



ESTIMATIVA DE CURVATURA EM SUPERFÍCIES VIA AJUSTE
QUADRÁTICO LOCAL E ANÁLISE COMPARATIVA

Breno Mota Sarmento

Projeto de Graduação apresentado ao Curso de Engenharia Mecânica da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro.

Orientadores: Gustavo Rabello dos Anjos

Daniel Barbedo Vasconcelos Santos

Rio de Janeiro

Maio de 2026



UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

Departamento de Engenharia Mecânica

DEM/POLI/UFRJ



ESTIMATIVA DE CURVATURA EM SUPERFÍCIES VIA AJUSTE
QUADRÁTICO LOCAL E ANÁLISE COMPARATIVA

Breno Mota Sarmento

PROJETO DE GRADUAÇÃO SUBMETIDO AO CORPO DOCENTE
DO DEPARTAMENTO DE ENGENHARIA MECÂNICA DA ESCOLA
POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO
GRAU DE ENGENHEIRO MECÂNICO.

Aprovado por:

Prof. Gustavo Rabello dos Anjos, Ph.D.

Daniel Barbedo Vasconcelos Santos, D.Sc.

Profa. Lavinia Maria Sanabio Alves Borges, D.Sc.

Prof. Marcelo Amorim Savi, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

MAIO DE 2026

Sarmento, Breno Mota

Estimativa de Curvatura em Superfícies via Ajuste Quadrático Local e Análise Comparativa/ Breno Mota Sarmento. – Rio de Janeiro: UFRJ/Escola Politécnica, 2026.

XXI, 139 p.: il.; 29,7cm.

Orientadores: Gustavo Rabello dos Anjos

Daniel Barbedo Vasconcelos Santos

Projeto de Graduação – UFRJ/ Escola Politécnica/
Curso de Engenharia Mecânica, 2026.

Referências Bibliográficas: p. 101 – 105.

1. Estimativa de Curvatura. 2. Curvatura Média. 3. Ajuste Quadrático Local. 4. Operador Laplace–Beltrami. 5. Superfícies Discretas. I. Rabello dos Anjos, Gustavo; Barbedo Vasconcelos Santos, Daniel. II. Universidade Federal do Rio de Janeiro, Escola Politécnica, Curso de Engenharia Mecânica. III. Estimativa de Curvatura em Superfícies via Ajuste Quadrático Local e Análise Comparativa.

Aos meus pais, pelo apoio incondicional e pelas oportunidades que me permitiram chegar até aqui.

Agradecimentos

Agradeço, em primeiro lugar, ao Professor Gustavo Rabello dos Anjos, pela oportunidade de iniciação científica que me concedeu e pela confiança depositada em mim desde o início. Este trabalho nasceu a partir do interesse despertado nas aulas de Mecânica dos Fluidos e Transferência de Calor Computacional, disciplina que cursei com ele e que foi fundamental para aproximar-me da área. Sou muito grato por ter acolhido minha iniciativa, viabilizado o projeto e tornado possível sua realização, inclusive por meio da obtenção da bolsa de iniciação científica. Agradeço, ainda, pela disponibilidade, pela orientação ao longo do desenvolvimento do trabalho e pelas oportunidades que me proporcionou durante essa trajetória.

Ao Daniel Barbedo Vasconcelos Santos, coorientador deste trabalho, expresso minha sincera gratidão pelo acompanhamento constante, pelo suporte técnico e pela paciência ao longo de todo o processo. Sua ajuda foi essencial em diversos momentos do desenvolvimento deste trabalho, contribuindo não apenas para a condução técnica da pesquisa, mas também para que eu pudesse seguir com mais segurança durante essa etapa.

Agradeço a todos os professores da Engenharia Mecânica da UFRJ pelos ensinamentos transmitidos ao longo da graduação. Sou especialmente grato àqueles que foram além da exposição do conteúdo teórico e mostraram as aplicações práticas dos temas estudados, contribuindo para uma compreensão mais concreta e interessante de como a engenharia se manifesta no mundo real. Em especial, agradeço aos professores Daniel Alves Castello, Gustavo César Rachid Bodstein, Hécio Rangel Barreto Orlande, Marcelo José Colaço, Nísio de Carvalho Lobo Brum, Sylvio José Ribeiro de Oliveira, Max Suell Dutra, cujas aulas, conhecimentos e experiências marcaram minha formação e me ajudaram a enxergar os fenômenos da engenharia de forma mais clara, aplicada e instigante.

Agradeço a toda a equipe do LABMFA pelo acolhimento e pelo ambiente de convivência acadêmica proporcionado durante esta etapa.

À Universidade Federal do Rio de Janeiro, agradeço pelas oportunidades de aprendizado, pela formação proporcionada e pela possibilidade de conviver com pessoas tão qualificadas e inspiradoras ao longo da graduação.

À FAPERJ, agradeço pela concessão da bolsa de iniciação científica, que foi fundamental para viabilizar o desenvolvimento deste trabalho.

Aos meus amigos do colégio — Matheus, Duran, João Guilherme, Enzo, Rayssa, Marcelle e Mariana —, agradeço pela amizade ao longo dos anos e por estarem presentes em diferentes momentos da minha trajetória. Mesmo com os caminhos que cada um seguiu, os momentos que passamos juntos sempre foram muito importantes para mim. Sou especialmente grato pelas saídas e momentos de descontração, que muitas vezes vieram justamente quando eu estava mais cansado; estar com vocês me ajudava a recuperar as energias e voltar mais leve para a rotina. Levo comigo todas as experiências compartilhadas e sou grato por poder contar com vocês.

Agradeço também aos meus colegas de turma, pelas experiências compartilhadas ao longo da graduação e pelos trabalhos desenvolvidos em conjunto. Em especial, agradeço ao Cristiano, uma das primeiras pessoas de quem me aproximei mais após a pandemia, por todo o apoio, parceria e ajuda ao longo do curso. Estendo também meu agradecimento ao nosso “grupo do almoço” — Eduardo, Felipe, Brandão, Guilherme e Stefano —, pelas conversas, pelas risadas, pela troca de materiais e por compartilharem comigo os desafios dessa trajetória. A convivência com vocês tornou essa caminhada mais leve e significativa.

Agradeço aos meus pais, Roni e Cristiane, pela oportunidade, pelo apoio e pelo suporte que sempre me deram, bem como pela compreensão durante os períodos de maior dedicação ao curso. Agradeço também aos meus familiares, pelo incentivo e pela confiança ao longo dessa jornada. À minha namorada, Roberta, agradeço pelo amor, pelo apoio e pelo conforto emocional, especialmente nesta fase final do curso, em que tantas vezes me senti mais perdido. Seu apoio foi imprescindível.

Resumo do Projeto de Graduação apresentado à Escola Politécnica/UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro Mecânico

ESTIMATIVA DE CURVATURA EM SUPERFÍCIES VIA AJUSTE QUADRÁTICO LOCAL E ANÁLISE COMPARATIVA

Breno Mota Sarmiento

Maio/2026

Orientadores: Gustavo Rabello dos Anjos

Daniel Barbedo Vasconcelos Santos

Programa: Engenharia Mecânica

A estimativa precisa da curvatura média de interfaces contínuas é essencial em simulações numéricas de escoamentos bifásicos, nas quais as forças de tensão superficial dependem diretamente da curvatura da interface. Neste trabalho, a interface é discretizada por malhas superficiais, sendo consideradas malhas triangulares nos experimentos, sobre as quais são aplicados os métodos numéricos de estimativa de curvatura. Este trabalho apresenta a implementação de um método de Ajuste Quadrático Local de Curvatura (AQFC) e sua análise comparativa com o operador Laplace–Beltrami discretizado via Método dos Elementos Finitos (FEM) para o cálculo da curvatura média em superfícies discretizadas. Geometrias canônicas, discretizadas com diferentes níveis de refinamento, foram utilizadas como casos de teste, com validação frente a soluções analíticas. Investigaram-se os efeitos do tamanho da vizinhança no AQFC, da regularidade da malha e da sensibilidade dos métodos a discretizações irregulares. Os resultados indicam que o AQFC apresenta maior robustez e estabilidade em malhas irregulares, enquanto o operador Laplace–Beltrami discretizado via FEM mantém elevada eficiência computacional em malhas bem estruturadas.

Palavras-chave: Estimativa de Curvatura; Curvatura Média; Ajuste Quadrático Local; Operador Laplace–Beltrami; Superfícies Discretas.

Abstract of Undergraduate Project presented to POLI/UFRJ as a partial fulfillment of the requirements for the degree of Mechanical Engineer

CURVATURE ESTIMATION ON SURFACES VIA LOCAL QUADRATIC FITTING AND COMPARATIVE ANALYSIS

Breno Mota Sarmiento

May/2026

Advisors: Gustavo Rabello dos Anjos

Daniel Barbedo Vasconcelos Santos

Department: Mechanical Engineering

Accurate estimation of the mean curvature of continuous interfaces is essential in numerical simulations of two-phase flows, in which surface tension forces depend directly on the interface curvature. In this work, the interface is discretized by surface meshes, with triangular meshes being considered in the experiments, on which numerical curvature estimation methods are applied. This study presents the implementation of an Algebraic Quadratic Fitting Curvature (AQFC) method and its comparative analysis with the Laplace–Beltrami operator discretized via the Finite Element Method (FEM) for computing mean curvature on discretized surfaces. Canonical geometries, discretized with different levels of refinement, were used as test cases, with validation against analytical solutions. The effects of neighborhood size in AQFC, mesh regularity, and the sensitivity of the methods to irregular discretizations were investigated. The results indicate that AQFC exhibits greater robustness and stability on irregular meshes, whereas the Laplace–Beltrami operator discretized via FEM maintains high computational efficiency on well-structured meshes.

Keywords: Curvature Estimation; Mean Curvature; Local Quadratic Fitting; Laplace–Beltrami Operator; Discrete Surfaces.

