxos das curvaturas principais, que são apresentados na Figura 19.

Figura 18 – Curvas de nível das linhas de curvatura e suas singularidades: (1) curvas de nível das linhas de curvatura vermelhas. (2) curvas de nível das linhas de curvatura azuis.

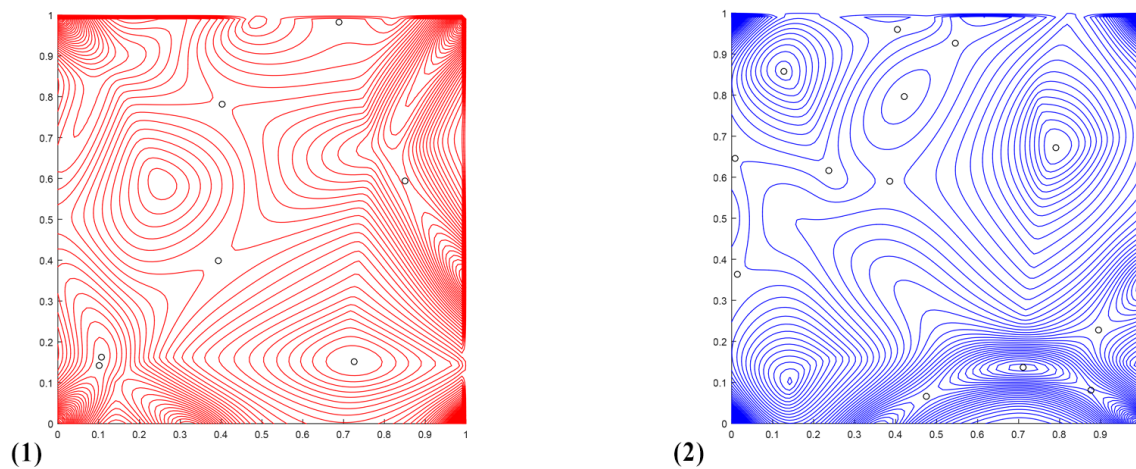
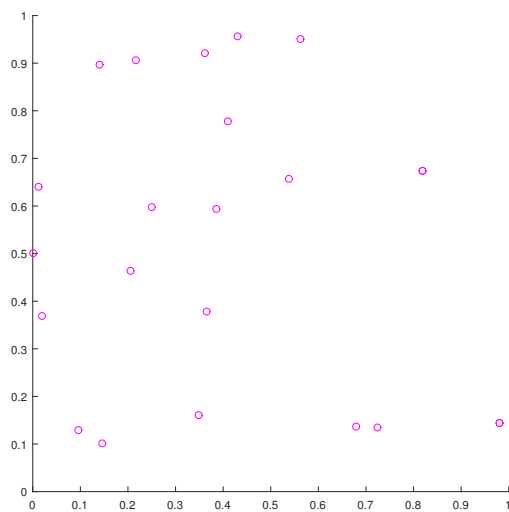


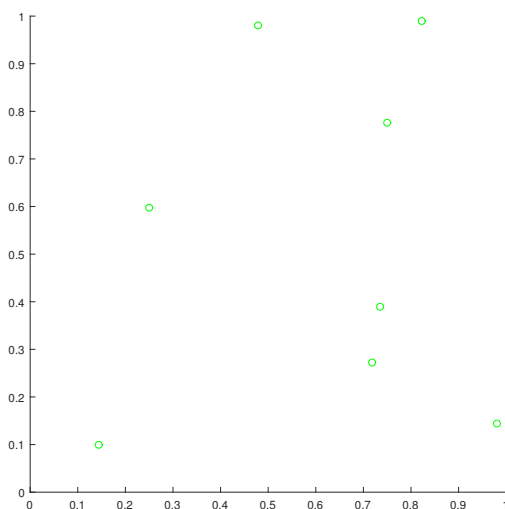
Figura 19 – Pontos roxos da superfície.



5.2.3 Pontos umbílicos

Executando o algoritmo descrito no [Capítulo 3](#) obtivemos os pontos umbílicos da superfície, que são apresentados na [Figura 20](#).