Sumário

Lista de Figuras	xiii
Lista de Tabelas	xvii
Lista de Símbolos	xix
Lista de Abreviaturas	xxi
1 Introdução	1
1.1 Contextualização e Motivação	1
1.2 Objetivos	5
1.3 Metodologia	6
1.4 Organização do Trabalho	8
2 Fundamentação Teórica e Revisão Bibliográfica	9
2.1 Fundamentos da Curvatura em Geometria Diferencial	9
2.1.1 Definição de Curvatura em Curvas e Superfícies	10
2.1.2 Representação Discreta da Curvatura	11
2.2 Conceitos Fundamentais	12
2.2.1 Vizinhança de um Nó	12
2.2.2 Vetores Normais	13
2.2.3 Áreas Associadas aos Nós da Malha	15
2.3 Estudos Relacionados: Métodos de Estimativa de Curvatura em Ma-	
lhas Poligonais	20
2.3.1 Métodos Discretos para Estimativa de Curvatura	20
2.3.2 Métodos Baseados em Ajuste Local e Reconstrução de Su-	
perfícies	23

2.3.3	Comparação entre Métodos, Robustez e Sensibilidade à Vizinhança	25
2.3.4	Qualidade de Malha e Influência da Discretização	28
3	Metodologia	30
3.1	Caracterização da Pesquisa	30
3.2	Configuração experimental	31
3.3	Métodos Utilizados	33
3.3.1	Ajuste Quadrático Local de Curvatura (AQFC)	33
3.3.2	Laplace-Beltrami (LB)	36
3.4	Procedimentos Metodológicos	40
3.4.1	Pré-processamento: Cálculo de Vizinhanças e Normais	40
3.4.2	Ajuste Quadrático Local de Curvatura (AQFC)	44
3.4.3	Operador Laplace-Beltrami Discreto (LB)	53
3.5	Curvatura Analítica de Referência	58
3.6	Métricas de Erro	60
3.7	Métricas de Qualidade da Malha	62
4	Resultados e Discussões	67
4.1	Configuração dos Testes e Métricas de Avaliação	67
4.2	Desempenho Computacional	69
4.3	Acurácia Global: Erro Relativo na Norma L2	71
4.4	Acurácia Local: Erro Relativo Máximo	73
4.5	Erro Relativo Médio	75
4.6	Exemplos Visuais e Padrões Espaciais do Erro	76
4.6.1	Curvaturas	76
4.6.2	Comparação de Malhas: Elipsoide 1 x Elipsoide 2	79
4.6.3	Relação entre a Qualidade da Malha e os Erros	83
4.7	Estudos de Sensibilidade	89
4.7.1	Sensibilidade ao Tamanho da Vizinhança no AQFC	89
4.7.2	Sensibilidade à Irregularidade da Malha	91
4.7.3	Influência do Refinamento da Malha	92
4.7.4	Interação entre Refinamento, Vizinhança e Qualidade da Malha	94

5	Conclusões e Trabalhos Futuros	96
5.1	Síntese das Contribuições	96
5.2	Trabalhos Futuros	98
	Referências Bibliográficas	101
A	Códigos Utilizados	106
A.1	Curvatura por Laplace-Beltrami	106
A.2	Curvatura por Ajuste Quadrático Local	114
A.3	Curvaturas Analíticas	124
B	Dados Coletados	130
B.1	ESFERA	130
B.1.1	Malhas - Info	130
B.1.2	Erro Médio - Norma L2	130
B.1.3	Erro Relativo Médio	131
B.1.4	Erro Relativo Máximo	131
B.2	ELIPSOIDE 1	132
B.2.1	Malhas - Info	132
B.2.2	Erro Médio - Norma L2	132
B.2.3	Erro Relativo Médio	132
B.2.4	Erro Relativo Máximo	133
B.3	ELIPSOIDE 2	134
B.3.1	Malhas - Info	134
B.3.2	Erro Médio - Norma L2	134
B.3.3	Erro Relativo Médio	134
B.3.4	Erro Relativo Máximo	135
B.4	TORO	136
B.4.1	Malhas - Info	136
B.4.2	Erro Médio - Norma L2	136
B.4.3	Erro Relativo Médio	136
B.4.4	Erro Relativo Máximo	137
B.5	CILINDRO	138
B.5.1	Malhas - Info	138

B.5.2	Erro Médio - Norma L2	138
B.5.3	Erro Relativo Médio	139
B.5.4	Erro Relativo Máximo	139

Lista de Figuras

1.1	Padrões de escoamento bifásico em tubulação horizontal	2
1.2	Correntes parasitas em simulação de gota estática	3
1.3	Representação discreta de uma superfície contínua	4
2.1	Comparação conceitual entre a curvatura em 2D e as curvaturas principais em 3D. (a) Em uma curva bidimensional, a curvatura é determinada pelo círculo osculador, o círculo que melhor aproxima a curva no ponto P . (b) Em uma superfície tridimensional, a curvatura é descrita a partir do plano tangente no ponto P , e as direções de curvatura máxima e mínima são ortogonais. Fonte: Adaptado de Gauthier et al. (2017) [1].	11
2.2	Representação da fórmula de Frenet para curvatura em duas dimensões: (a) descrição contínua baseada na variação do vetor tangente ao longo do comprimento de arco; (b) aproximação discreta utilizada em malhas computacionais. Fonte: Adaptado de Anjos (2012) [2].	12
2.3	Representação da vizinhança para o nó i em uma malha no domínio Ω_s	14
2.4	Avaliação do vetor normal em espaços tridimensionais. (a) O vetor normal de cada triângulo na superfície pode ser encontrado aplicando o produto vetorial de dois vetores tangentes que estão no plano do triângulo. (b) O vetor normal nodal final n_i é encontrado somando os vetores normais n_e para $e = \{1..j\}$. No caso ilustrado, $j = 5$. Fonte: Adaptado de Anjos (2012) [2].	15
2.5	Triângulo $\overline{IJK}$ inscrito em uma circunferência de centro C , mostrando os vetores tangentes $\mathbf{t}_1$ e $\mathbf{t}_2$ e os pontos médios $\mathbf{m}_1$ e $\mathbf{m}_2$	16

2.6	Configuração geométrica local associada à aresta (i, j) utilizada no cálculo da área de Voronoi discreta. Os ângulos α_{ij} e β_{ij} correspondem aos ângulos opostos à aresta (i, j) nos dois triângulos adjacentes. O polígono tracejado representa o anel de 1-vizinhança do vértice i .	
	Fonte: Adaptado de Meyer (2003) [3].	18
2.7	Triângulo obtuso IKJ e seu circuncentro C , marcado com um ponto vermelho fora dos limites do triângulo. A área de Voronoi para este triângulo excede a área total do triângulo quando calculada para todos os três vértices. A área calculada usando as arestas conectadas ao ângulo obtuso causa a distorção.	
	Fonte: Adaptado de Santos (2024) [4].	19
3.1	Geometrias Analisadas	32
3.2	Comparação entre dois métodos de criação para as malhas do elipsoide	32
3.3	Raio inscrito r e circunscrito R de um triângulo (esquema ilustrativo).	63
3.4	Comparação entre elementos equiláteros e elementos distorcidos. . . .	63
3.5	Exemplos de diferentes definições de razões de aspecto em triângulos.	64
4.1	CILINDRO: Percentual de Pontos Ignorados x Malha h	69
4.2	Tempo de execução em função de h para cada geometria analisada (legenda comum aos métodos).	70
4.3	Erro relativo da norma L_2 em função de h para cada geometria analisada (legenda comum aos métodos).	72
4.4	Erro relativo máximo em função de h para cada geometria analisada (legenda comum aos métodos).	74
4.5	Erro relativo médio em função de h para cada geometria analisada (legenda comum aos métodos).	75
4.6	Curvatura da esfera na malha com $h = 0.12$, comparando lado a lado: (i) a curvatura média analítica, (ii) a curvatura média estimada pelo operador Laplace–Beltrami e (iii) a curvatura média estimada pelo método AQFC com dois anéis de vizinhança.	76

4.7	Curvatura do elipsoide 1 na malha com $h = 0.12$, comparando lado a lado: (i) a curvatura média analítica, (ii) a curvatura média estimada pelo operador Laplace–Beltrami e (iii) a curvatura média estimada pelo método AQFC com dois anéis de vizinhança.	77
4.8	Curvatura do elipsoide 2 na malha com $h = 0.25$, comparando lado a lado: (i) a curvatura média analítica, (ii) a curvatura média estimada pelo operador Laplace–Beltrami e (iii) a curvatura média estimada pelo método AQFC com dois anéis de vizinhança.	77
4.9	Curvatura do toro na malha com $h = 0.12$, comparando lado a lado: (i) a curvatura média analítica, (ii) a curvatura média estimada pelo operador Laplace–Beltrami e (iii) a curvatura média estimada pelo método AQFC com dois anéis de vizinhança.	78
4.10	Curvatura do cilindro na malha com $h = 0.12$, comparando lado a lado: (i) a curvatura média analítica, (ii) a curvatura média estimada pelo operador Laplace–Beltrami e (iii) a curvatura média estimada pelo método AQFC com dois anéis de vizinhança.	78
4.11	Malha do elipsoide 1 com $h = 0.30$. Métricas de qualidade da malha plotadas lado a lado: razão de aspecto (valor ideal = 1.0), razão r/R (valor ideal = 0.5), distorção angular (valor ideal = 0.0) e qualidade composta (valor ideal = 1.0), respectivamente.	79
4.12	Malha do elipsoide 2 com $h = 0.17$. Métricas de qualidade da malha plotadas lado a lado: razão de aspecto (valor ideal = 1.0), razão r/R (valor ideal = 0.5), distorção angular (valor ideal = 0.0) e qualidade composta (valor ideal = 1.0), respectivamente.	80
4.13	Malha do elipsoide 1 com $h = 0.07$: comparação da mesma região ampliada, apresentando a razão r/R (valor ideal = 0.5) e o erro absoluto da curvatura média calculada pelo método LB.	83
4.14	Malha do elipsoide 1 com $h = 0.04$: comparação entre os mapas de erro relativo percentual da curvatura média (LB, AQFC1 e AQFC3) e a distorção angular (valor ideal = 0.0).	84