Figura 20 – Pontos umbólicos da superfície.

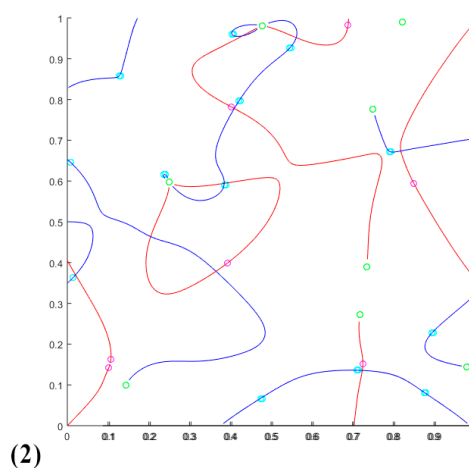
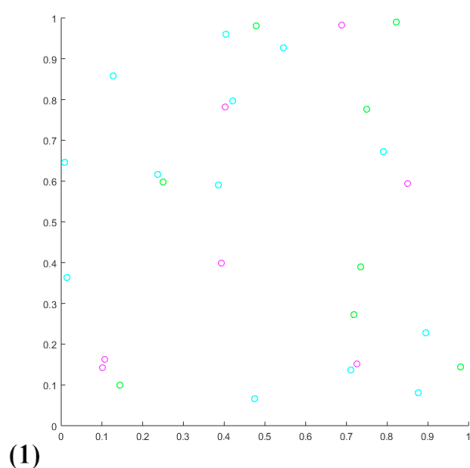


5.3 Curvas Ridge

5.3.1 Curvas ridges traçadas a partir dos pontos críticos

Como vimos no [Capítulo 4](#), utilizamos os pontos críticos vermelhos e azuis como pontos de partida para traçar curvas ridge. A [Figura 21](#) apresenta os traços que obtivemos a partir dos pontos críticos, utilizando os pontos umbólicos como uma das condições de parada.

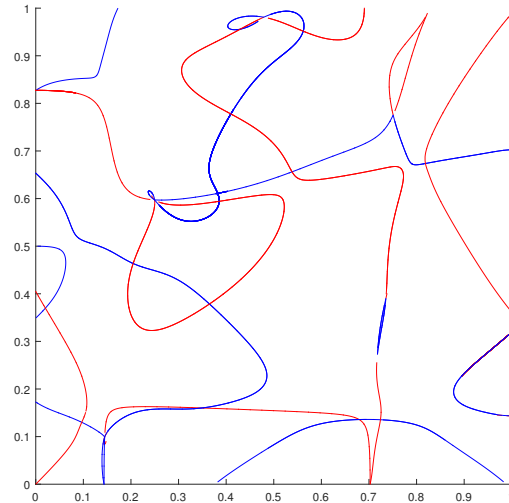
Figura 21 – Traçando ridges a partir de pontos críticos. Em (1) são apresentados os pontos umbólicos em verde, pontos críticos vermelhos em magenta e pontos críticos azuis em ciano. Em (2) são apresentados traços de curvas ridges (em suas respectivas cores) obtidas a partir dos pontos críticos utilizando os pontos umbólicos como uma das condições de parada.