4.15	Malha do elipsoide 2 com $h = 0.45$: comparação entre os mapas de erro relativo percentual da curvatura média (LB, AQFC1 e AQFC3) e a distorção angular (valor ideal = 0.0).	84
4.16	Malha do elipsoide 1 com $h = 0.30$: comparação entre os mapas de erro absoluto da curvatura média (LB e AQFC com 1, 2 e 3 vizinhanças).	85
4.17	Malha do elipsoide 2 com $h = 0.17$: comparação entre os mapas de erro relativo percentual da curvatura média (LB e AQFC com 1, 2 e 3 vizinhanças).	86
4.18	Malha da esfera com $h = 0.07$: comparação ampliada entre os mapas de erro absoluto da curvatura média para os métodos LB e AQFC (1, 2 e 3 vizinhanças). A escala de cores (barra lateral) representa a magnitude do erro absoluto, evidenciando a diminuição dos valores à medida que a vizinhança é ampliada.	87
4.19	Malha do cilindro com $h = 0.12$: comparação entre os mapas de erro absoluto da curvatura média para os métodos LB e AQFC com 2 vizinhanças.	88

Lista de Tabelas

4.1	Estatísticas globais das métricas de qualidade para a malha do elipsoide 1 com $h = 0.30$. Reportam-se média, desvio-padrão (DP), mínimo, percentis (P5, P50/mediana, P95) e máximo.	81
4.2	Estatísticas globais das métricas de qualidade para a malha do elipsoide 2 com $h = 0.17$. Reportam-se média, desvio-padrão (DP), mínimo, percentis (P5, P50/mediana, P95) e máximo.	81
B.1	Dados das Malhas da Esfera	130
B.2	Erro Médio - Norma L2, Esfera	130
B.3	Erro Relativo Médio, Esfera	131
B.4	Máximo Erro Relativo, Esfera	131
B.5	Dados das Malhas do Elipsoide 1	132
B.6	Erro Médio - Norma L2, Elipsoide 1	132
B.7	Erro Relativo Médio, Elipsoide 1	132
B.8	Máximo Erro Relativo, Elipsoide 1	133
B.9	Dados das Malhas do Elipsoide 2	134
B.10	Erro Médio - Norma L2, Elipsoide 2	134
B.11	Erro Relativo Médio, Elipsoide 2	134
B.12	Máximo Erro Relativo, Elipsoide 2	135
B.13	Dados das Malhas do Toro	136
B.14	Erro Médio - Norma L2, Toro	136
B.15	Erro Relativo Médio, Toro	136
B.16	Máximo Erro Relativo, Toro	137
B.17	Dados das Malhas do Cilindro	138

B.18 Percentual de Zeros e Pontos Ignorados (NaNs) por Método nas	
Métricas Globais	138
B.19 Erro Médio - Norma L2, Cilindro	138
B.20 Erro Relativo Médio, Cilindro	139
B.21 Máximo Erro Relativo, Cilindro	139

Lista de Símbolos

AR	Razão de Aspecto, p. 64
A_e	área da face triangular e , p. 14
A_{Mista}	área mista, p. 19
A_{Voronoi}	área de Voronoi, p. 17
A_{circ}	área circuncêntrica, p. 17
C	circuncentro, p. 16
H	curvatura média, p. 10
H_q	matriz Hessiana, p. 35
IEN	matriz de conectividade, p. 53
K	curvatura gaussiana, p. 10
N	vizinhança, p. 33
Q	Qualidade composta, p. 66
Q_{norm}	Qualidade composta normalizada, p. 66
Ω_s	domínio discreto (malha), p. 13
$\kappa_{a,i}$	curvatura analítica no ponto i , p. 61
$\kappa_{n,i}$	curvatura numérica no ponto i , p. 61
$\tilde{\mathbf{A}}_t$	vetor área da face triangular, p. 15
$\mathbf{e}$	Índice de elementos/faces triangulares, p. 14

$\mathbf{r}_u, \mathbf{r}_v$	vetores tangentes à superfície, p. 14
$\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k$	vértices da face triangular, p. 15
$\mathbf{A}_t$	área escalar da face triangular, p. 15
ρ	razão incentro/circuncentro, p. 62
θ	ângulo, p. 63
a_{ij}	coeficientes da função objetivo, p. 34
e_{L2}	erro médio norma-L2, p. 60
$e_{\max}$	erro relativo máximo, p. 61
e_{mean}	erro relativo médio, p. 60
$q(x, y, z)$	função quadrática, p. 34
s	semi-perímetro, p. 62
w_i	Pesos da métrica composta, p. 66
$Z(q)$	superfície implícita, p. 34
$\omega_{V_i}, \omega_{n_i}$	pesos da função objetivo, p. 34
h	comprimento médio das arestas, p. 31

Lista de Abreviaturas

AQFC	Algebraic Quadric Fitting Curvature, p. 5
CSF	Continuum Surface Force, p. 2
FEM	Método dos Elementos Finitos, p. 36
LB	Laplace–Beltrami, p. 4
LP	programação linear, p. 49
QP	programação quadrática, p. 49
SLSQP	Sequential Least Squares Programming, p. 49
VTK	Visualization Toolkit, p. 31

Capítulo 1

Introdução

1.1 Contextualização e Motivação

A estimativa de curvatura é um problema recorrente em aplicações computacionais que envolvem superfícies discretizadas. Em problemas de engenharia e ciências aplicadas, superfícies podem representar fronteiras físicas, interfaces entre materiais, geometrias de componentes mecânicos ou interfaces entre fluidos imiscíveis. Nesses casos, a curvatura constitui uma grandeza geométrica fundamental para a caracterização local da superfície e pode influenciar diretamente modelos físicos, critérios numéricos e análises de desempenho.

Uma das motivações práticas para o estudo da curvatura em superfícies discretizadas está associada à simulação de escoamentos multifásicos. Nesses escoamentos, a interface entre fluidos imiscíveis pode assumir diferentes morfologias, como gotas, bolhas, filmes líquidos e estruturas interfaciais mais complexas, conforme ilustrado na Figura 1.1. Essas diferentes configurações geométricas não são apenas uma característica visual do escoamento, mas estão diretamente relacionadas aos mecanismos físicos atuantes na interface.

Em particular, a tensão superficial estabelece uma relação entre a geometria local da interface e o equilíbrio mecânico entre as fases. Essa relação é classicamente descrita pela equação de Young–Laplace, segundo a qual o salto de pressão através de uma interface curva depende da tensão superficial e da curvatura local da superfície. Assim, regiões com diferentes curvaturas podem estar associadas a diferentes contribuições capilares, influenciando a dinâmica e a estabilidade de gotas, bolhas e

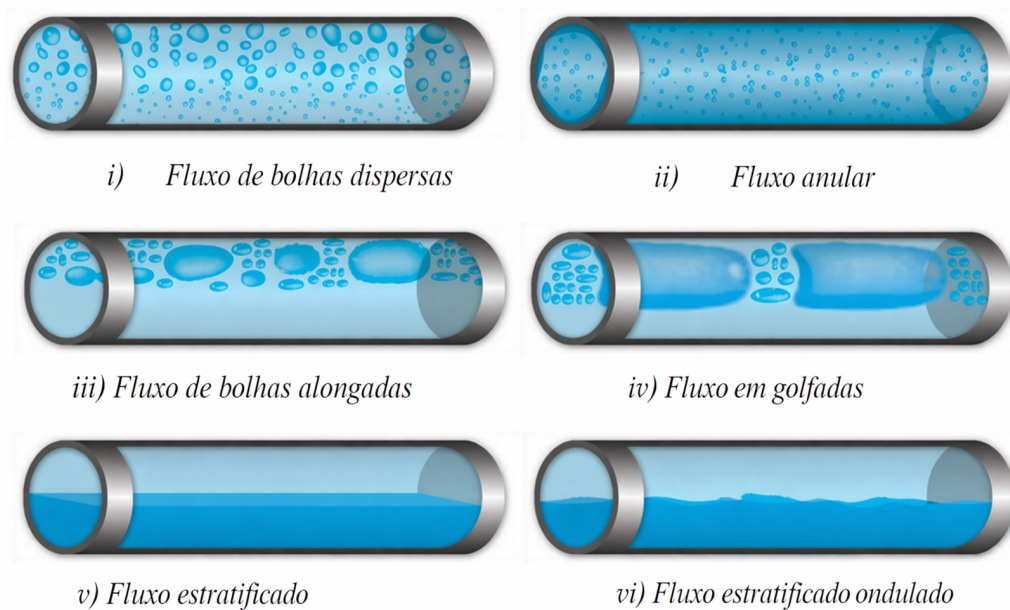


Figura 1.1: Padrões de escoamento bifásico em tubulação horizontal, ilustrando a diversidade de morfologias que a interface gás-líquido pode assumir. Os padrões variam desde bolhas dispersas (i) e alongadas (iii) até regimes mais complexos como o escoamento em golfadas (*slug flow*, iv) e o anular (ii). A forma e a dinâmica dessas interfaces resultam do balanço entre efeitos inerciais, gravitacionais, viscosos e capilares, entre outros fatores.

Fonte: Adaptado de Bratland (2010) [5].

outras estruturas interfaciais [6].

Essa relação entre curvatura e tensão superficial também aparece em formulações numéricas para escoamentos multifásicos. Um exemplo clássico é o modelo de Força de Superfície Contínua, ou *Continuum Surface Force* (CSF), proposto por Brackbill et al. [7]. Nessa formulação, os efeitos da tensão superficial são representados por uma força volumétrica distribuída na região da interface, cuja definição depende da curvatura local da superfície. Assim, a curvatura atua como uma grandeza geométrica de ligação entre a forma da interface e a representação numérica dos efeitos interfaciais.

A relevância dessa representação pode ser observada em estudos sobre correntes parasitas em simulações multifásicas. Esses campos de velocidade não físicos podem surgir nas proximidades da interface em formulações numéricas de tensão superficial, comprometendo a estabilidade e a confiabilidade da solução [8], conforme ilustrado na Figura 1.2.

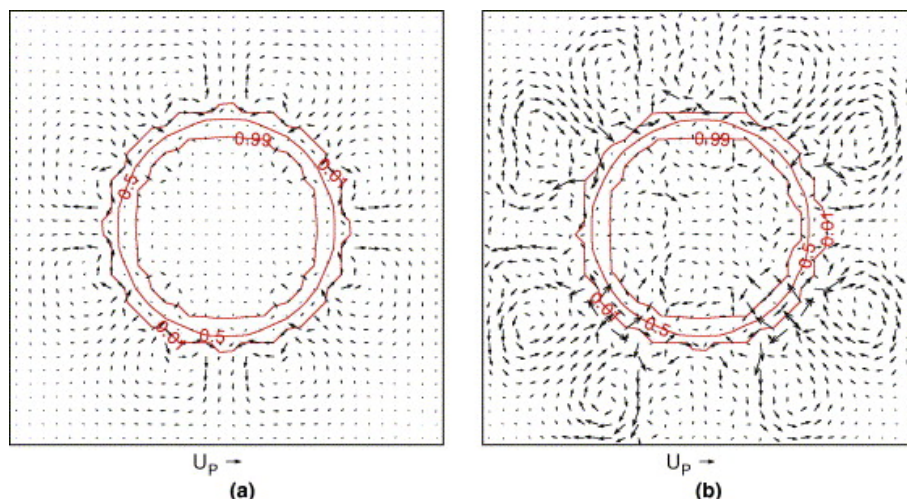


Figura 1.2: Visualização de correntes parasitas em uma simulação de gota estática bidimensional. Os vetores de velocidade (setas) representam o fluxo espúrio gerado numericamente. Em (a), a magnitude das correntes parasitas é limitada predominantemente pelo termo viscoso, enquanto em (b), é limitada pelo termo de advecção inercial. Em ambos os casos, as linhas de contorno representam a fração de volume constante (ϕ), delimitando a interface da gota. Esses fluxos não físicos evidenciam a sensibilidade de simulações multifásicas à representação numérica dos efeitos interfaciais.

Fonte: Adaptado de Harvie (2006) [8].

Quando uma superfície contínua é representada computacionalmente, sua geometria passa a ser aproximada por uma malha composta por vértices, arestas e elementos, conforme ilustrado na Figura 1.3. Em simulações tridimensionais, é comum que superfícies sejam representadas por malhas compostas por elementos planos, como triângulos, que aproximam localmente porções da geometria. Nessa representação discreta, a curvatura não é obtida diretamente a partir de uma superfície analítica contínua, mas precisa ser estimada numericamente a partir da posição dos vértices e da conectividade da malha.

Para avaliar métodos de estimativa de curvatura de forma controlada, é conveniente utilizar geometrias de referência cuja curvatura seja conhecida analiticamente. Superfícies como esfera, cilindro, toro e elipsoide não reproduzem a complexidade geométrica e dinâmica de interfaces reais em escoamentos multifásicos, mas permitem comparar diretamente as estimativas numéricas com valores de referência. Por esse motivo, tais geometrias constituem casos de teste adequados para validar imple-

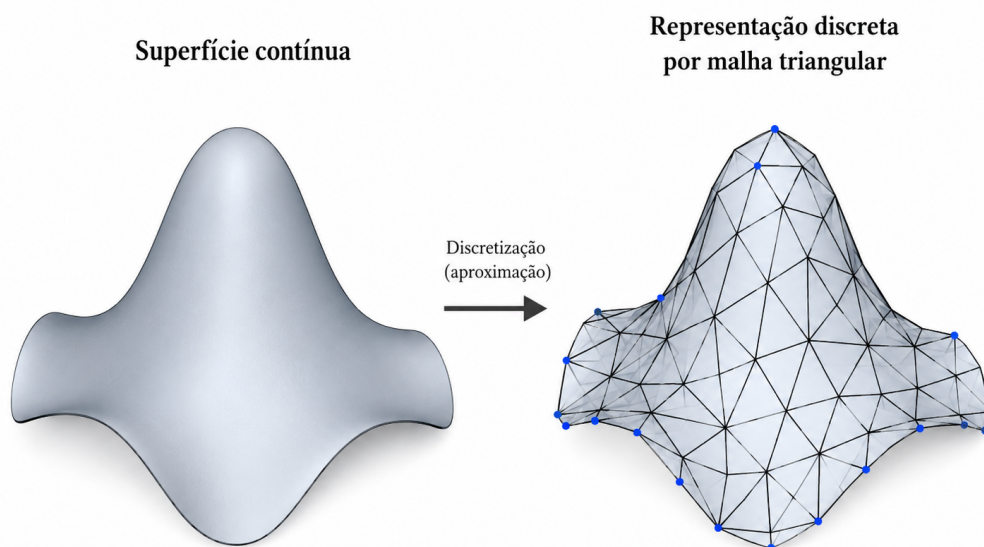


Figura 1.3: Representação esquemática de uma superfície contínua e de sua aproximação por uma malha triangular. Na representação discreta, a geometria da superfície passa a ser descrita por vértices, arestas e elementos planos, de modo que propriedades geométricas como a curvatura precisam ser estimadas numericamente a partir da malha.

mentações, quantificar erros e investigar a influência da discretização da superfície.

Entre as abordagens utilizadas para estimar curvatura em superfícies discretizadas, destaca-se a formulação baseada no **operador Laplace–Beltrami (LB)**. Em termos gerais, esse operador permite relacionar a geometria de uma superfície à sua curvatura média, sendo frequentemente empregado em métodos discretos para análise geométrica de malhas. No presente trabalho, foi utilizada a formulação do operador Laplace–Beltrami discretizada por elementos finitos, conforme apresentada em [4].

Apesar de possuir fundamentação matemática sólida, a aplicação do operador Laplace–Beltrami em superfícies discretizadas pode apresentar sensibilidade à qualidade da malha, especialmente em situações envolvendo elementos irregulares ou distribuições não uniformes de vértices.

Essa limitação motiva a investigação de métodos alternativos para estimativa

de curvatura em superfícies discretizadas. Nesse contexto, destaca-se o método de Ajuste Quadrático Local, conhecido como **Algebraic Quadric Fitting Curvature (AQFC)**, proposto por Makovník e Chalmoviansky [9]. Nesse método, a geometria da superfície em torno de cada vértice é aproximada por uma superfície quadrática ajustada a um conjunto de pontos vizinhos da malha. Esse conjunto, denominado vizinhança do vértice, é apresentado em maior detalhe na Seção 2.2.1. A curvatura é então obtida a partir das propriedades diferenciais da superfície ajustada.

Diante desses aspectos, este trabalho concentra-se na implementação computacional e na avaliação do método AQFC para estimativa de curvatura média em superfícies discretizadas, particularizadas neste estudo por malhas triangulares. A análise busca investigar como a escolha da vizinhança utilizada no ajuste quadrático local, o nível de refinamento da malha e a qualidade geométrica dos elementos influenciam a precisão das estimativas. Para contextualizar o desempenho do AQFC, os resultados são comparados com aqueles obtidos por uma formulação baseada no operador Laplace–Beltrami. Dessa forma, o trabalho busca não apenas comparar os métodos considerados, mas também compreender o comportamento de cada abordagem diante de diferentes condições de discretização.

1.2 Objetivos

O objetivo principal deste trabalho é **implementar e avaliar computacionalmente o método de Ajuste Quadrático Algébrico** (*Algebraic Quadric Fitting Curvature* – AQFC) para estimativa de curvatura média em malhas triangulares, investigando sua sensibilidade à vizinhança de ajuste, ao refinamento e à qualidade da malha. Adicionalmente, os resultados são comparados com aqueles obtidos por uma formulação discreta baseada no operador Laplace–Beltrami (LB).

Para atingir esse objetivo geral, definem-se os seguintes objetivos específicos:

1. **Implementar o método AQFC em malhas triangulares:** desenvolver, em Python, uma implementação do método de ajuste quadrático local para estimativa da curvatura média em vértices de malhas triangulares.
2. **Investigar a influência da vizinhança no AQFC:** avaliar o efeito da ampliação da vizinhança utilizada no ajuste quadrático, considerando vizinhanças

de um, dois e três anéis de vértices adjacentes, de modo a analisar como essa escolha afeta a precisão das estimativas.

3. **Validar as estimativas em geometrias de referência:** utilizar superfícies com curvatura analítica conhecida, como esfera, cilindro, toro e elipsoide, para comparar os valores estimados numericamente com valores analíticos de referência.
4. **Comparar o AQFC com o método baseado no operador Laplace–Beltrami:** realizar uma análise quantitativa entre os erros produzidos pelos dois métodos nas geometrias avaliadas.
5. **Analisar o efeito do refinamento da malha:** investigar o comportamento dos erros numéricos em diferentes níveis de discretização, com o objetivo de verificar o desempenho e consistência das estimativas.
6. **Avaliar a influência da qualidade da malha:** examinar a sensibilidade dos métodos à regularidade geométrica dos elementos triangulares, incluindo o estudo comparativo entre diferentes configurações de malha para o elipsoide.

Como desdobramento exploratório, foram também realizados testes preliminares envolvendo uma estratégia híbrida baseada em métricas locais de qualidade da malha. Contudo, essa abordagem não compõe o escopo central do trabalho, sendo tratada apenas como possibilidade de continuidade futura.