5.3.2 Curvas ridge traçadas a partir dos pontos umbílicos

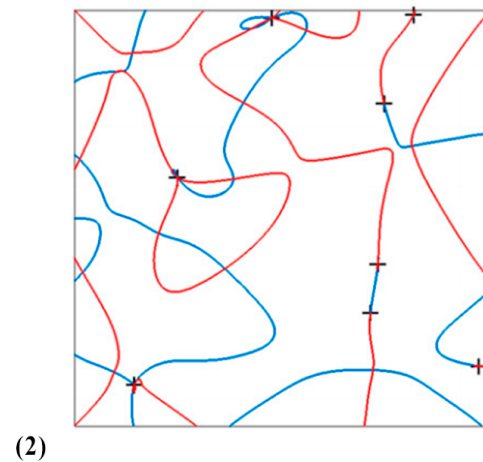
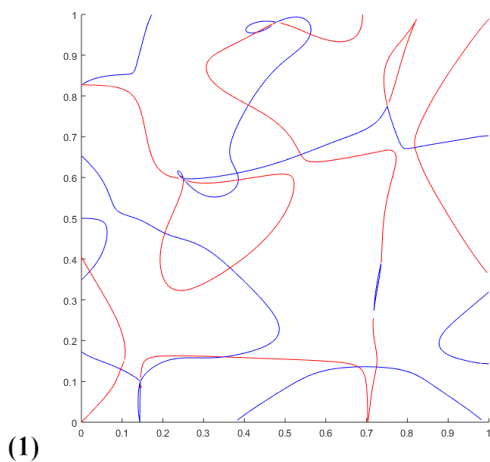
A Figura 22 apresenta os traços de curvas ridges que obtivemos com o algoritmo. Eles foram obtidos a partir dos pontos críticos e dos pontos de partida calculados numa pequena bola ao redor dos pontos umbílicos.

Figura 22 – Traços de curvas ridge obtidas a partir dos pontos críticos e dos pontos umbílicos.



A Figura 23 apresenta que o nosso algoritmo não foi capaz de reproduzir com total fidelidade os resultados obtidos por Musuvathy *et al.* (2011). Pode ser pelo fato de estarmos trabalhando com cálculos numéricos numa superfície muito susceptível a variações de ponto flutuante. Investigações mais profundas e refinamento do algoritmo cabem em trabalhos futuros, como indicamos no Capítulo 6.

Figura 23 – Comparação entre os traços de ridges obtidos pelo nosso algoritmo na figura (1) e os traços de ridges obtidos por Musuvathy *et al.* (2011) na figura (2).



5.4 Métodos para verificação do algoritmo

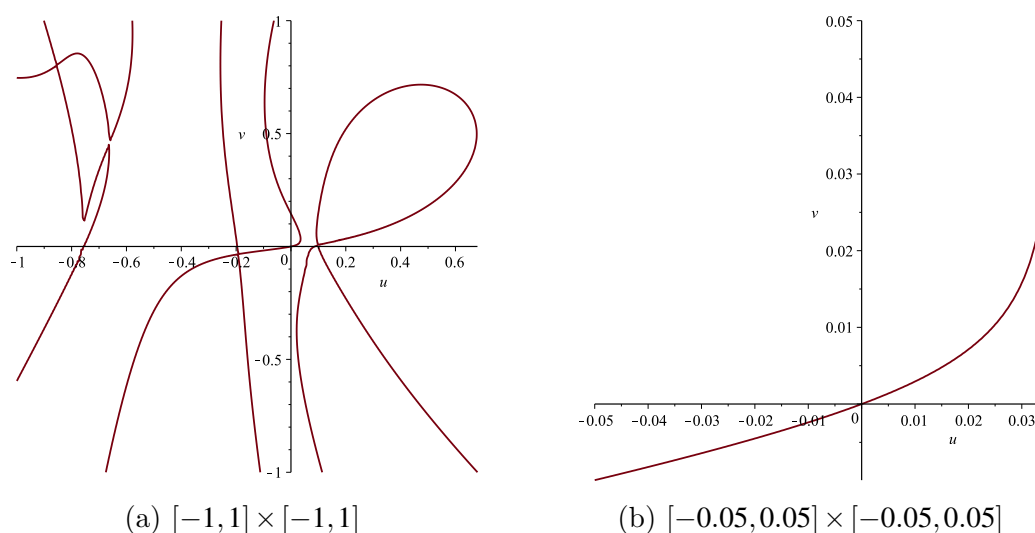
O método de [Cazals et al. \(2006\)](#) utiliza os coeficientes de Weingarten (como na [Definição 11](#)) para obter uma equação implícita que codifica os ridges azuis, vermelhos e os pontos umbílicos de uma superfície parametrizada por polinômios.