1.3 Metodologia

A metodologia adotada neste trabalho baseia-se na implementação computacional e na avaliação comparativa de dois métodos de estimativa de curvatura média em malhas triangulares: o método de Ajuste Quadrático Algébrico (*Algebraic Quadric Fitting Curvature* – AQFC), proposto por Makovník e Chalmovianský [9], e a discretização do operador Laplace–Beltrami por elementos finitos, conforme apresentada por Santos [4].

Inicialmente, foram selecionadas geometrias analíticas com curvatura conhecida, de modo a permitir a validação e análise da precisão dos resultados numéricos. As

superfícies consideradas foram esfera, cilindro, toro e elipsoide. Para cada geometria, foram geradas malhas triangulares com diferentes níveis de refinamento por meio do software *Gmsh*. No caso do elipsoide, foram consideradas duas famílias de malha: uma configuração padrão e uma configuração mais regular, construída com o objetivo de investigar de forma mais direta a influência da qualidade dos elementos na estimativa de curvatura.

Em seguida, foi implementado em Python o método AQFC proposto por Makovník e Chalmovianský [9], no qual a geometria local da superfície é aproximada por uma quádrlica ajustada a uma vizinhança de pontos ao redor de cada vértice. Além da formulação original baseada na vizinhança imediata, investigou-se também a influência da ampliação da região local de ajuste, por meio do uso de vizinhanças de um, dois e três anéis de adjacência.

Para fins de comparação, foi utilizada a estimativa de curvatura baseada na discretização do operador Laplace–Beltrami pelo método dos elementos finitos, conforme a formulação empregada por Santos [4]. As implementações computacionais utilizaram bibliotecas científicas da linguagem Python, com destaque para NumPy e SciPy, e os resultados foram exportados em formato VTK para pós-processamento e visualização no software ParaView.

A validação dos métodos foi realizada por comparação direta entre os valores numéricos estimados e as soluções analíticas de referência. Para quantificar os desvios observados, foram empregadas três métricas de erro, incluindo erro relativo na norma L_2 , erro relativo médio e erro relativo máximo.

Além da análise global de erro, foram também avaliadas métricas de qualidade da malha com o objetivo de investigar a relação entre distorção geométrica local e precisão das estimativas de curvatura. Essa análise permitiu examinar, de forma sistemática, não apenas o efeito do refinamento, mas também o impacto da regularidade dos elementos triangulares sobre o desempenho de cada método.

1.4 Organização do Trabalho

Este trabalho está organizado em cinco capítulos, distribuídos da seguinte forma:

- **Capítulo 1 – Introdução:** apresenta a motivação do trabalho, a contextualização do problema, os objetivos da pesquisa e a visão geral da metodologia adotada.
- **Capítulo 2 – Fundamentação Teórica e Revisão Bibliográfica:** apresenta os conceitos fundamentais de curvatura em geometria diferencial e discute os principais métodos de estimativa de curvatura em malhas triangulares reportados na literatura, com ênfase em métodos discretos e em abordagens baseadas em ajuste local da superfície.
- **Capítulo 3 – Metodologia:** descreve a caracterização da pesquisa, a configuração experimental, as geometrias analisadas, a geração das malhas, os métodos implementados, as métricas de erro e os procedimentos utilizados na comparação numérica.
- **Capítulo 4 – Resultados e Discussões:** apresenta os resultados obtidos para os métodos analisados, discutindo o efeito do refinamento da malha, da ampliação da vizinhança no AQFC e da qualidade geométrica dos elementos sobre a precisão das estimativas.
- **Capítulo 5 – Conclusões e Trabalhos Futuros:** sintetiza as principais conclusões da pesquisa, destaca as contribuições do trabalho e discute possíveis desdobramentos para investigações futuras, incluindo considerações sobre testes exploratórios de uma abordagem híbrida para estimativa de curvatura.

Capítulo 2

Fundamentação Teórica e Revisão Bibliográfica

Este capítulo apresenta os fundamentos teóricos necessários à compreensão da curvatura em superfícies e dos métodos numéricos empregados em sua estimativa. Além da base conceitual em geometria diferencial e processamento geométrico, realiza-se também uma revisão bibliográfica dos principais métodos de estimativa de curvatura em malhas triangulares, com ênfase em formulações discretas e em abordagens baseadas em reconstrução local da superfície.

2.1 Fundamentos da Curvatura em Geometria Diferencial

A curvatura é uma grandeza geométrica fundamental no estudo das propriedades locais de curvas e superfícies. Por propriedades locais, entendem-se aquelas que dependem exclusivamente do comportamento da entidade geométrica em uma vizinhança arbitrariamente pequena de um ponto [10].. Nesse contexto, a geometria diferencial utiliza ferramentas do cálculo diferencial para descrever tais propriedades, representando curvas e superfícies por funções suficientemente regulares.

2.1.1 Definição de Curvatura em Curvas e Superfícies

No caso bidimensional, a curvatura expressa o desvio local de uma curva em relação à sua reta tangente no ponto considerado [10]. Geometricamente, essa interpretação está associada ao **círculo osculador**, definido como o círculo no plano osculador com centro no centro de curvatura da curva no ponto considerado [11].

Matematicamente, a definição da curvatura é formalizada pelas **fórmulas de Frenet**, que relacionam as variações dos vetores tangente $\mathbf{t}$ e normal $\mathbf{n}$ em relação ao comprimento de arco s [11]:

$$\kappa \mathbf{t} = -\frac{d\mathbf{n}}{ds} \quad \text{ou} \quad \kappa \mathbf{n} = \frac{d\mathbf{t}}{ds}. \quad (2.1)$$

Ao considerar uma superfície regular $S \subset \mathbb{R}^3$, a noção de curvatura torna-se direcional. Em cada ponto, define-se o plano tangente e consideram-se as **seções normais**, obtidas pela interseção da superfície com planos que contêm o vetor normal nesse ponto. A curvatura associada a cada uma dessas seções é denominada **curvatura normal** ($\kappa_n = 1/R$), onde R representa o raio de curvatura.

Como diferentes direções no plano tangente geram diferentes seções normais, a curvatura depende da direção considerada. Em particular, existem duas direções ortogonais nas quais a curvatura normal assume valores extremos, denominados **curvaturas principais**, κ_1 e κ_2 .

A partir das curvaturas principais definem-se a *curvatura média*

$$H = \frac{\kappa_1 + \kappa_2}{2}$$

e a *curvatura gaussiana*

$$K = \kappa_1 \kappa_2,$$

que constituem grandezas fundamentais no estudo local das superfícies [10].

A representação geométrica desses conceitos, tanto para curvas planas quanto para superfícies em $\mathbb{R}^3$, é ilustrada na Figura 2.1.

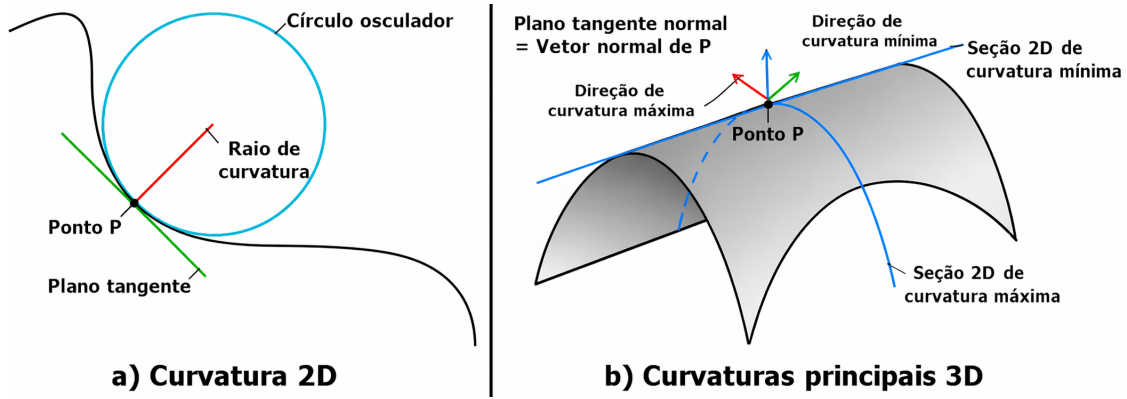


Figura 2.1: Comparação conceitual entre a curvatura em 2D e as curvaturas principais em 3D. (a) Em uma curva bidimensional, a curvatura é determinada pelo círculo osculador, o círculo que melhor aproxima a curva no ponto P . (b) Em uma superfície tridimensional, a curvatura é descrita a partir do plano tangente no ponto P , e as direções de curvatura máxima e mínima são ortogonais.

Fonte: Adaptado de Gauthier et al. (2017) [1].

2.1.2 Representação Discreta da Curvatura

Em aplicações de computação gráfica e engenharia computacional, superfícies são frequentemente representadas por **malhas triangulares**, que constituem aproximações poligonais de superfícies suaves [12].

Uma malha triangular consiste em um conjunto de vértices, arestas e faces triangulares que definem uma superfície contínua por partes [12]. Embora a superfície resultante seja contínua (C^0), ela não é diferenciável ao longo das arestas.

Como a curvatura, no contexto clássico, é definida a partir de derivadas e limites locais, a ausência de uma estrutura diferenciável global na malha impede a aplicação direta dos instrumentos da geometria diferencial. Torna-se necessário, portanto, o desenvolvimento de métodos específicos para a **estimação de curvaturas** em malhas poligonais, tema amplamente investigado na literatura.

Esse processo envolve a **discretização** das expressões diferenciais que definem a curvatura, isto é, a substituição das derivadas contínuas por aproximações baseadas em operadores discretos construídos a partir da conectividade e da geometria local dos elementos da malha. A Figura 2.2 ilustra esse princípio no caso bidimensional, mostrando a passagem da formulação contínua baseada nas fórmulas de Frenet para uma aproximação discreta construída a partir de segmentos consecutivos da

interface.