Implementamos o método de [Cazals et al. \(2006\)](#) utilizando *Maple 18*. Depois calculamos os coeficientes necessários para o gráfico das seguintes funções:

$$h_2(u, v) = 7u^3 + 2u^2v + 3uv^2 + u^2 + 2v^2, \quad (5.3)$$

$$h_3(u, v) = -\frac{1}{12}u - \frac{35}{24}v + \frac{5}{2}uv + \frac{1}{2}u^2v - 3uv^2 - \frac{17}{12}u^2 + \frac{4}{3}v^2 + u^3 + \frac{1}{3}v^3 + \frac{5}{16}, \quad (5.4)$$

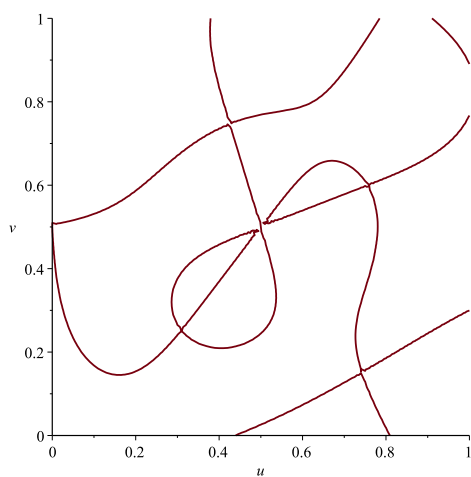
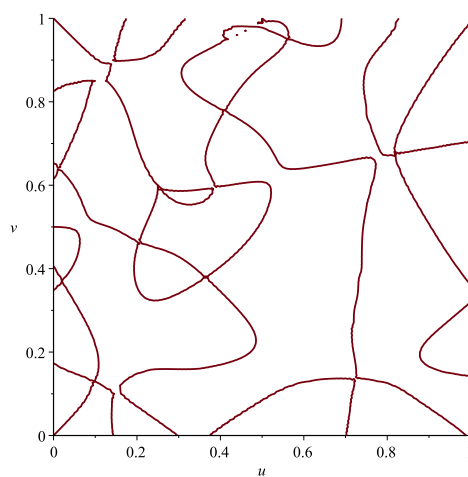
Figura 24 – Ridges da superfície $\Phi_2(u, v) = (u, v, h_2(u, v))$



A [Figura 24](#) apresenta os ridges obtidos com a implementação do código elaborado por [Cazals et al. \(2006\)](#) para o gráfico de h_2 (5.4). A [Figura 24a](#) mostra vários ridges da região $[-1, 1] \times [-1, 1]$ do domínio. De acordo com [Bruce, Giblin e Tari \(1999, p. 262\)](#), existe um aberto em torno do zero onde a curva ridge que passa ali é uma curva regular, como podemos ver na [Figura 24b](#).

A [Figura 25](#) mostra os ridges dos gráficos das funções h_1 (5.2) e h_3 (5.4) presentes nos trabalhos [Cazals et al. \(2006\)](#) e [Cazals et al. \(2005\)](#).

Além do método no artigo [Cazals et al. \(2006\)](#), implementamos também no Maple 18 o cálculo de todas as equações apresentadas nos capítulos 2, 3 e 4 para poder avaliar essas equações e validar com os resultados numéricos obtidos no Matlab. Em comparação com os resultados do Maple (que consideramos mais precisos), todos os coeficientes calculados no Matlab foram obtidos com precisão de 6 casas decimais.

Figura 25 – Ridges de superfícies na região $[0, 1] \times [0, 1]$ (a) $\Phi_3(u, v) = (u, v, h_3(u, v))$ (b) $\Phi_1(u, v) = (u, v, h_1(u, v))$

CONCLUSÃO

Apesar das características robustas poderem ser obtidas através de equações isso não significa que as suas extrações computacionais sejam simples. Vimos nos resultados deste trabalho no [Capítulo 5](#) que ao realizar cálculos próximos a singularidades das curvas robustas os resultados obtidos são muito susceptíveis às imprecisões numéricas. Isso se deve ao fato de que para obter cada um dos pontos ridge da superfície, dependemos de realizar uma longa sequência de cálculos numéricos.

Isso leva a uma inquietação: como podemos ver em [Mari, Hetroy-Wheeler e Subsol \(2020\)](#), os métodos para extração de características robustas, em geral, “traduzem literalmente” os conceitos da Geometria Diferencial sobre características robustas para os métodos disponíveis de Processamento Geométrico para tratar nuvens de pontos, malhas e superfícies parametrizadas. Essa tradução geralmente acompanha perdas de desempenho e de precisão dos resultados, uma vez que sempre haverá imprecisões numéricas inerentes aos métodos quando estamos operando com uma teoria contínua no mundo discreto.

Para melhorar essa situação, um caminho possível é desenvolver uma teoria equivalente à das características robustas dentro da Geometria Discreta. Isso permitiria trabalhar com mais naturalidade com os conceitos de variedades computacionais que já conhecemos. Esperamos que o presente trabalho abra caminho para o desenvolvimento de uma teoria como essa.

Dito isso, indicamos aqui possíveis melhoramentos e ramificações dos resultados obtidos nesta dissertação.

6.1 Trabalhos futuros

6.1.1 Refinamento do algoritmo ao redor de pontos umbílicos

Durante toda a realização do nosso trabalho, pudemos notar que o comportamento das ridges num aberto pequeno ao redor dos pontos umbílicos é bastante sensível a variações numéricas. Seria possível, para obter pontos de partida ridge ao redor de pontos umbílicos, aplicar os métodos de subdivisão de domínio que utilizamos para obter os outros pontos de partida.

6.1.2 Extração de outras características robustas

Consideramos que é possível aplicar o algoritmo preditor-corretor para traçar também as curvas parabólicas (Definição 16) e as curvas sub-parabólicas (Definição 17) de uma superfície. Uma vez explicitadas as equações necessárias para calculá-las acreditamos que o mesmo método preditor-corretor que utilizamos para obter os ridges possa ser utilizado para obtê-las. É necessário definir condições de parada para o traço delas estudando a teoria mais a fundo.

6.1.3 Extração de Regiões Geometricamente Salientes

Retomando as perguntas apresentadas no Capítulo 1 e nossas motivações para este trabalho apresentadas no Capítulo 1, a extração das características robustas como fizemos pode nos oferecer ferramentas e maneiras de diferenciar rostos.

As Regiões Geometricamente Salientes apresentadas por Musuvathy *et al.* (2011) podem ser um caminho para essa tarefa. O artigo já demonstra indícios de que mesmo com curvas ridge semelhantes essas regiões são perceptivelmente diferentes a olho nu. Esse tipo de propriedade indica que essas regiões podem ser utilizadas como um método de comparação de superfícies que seja menos sensível a erros numéricos.

6.1.4 Reconstrução de superfícies

Já dentro do projeto temático da FAPESP, em que este trabalho está inserido, temos um trabalho sendo desenvolvido em paralelo que permite reconstruir superfícies com bastante precisão utilizando apenas algumas curvas.

Os resultados preliminares do projeto indicam que “existem curvas que são melhores que outras” para a reconstrução. Supomos que as diversas características robustas de uma superfície podem ser curvas adequadas a essa tarefa de reconstrução.

Um trabalho como esse permitiria lidar com uma quantidade muito menor de dados ao lidar com superfícies, sendo possível até mesmo ter “uma equação” (ou equações) que

caracterize cada rosto e cada superfície, no sentido de uma impressão digital. Tal impressão digital poderia ser útil não apenas para trabalhos de reconhecimento facial, mas também como uma assinatura digital de superfícies como um todo.

6.1.5 Reconhecimento Facial

Nos métodos de reconhecimento facial muitas vezes é necessário encontrar regiões comuns a todo rosto humano como ponto de partida para buscar regiões únicas que caracterizem um rosto específico. Como indicado no trabalho de [Hallinan *et al.* \(2018\)](#), o *nose ridge* já é uma curva utilizada como referência para trabalhos do tipo.