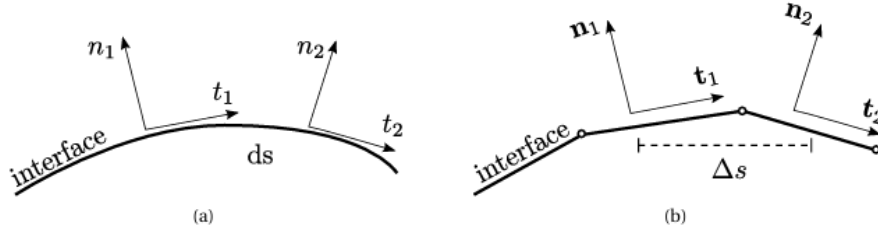


Figura 2.2: Representação da fórmula de Frenet para curvatura em duas dimensões: (a) descrição contínua baseada na variação do vetor tangente ao longo do comprimento de arco; (b) aproximação discreta utilizada em malhas computacionais.

Fonte: Adaptado de Anjos (2012) [2].

Uma possibilidade de discretização da Equação 2.1 no contexto de curvas discretas é dada pela aproximação de primeira ordem

$$\kappa_n = \frac{\mathbf{t}_1 - \mathbf{t}_2}{\Delta s}, \quad (2.2)$$

onde $\mathbf{t}_1$ e $\mathbf{t}_2$ representam aproximações dos vetores tangentes unitários a duas arestas consecutivas da interface, e Δs é uma estimativa do comprimento de arco correspondente [2]. Essa expressão fornece uma forma discreta de quantificar a variação direcional do vetor tangente ao longo da curva poligonal que aproxima a curva suave original.

2.2 Conceitos Fundamentais

Para a implementação e análise dos métodos de estimativa de curvatura (AQFC e LB), é crucial definir e quantificar as propriedades geométricas locais da malha. Os conceitos a seguir são a base para a discretização e o cálculo numérico:

2.2.1 Vizinhança de um Nó

Nos **métodos computacionais aplicados a superfícies**, a vizinhança de um nó desempenha um papel essencial em operações como o *cálculo de curvaturas*, *vetores normais*, *gradientes* e *laplacianos discretos*. Isso ocorre porque essas quantidades

dependem das variações geométricas locais da superfície, que são inferidas a partir da posição relativa dos nós vizinhos.

A **vizinhança de um nó** em uma malha é o conjunto de nós conectados a ele por meio das arestas que compõem os elementos (geralmente triângulos ou polígonos) da malha. Essa estrutura local é fundamental para a descrição da geometria e da topologia da superfície, pois permite estimar propriedades diferenciais e realizar cálculos numéricos de forma discreta.

Para organizar essa estrutura local, definem-se **anéis de vizinhança** em torno de um nó central i . No presente trabalho, essa noção é particularmente importante para o método AQFC, no qual será analisada a influência da ampliação da vizinhança até o terceiro anel na qualidade da estimativa de curvatura, onde:

- O **1º anel de vizinhança** (ou primeira vizinhança) é formado pelos nós conectados diretamente ao nó i por **uma aresta**.
- O **2º anel de vizinhança** inclui os nós que estão a **duas arestas de distância** de i , ou seja, vizinhos dos vizinhos do nó central.
- O **3º anel de vizinhança** contém os nós situados a **três arestas de distância** do nó i .

Esses anéis permitem controlar o **alcance da influência local** em diferentes operações numéricas. Por exemplo, para o cálculo preciso de curvaturas médias ou gaussianas, pode ser necessário considerar vizinhanças mais amplas, enquanto operações mais locais, como o cálculo de normais, geralmente utilizam apenas o primeiro anel. A Figura 2.3 apresenta uma representação visual desses anéis.

2.2.2 Vetores Normais

As **normais** são vetores perpendiculares à superfície em um ponto e representam a orientação local dessa superfície. Em outras palavras, a normal indica a direção "para fora" ou "para dentro" da superfície, sendo um elemento essencial para a caracterização geométrica de modelos tridimensionais.

Em sua **definição contínua**, a normal a uma superfície parametrizada $\mathbf{r}(u, v)$

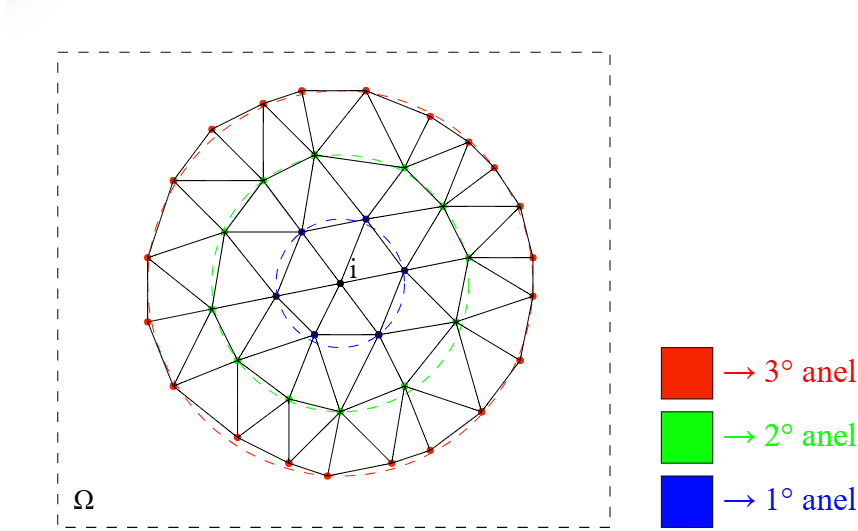


Figura 2.3: Representação da vizinhança para o nó i em uma malha no domínio Ω_s

é dada pelo produto vetorial das derivadas parciais em relação aos parâmetros:

$$\mathbf{n} = \frac{\mathbf{r}_u \times \mathbf{r}_v}{\|\mathbf{r}_u \times \mathbf{r}_v\|} \quad (2.3)$$

onde $\mathbf{r}_u$ e $\mathbf{r}_v$ são os vetores tangentes à superfície. Essa definição é amplamente utilizada em geometria diferencial e fornece a base teórica para o cálculo de curvaturas e outras propriedades locais.

No contexto **discreto**, como em malhas triangulares, a normal em um nó pode ser aproximada a partir das normais das faces adjacentes. Como pode ser observado na figura 2.4, o nó $\mathbf{i}$ tem os elementos vizinhos $\mathbf{e}$ que vão de $\{1..j\}$, j representa o número de elementos vizinhos, e cada triângulo possui uma normal calculada ($\mathbf{n}_e$) pelo produto vetorial de duas de suas arestas adjacentes ao nó de interesse (vetores tangentes à face, $\mathbf{t}_j$ e $\mathbf{t}_k$). A normal do nó ($\mathbf{n}_i$) é obtida, por exemplo, pela média ponderada das normais das faces incidentes:

$$\mathbf{n}_i = \frac{\sum_{e=1}^j A_e \mathbf{n}_e}{\left\| \sum_{e=1}^j A_e \mathbf{n}_e \right\|}; \quad \text{sendo } \mathbf{n}_e = \frac{\mathbf{t}_j \times \mathbf{t}_k}{\|\mathbf{t}_j \times \mathbf{t}_k\|} \quad (2.4)$$

onde A_e é a área da face e e $\mathbf{n}_e$ sua normal unitária.

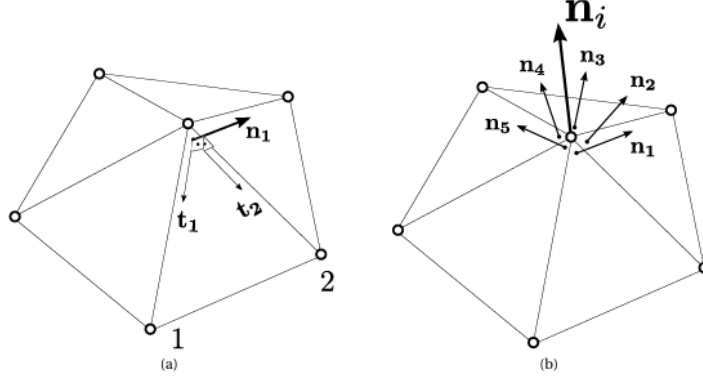


Figura 2.4: Avaliação do vetor normal em espaços tridimensionais. (a) O vetor normal de cada triângulo na superfície pode ser encontrado aplicando o produto vetorial de dois vetores tangentes que estão no plano do triângulo. (b) O vetor normal nodal final n_i é encontrado somando os vetores normais n_e para $e = \{1..j\}$. No caso ilustrado, $j = 5$.

Fonte: Adaptado de Anjos (2012) [2].

2.2.3 Áreas Associadas aos Nós da Malha

Matematicamente, a área associada a um nó i , denotada por A_i , pode ser definida de diferentes maneiras, dependendo do método e do tipo de malha considerada. As formulações mais utilizadas são apresentadas a seguir.

Área Vetorial e Relação com a Normal da Face

Para uma face triangular t definida pelos vértices $\mathbf{v}_i$, $\mathbf{v}_j$ e $\mathbf{v}_k$, o cálculo da área pode ser feito de forma vetorial por meio do produto vetorial entre duas arestas da face. O **vetor área** da face é definido como:

$$\tilde{\mathbf{A}}_t = \frac{1}{2} [(\mathbf{v}_j - \mathbf{v}_i) \times (\mathbf{v}_k - \mathbf{v}_i)] \quad (2.5)$$

esse vetor possui direção perpendicular à face e módulo proporcional à sua área. O **módulo** de $\mathbf{A}_t$ fornece a área escalar da face:

$$A_t = \frac{1}{2} \|(\mathbf{v}_j - \mathbf{v}_i) \times (\mathbf{v}_k - \mathbf{v}_i)\| \quad (2.6)$$

enquanto sua **direção normalizada** define a normal unitária da face:

$$\mathbf{n}_t = \frac{(\mathbf{v}_j - \mathbf{v}_i) \times (\mathbf{v}_k - \mathbf{v}_i)}{\|(\mathbf{v}_j - \mathbf{v}_i) \times (\mathbf{v}_k - \mathbf{v}_i)\|} \quad (2.7)$$

Essa formulação vetorial é amplamente utilizada em computação geométrica por permitir o cálculo simultâneo da área e da orientação da superfície. Além disso, o vetor área $\vec{A}_t$ é usado em diversos algoritmos de integração sobre malhas, no cálculo de curvaturas discretas e na determinação de áreas nodais por médias ponderadas.

Área de Voronoi

A área de Voronoi associada a um nó i em uma malha triangular é construída a partir da subdivisão das faces incidentes a esse nó em regiões menores, de forma que cada ponto da superfície pertença ao vértice mais próximo. Essa subdivisão é obtida por meio dos **circuncentros** dos triângulos, que são os pontos equidistantes dos três vértices de cada face.

Em cada triângulo incidente ao nó i , a região de Voronoi correspondente é delimitada pelos segmentos que conectam o circuncentro aos pontos médios das arestas adjacentes a i . A união dessas pequenas regiões em torno do nó forma o **polígono de Voronoi local**.

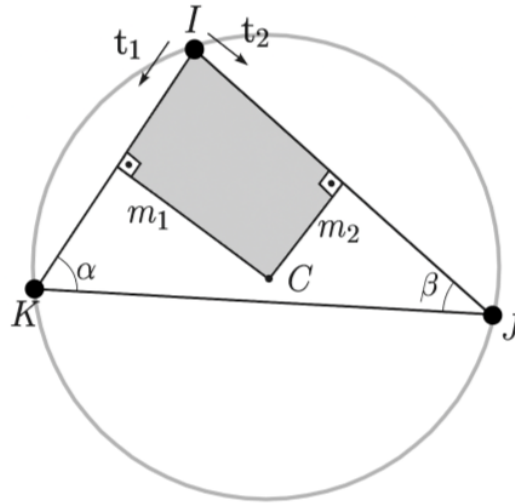


Figura 2.5: Triângulo $\overline{IJK}$ inscrito em uma circunferência de centro C , mostrando os vetores tangentes t_1 e t_2 e os pontos médios m_1 e m_2 .

A **área circuncêntrica** de um triângulo fornece uma forma geométrica direta de calcular a contribuição de cada face para a área de Voronoi de seus vértices. Considerando o triângulo $\overline{IJK}$, com lados opostos de comprimentos a , b e c , o

circuncentro $\mathbf{C}$ é obtido por:

$$\mathbf{C} = \frac{a^2(b^2 + c^2 - a^2)\mathbf{I} + b^2(a^2 + c^2 - b^2)\mathbf{J} + c^2(a^2 + b^2 - c^2)\mathbf{K}}{a^2(b^2 + c^2 - a^2) + b^2(a^2 + c^2 - b^2) + c^2(a^2 + b^2 - c^2)} \quad (2.8)$$

onde $\mathbf{I}$, $\mathbf{J}$ e $\mathbf{K}$ são as posição dos vértices do triângulo. Essa expressão representa uma média ponderada das posições dos vértices, garantindo que $\mathbf{C}$ esteja equidistante de $\mathbf{I}$, $\mathbf{J}$ e $\mathbf{K}$.

A partir do vértice $\mathbf{I}$, definem-se os vetores tangentes às arestas incidentes:

$$\tilde{\mathbf{t}}_1 = \frac{\mathbf{J} - \mathbf{I}}{2}, \quad \tilde{\mathbf{t}}_2 = \frac{\mathbf{K} - \mathbf{I}}{2} \quad (2.9)$$

Os vetores $\tilde{\mathbf{t}}_1$ e $\tilde{\mathbf{t}}_2$ apontam na direção das arestas $\overline{IJ}$ e $\overline{IK}$, respectivamente, e possuem módulo igual à metade do comprimento de cada uma.

Em seguida, são definidos os vetores que conectam o circuncentro $\mathbf{C}$ aos pontos médios das arestas correspondentes:

$$\tilde{\mathbf{m}}_1 = \mathbf{C} - \frac{\mathbf{I} + \mathbf{J}}{2}, \quad \tilde{\mathbf{m}}_2 = \mathbf{C} - \frac{\mathbf{I} + \mathbf{K}}{2} \quad (2.10)$$

Cada vetor $\tilde{\mathbf{m}}_k$ aponta do ponto médio da aresta para o circuncentro.

A **área circuncêntrica associada ao vértice $\mathbf{I}$ dentro do triângulo $\overline{IJK}$** é então dada por:

$$A_{\text{circ}} = \frac{1}{2} (\tilde{\mathbf{t}}_1 \times \tilde{\mathbf{m}}_1 + \tilde{\mathbf{t}}_2 \times \tilde{\mathbf{m}}_2) \quad (2.11)$$

Essa formulação fornece uma forma vetorial e geométrica de determinar a subárea correspondente ao vértice $\mathbf{I}$, preservando a orientação local da malha.

A soma das áreas circuncêntricas associadas a todos os triângulos incidentes a um vértice resulta na **área de Voronoi** desse vértice:

$$A_{\text{Voronoi}} = \sum_{j \in \mathcal{N}_1^i} A_{\text{circ}} \quad (2.12)$$

O uso do circuncentro garante que as fronteiras entre regiões de Voronoi sejam perpendiculares às arestas da malha, assegurando consistência geométrica. Para triângulos *agudos*, o circuncentro está contido na face, e a definição é direta; entretanto, em triângulos *obtusos*, o circuncentro situa-se fora da face, o que motiva o uso da **correção de área para o triângulo obtuso** como correção, conforme proposto por Santos [4].

De acordo com Meyer [3], uma outra forma de calcular a área de Voronoi correspondente ao nó i , pode ser feita diretamente por:

$$A_i = \frac{1}{8} \sum_{j \in N_1(i)} (\cot \alpha_{ij} + \cot \beta_{ij}) \|\mathbf{v}_i - \mathbf{v}_j\|^2 \quad (2.13)$$

onde α_{ij} e β_{ij} são os ângulos opostos à aresta (i, j) nos dois triângulos adjacentes (ver Figura 2.6), e $N_1(i)$ é o conjunto de nós do primeiro anel de vizinhança de i .

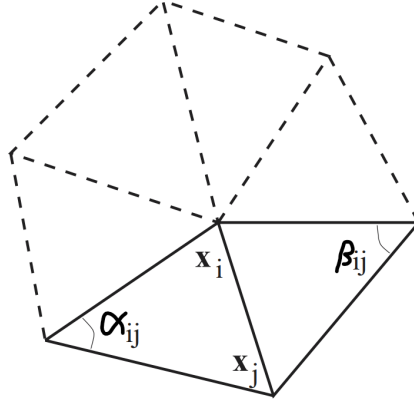


Figura 2.6: Configuração geométrica local associada à aresta (i, j) utilizada no cálculo da área de Voronoi discreta. Os ângulos α_{ij} e β_{ij} correspondem aos ângulos opostos à aresta (i, j) nos dois triângulos adjacentes. O polígono tracejado representa o anel de 1-vizinhança do vértice i .

Fonte: Adaptado de Meyer (2003) [3].

Correção de Área para o Triângulo Obtuso

Para lidar com triângulos obtusos, emprega-se a **correção de área** conforme proposto por Santos [4], que combina a definição de Voronoi com subdivisões triangulares.

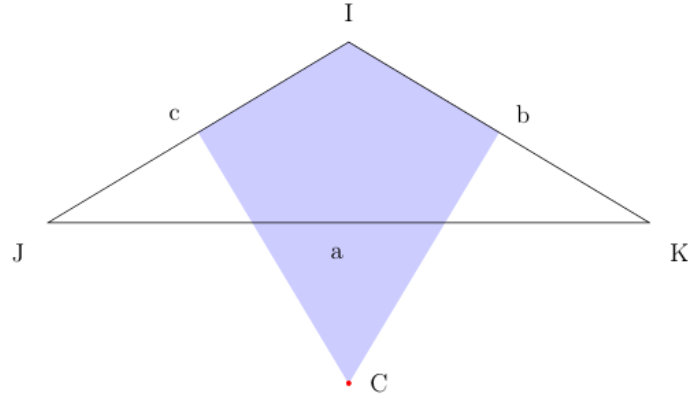


Figura 2.7: Triângulo obtuso IJK e seu circuncentro C , marcado com um ponto vermelho fora dos limites do triângulo. A área de Voronoi para este triângulo excede a área total do triângulo quando calculada para todos os três vértices. A área calculada usando as arestas conectadas ao ângulo obtuso causa a distorção.

Fonte: Adaptado de Santos (2024) [4].

A figura 2.7 exemplifica um triângulo obtuso IJK . Então, para cada triângulo t incidente ao nó i da malha, a contribuição é definida como:

$$A_i^t = \begin{cases} A_{circ}, & \text{se } t \text{ é agudo;} \\ A_{circ}, & \text{para } I = i, \text{ se } t \text{ é obtuso em } I; \\ \frac{c}{b+c}(A_t - A_{circ}), & \text{para } J = i, \text{ se } t \text{ é obtuso em } I; \\ \frac{b}{b+c}(A_t - A_{circ}), & \text{para } K = i, \text{ se } t \text{ é obtuso em } I. \end{cases}$$

A área total associada ao nó i é então obtida por:

$$A_{corrigida} = \sum_{t \in T(i)} A_i^t \quad (2.14)$$

onde $T(i)$ é o conjunto de triângulos incidentes ao nó i .

Essas diferentes definições de área permitem adaptar o modelo discreto às propriedades geométricas da superfície analisada. Em particular, as áreas de Voronoi e mistas são preferidas para o cálculo de operadores diferenciais discretos, enquanto as áreas baricêtricas são mais comuns em aplicações de suavização, renderização e reconstrução de malhas.