Uma classificação adequada das características robustas dos rostos humanos pode facilitar o reconhecimento de expressões faciais, características étnicas, de gênero etc. para aprimorar os algoritmos de reconhecimento facial utilizando a geometria dos rostos a seu favor.

REFERÊNCIAS

ALLGOWER, E. L.; GEORG, K. **Introduction to Numerical Continuation Methods**. Society for Industrial and Applied Mathematics, 2003. ISBN 978-0-89871-544-6. Disponível em: <<https://doi.org/10.1137/1.9780898719154>>. Citado nas páginas 21, 47, 49 e 51.

BITTNER, L. Simplicial methods for the solution of systems of nonlinear equations. **Journal of Applied Mathematics and Mechanics**, v. 56, p. 65–73, 1976. Citado na página 36.

BRUCE, J.; FIDAL, D. On binary differential equations and umbilics. **Proceedings of the Royal Society of Edinburgh: Section A Mathematics**, Royal Society of Edinburgh Scotland Foundation, v. 111, n. 1-2, p. 147–168, 1989. Citado nas páginas 27 e 28.

BRUCE, J. W. Projections and reflections of generic surfaces in \mathbb{R}^3 . **Mathematica Scandinavica**, *Mathematica Scandinavica*, v. 54, n. 2, p. 262–278, 1984. ISSN 00255521, 19031807. Disponível em: <<http://www.jstor.org/stable/24491554>>. Citado nas páginas 28 e 30.

BRUCE, J. W.; GIBLIN, P. J.; TARI, F. Families of surfaces: Focal sets, ridges and umbilics. **Mathematical Proceedings of the Cambridge Philosophical Society**, v. 125, p. 243 – 268, 01 1999. Citado na página 59.

CAZALS, F.; FAUGÈRE, J.-C.; POUGET, M.; ROUILLIER, F. Topologically certified approximation of umbilics and ridges on polynomial parametric surface. **Bulletin of Sociological Methodology/Bulletin de Méthodologie Sociologique**, v. 37, n. 1, p. 55–57, 9 2005. ISSN 0759-1063, 2070-2779. Citado nas páginas 54 e 59.

_____. The implicit structure of ridges of a smooth parametric surface. **Computer Aided Geometric Design**, v. 23, n. 7, p. 582–598, 10 2006. ISSN 01678396. Citado nas páginas 20, 53, 54 e 59.

CAZALS, F.; POUGET, M. Topology driven algorithms for ridge extraction on meshes. INRIA, 01 2005. [Research Report]. Citado na página 18.

CHAPRA, S. C. **Métodos Numéricos Para Engenharia**. 5. ed. [S.l.]: McGraw-Hill, 2011. ISBN 8586804878,9788586804878. Citado na página 35.

do Carmo, M. P. **Differential Geometry of Curves and Surfaces: Revised and Updated Second Edition**. 2ed.. ed. [S.l.]: Dover Publications, Inc, 2016. (Dover Books on Mathematics). ISBN 0-486-80699-5,978-0-486-80699-0. Citado na página 23.

FIGUR, A.; CASTELO, A.; TARI, F. Extraction of robust features on human faces. **Proceeding Series of the Brazilian Society of Computational and Applied Mathematics**, v. 8, 2021. ISSN 2359-0793. Disponível em: <<https://proceedings.sbmac.org.br/sbmac/article/view/133702>>. Citado na página 53.

HALLINAN, P. L.; GORDON, G. G.; YUILLE, A. L.; GIBIN, P.; MUMFORD, D. **Two- and three-dimensional patterns of the face**. [S.l.]: CRC Press, 2018. ISBN 978-1-56881-087-4. Citado nas páginas 18 e 63.

HILBERT, D.; COHN-VOSSEN, S. **Geometry and the Imagination**. 2. ed. [S.l.]: Chelsea Publishing Company, 1990. ISBN 0828410879. Citado na página 17.

IZUMIYA, S.; FUSTER, M. del C. R.; RUAS, M. A. S.; TARI, F. **Differential geometry from singularity theory viewpoint**. [S.l.]: World Scientific, 2016. ISBN 978-981-4590-44-0. Citado na página 23.

KLEIN, F. **Apollo of the Belvedere with parabolic curve**. ano desconhecido. [Online; Acesso em: 11 jan. 2021]. Disponível em: <<http://modellsammlung.uni-goettingen.de/index.php?lang=en&r=0&sr=0&m=211>>. Citado na página 17.

MAEKAWA, T.; PATRIKALAKIS, N. M. Interrogation of differential geometry properties for design and manufacture. **The Visual Computer**, v. 10, n. 4, p. 216–237, Apr 1994. ISSN 0178-2789, 1432-8726. Citado na página 39.

MARI, J.-L.; HETROY-WHEELER, F.; SUBSOL, G. **Geometric and Topological Mesh Feature Extraction for 3D Shape Analysis**. [S.l.]: John Wiley & Sons, Incorporated, 2020. ISBN 978-1-119-68104-5. Citado nas páginas 19, 20 e 61.

MUSUVATHY, S.; COHEN, E.; DAMON, J.; SEONG, J.-K. Principal curvature ridges and geometrically salient regions of parametric b-spline surfaces. **Computer-Aided Design**, v. 43, p. 756–770, 07 2011. Citado nas páginas 19, 20, 21, 31, 39, 40, 41, 42, 44, 51, 53, 54, 58 e 62.

OHTAKE, Y.; BELYAEV, A.; SEIDEL, H.-P. Ridge-valley lines on meshes via implicit surface fitting. **ACM Trans. Graph.**, Association for Computing Machinery, New York, NY, USA, v. 23, n. 3, p. 609–612, ago. 2004. ISSN 0730-0301. [Online; Acesso em: 21 jan. 2021]. Disponível em: <<https://doi.org/10.1145/1015706.1015768>>. Citado na página 18.

PIEGL, L.; TILLER, W. **The NURBS Book**. Springer Berlin Heidelberg, 1997. (Monographs in Visual Communication). [Online; Acesso em: 21 jan. 2021]. ISBN 978-3-540-61545-3. Disponível em: <<http://link.springer.com/10.1007/978-3-642-59223-2>>. Citado nas páginas 20, 23, 33 e 37.

PORTEOUS, I. R. **Geometric differentiation for the intelligence of curves and surfaces**. [S.l.]: Cambridge University Press, 1994. ISBN 052139063X. Citado nas páginas 17, 18 e 28.

QUARTERONI, A.; SACCO, R.; SALERI, F. **Numerical Mathematics**. second. Springer-Verlag Berlin Heidelberg, 2007. (Texts in Applied Mathematics 37). ISBN 0387989595,9780387989594,9780387227504. Disponível em: <<http://gen.lib.rus.ec/book/index.php?md5=50652392ea858dc972f04f363110c6f7>>. Citado na página 35.

ROGERS, D. F. **An Introduction to NURBS: With Historical Perspective (The Morgan Kaufmann Series in Computer Graphics)**. 1st. ed. [S.l.]: Morgan Kaufmann, 2000. (The Morgan Kaufmann Series in Computer Graphics). ISBN 1558606696,9781558606692. Citado nas páginas 20 e 23.

RYABEN'KII, V. S.; TSYNKOV, S. V. **A Theoretical Introduction to Numerical Analysis**. 1. ed. [S.l.: s.n.], 2006. ISBN 1584886072,9781584886075,9780367453398. Citado na página 35.

SOTOMAYOR, J.; GUTIERREZ, C. Structurally stable configurations of lines of principal curvature. In: **Bifurcation, théorie ergodique et applications - 22-26 juin 1981**. Société mathématique de France, 1982, (Astérisque, 98-99). [Online; Acesso em: 04 fev. 2021]. Disponível em: <www.numdam.org/item/AST_1983__98-99__195_0/>. Citado nas páginas 27 e 28.