2.3 Estudos Relacionados: Métodos de Estimativa de Curvatura em Malhas Poligonais

De modo geral, os métodos de estimativa de curvatura em malhas triangulares podem ser organizados em duas grandes famílias: **métodos discretos**, que aproximam diretamente operadores ou quantidades diferenciais sobre a malha, e **métodos baseados em ajuste local ou reconstrução de superfície**, nos quais se aproxima localmente uma superfície suave para então extrair suas grandezas geométricas. Essa divisão aparece de forma explícita, por exemplo, em Razdan e Bae [13], que distinguem métodos *discretos* e *contínuos*, e em Zhihong et al. [14], que classificam as técnicas em métodos de *ajuste de superfície* ou *discretos*. Em classificações mais detalhadas, contudo, alguns autores refinam essa organização e separam uma terceira classe associada à estimação do tensor de curvatura, como em Gatzke e Grimm [15, 16]. Ainda assim, em uma visão mais ampla, essa terceira linha pode ser entendida como uma especialização dos métodos discretos, pois também busca obter quantidades diferenciais diretamente a partir da estrutura discreta da malha, sem reconstruir explicitamente uma superfície.

2.3.1 Métodos Discretos para Estimativa de Curvatura

Entre os trabalhos fundamentais na linha de métodos discretos, Meyer et al. [3] apresentam uma formulação unificada para a aproximação de quantidades geométricas de primeira e segunda ordem em malhas triangulares arbitrárias, incluindo vetor normal, curvaturas média e Gaussiana, curvaturas principais e direções principais. O desenvolvimento baseia-se em uma combinação entre elementos finitos e volumes finitos, interpretando essas quantidades como médias espaciais locais sobre a vizinhança imediata do vértice. No caso da curvatura média normal $H(x)$, os autores obtêm a conhecida formulação cotangente,

$$H(x_i) \approx \frac{1}{2A_{\text{Mixed}}} \sum_{j \in N_1(i)} (\cot \alpha_{ij} + \cot \beta_{ij})(x_i - x_j),$$

em que α_{ij} e β_{ij} são os ângulos opostos à aresta (x_i, x_j) e A_M é uma área local mista associada ao vértice. Essa expressão é particularmente relevante para o presente trabalho porque mostra que a discretização do operador Laplace–Beltrami depende

explicitamente da geometria local da triangulação, incorporando ângulos e áreas associadas aos vértices, e não apenas sua conectividade. Além disso, Meyer et al. interpretam essa construção como uma quadratura local das curvaturas normais ao longo das arestas do 1-anel, o que reforça a conexão entre o operador discreto e a geometria diferencial da superfície.

No contexto do processamento geométrico, Desbrun et al. [17] mostram que a escolha do operador laplaciano discreto afeta diretamente a estabilidade numérica e a fidelidade geométrica de algoritmos aplicados a malhas irregulares. Em particular, os autores contrastam operadores geometricamente ponderados com discretizações uniformes simples, entre elas o chamado operador *umbrella*, no qual cada vértice é atualizado a partir de uma média uniforme de seus vizinhos imediatos. Trata-se, portanto, de uma discretização essencialmente combinatória, que desconsidera informações geométricas mais refinadas, como ângulos e áreas locais. Segundo os autores, esse tipo de aproximação pode induzir distorções geométricas e comportamento numérico inadequado em malhas irregulares, motivando o emprego de operadores dependentes da escala local e mais consistentes com a geometria da triangulação. Embora o foco do trabalho seja a suavização de malhas, suas conclusões são importantes para a estimativa de curvatura, pois reforçam que o desempenho de operadores discretos depende fortemente da qualidade da malha e da adequação da discretização à geometria local.

Essa limitação estrutural é tratada de forma teórica por Wardetzky et al. [18], que demonstram que não existe um Laplaciano discreto capaz de satisfazer simultaneamente, em malhas triangulares gerais, todas as propriedades desejáveis herdadas do caso contínuo, como simetria, localidade, precisão linear, positividade e convergência. Esse resultado fornece uma interpretação unificadora para a diversidade de operadores propostos na literatura: diferentes discretizações preservam diferentes subconjuntos de propriedades, de acordo com a aplicação visada. Para o presente trabalho, essa análise é especialmente importante porque esclarece que métodos baseados no operador Laplace–Beltrami envolvem compromissos teóricos inevitáveis, o que ajuda a interpretar diferenças de desempenho observadas entre malhas e geometrias distintas.

Em complemento, Xu [19] investiga diversos esquemas de discretização do ope-

rador Laplace–Beltrami sobre superfícies trianguladas e estabelece resultados de convergência sob condições geométricas específicas. O autor mostra que a validade assintótica desses operadores depende não apenas da expressão algébrica adotada, mas também de hipóteses sobre a triangulação e sobre o regime de refinamento considerado. Além disso, os experimentos numéricos indicam que operadores com suporte local mais amplo tendem a apresentar maior robustez diante de perturbações de alta frequência. Esse ponto é particularmente pertinente para este trabalho, pois sugere que o tamanho da vizinhança pode influenciar de maneira decisiva a qualidade da estimativa de curvatura, questão que também motiva a análise de vizinhanças ampliadas no método AQFC.

Uma formulação particularmente relevante nesse contexto é a de Bobenko e Springborn [20], que propõem um operador Laplace–Beltrami discreto para superfícies simpliciais baseado na triangulação intrínseca de Delaunay. Os autores mostram que essa construção depende apenas da geometria intrínseca da superfície e produz pesos positivos, superando duas limitações clássicas da formulação co-tangente em triangulações arbitrárias: a possível ocorrência de pesos negativos e a dependência da triangulação extrínseca adotada. Para a presente revisão, esse trabalho é importante porque evidencia que diferentes discretizações do operador Laplace–Beltrami não se distinguem apenas por custo computacional, mas também por propriedades geométricas fundamentais, como intrinsecidade, positividade e consistência com a métrica da superfície.

Outra vertente relevante entre os métodos discretos concentra-se na estimação local do tensor de curvatura. Taubin [21] propõe um método para estimar, em vértices de superfícies poliédricas, o tensor de curvatura a partir de matrizes simétricas definidas por fórmulas integrais, recuperando curvaturas principais e direções principais por meio de autovalores e autovetores. Além de sua formulação conceitualmente elegante, o método apresenta complexidade linear em tempo e memória, o que contribuiu para sua ampla difusão na literatura. Já Theisel et al. [22] introduzem uma abordagem baseada nas normais dos vértices, na qual cada triângulo da malha é tratado de forma independente para produzir uma função suave do tensor de curvatura no interior da face. Em vez de estimar curvaturas apenas nos vértices, os autores constroem uma estimativa por triângulo a partir da interpolação linear da geometria

e das normais, aproximando localmente a variação da normal sobre a superfície.

Rusinkiewicz [23], por sua vez, propõe um método eficiente por diferenças finitas para estimar curvaturas e derivadas de curvatura em malhas triangulares irregulares. Um dos aspectos mais relevantes do trabalho é a discussão explícita sobre o papel da vizinhança local: vizinhanças pequenas tendem a preservar melhor os detalhes geométricos em dados limpos, enquanto o aumento da vizinhança suaviza as estimativas e reduz sua sensibilidade ao ruído, ao custo de introduzir borramento da curvatura. Essa análise é particularmente importante porque mostra que a escolha da vizinhança não é apenas um detalhe implementacional, mas um fator central na precisão e na robustez dos estimadores.

No contexto do presente trabalho, essa literatura destaca dois pontos centrais. Primeiro, o método baseado em Laplace–Beltrami se insere em uma tradição consolidada de operadores discretos locais, computacionalmente eficientes e geometricamente bem fundamentados. Segundo, a própria literatura mostra que tais operadores e estimadores discretos são sensíveis às propriedades geométricas da malha, ao tipo de discretização adotada e à extensão da vizinhança considerada, o que torna particularmente pertinente compará-los com abordagens baseadas em reconstrução local da superfície, como o AQFC.

2.3.2 Métodos Baseados em Ajuste Local e Reconstrução de Superfícies

A segunda grande família de métodos formula a estimação de curvatura como um problema de reconstrução ou aproximação local da geometria suave subjacente à malha. Nessa perspectiva, a superfície é aproximada, em uma vizinhança do ponto de interesse, por uma representação contínua, como um polinômio, uma superfície quadrática, uma superfície implícita ou um *patch* paramétrico (trecho local de superfície descrito por uma parametrização). As quantidades diferenciais são então calculadas a partir da superfície ajustada, e não diretamente da malha original.

Entre os trabalhos clássicos dessa linha, Hamann [24] propõe a aproximação das curvaturas principais em superfícies trianguladas a partir do ajuste local, em mínimos quadrados, de um polinômio bivariado quadrático. A construção parte da representação local da superfície como o gráfico de uma função bivariada, de modo

que as curvaturas sejam inferidas a partir da superfície ajustada, e não diretamente da malha triangular. A relevância desse trabalho está em estabelecer de forma clara a ligação entre ajuste local de superfícies quadráticas e estimação diferencial em malhas.

Em continuidade, Cazals e Pouget [25] formulam a estimação de quantidades diferenciais como um problema geral de ajuste de *jatos osculadores* — isto é, expansões polinomiais locais truncadas que preservam a informação diferencial da superfície até uma dada ordem. Os autores mostram que um *jato* de ordem n contém toda a informação diferencial até essa ordem e que, em particular, as quantidades de curvatura podem ser recuperadas a partir do *jato* de segunda ordem. Na forma de Monge, em que o eixo normal e as direções principais estão alinhados com o sistema local, os coeficientes quadráticos da expansão determinam diretamente as curvaturas principais. Essa formulação fornece um embasamento teórico mais robusto para métodos de ajuste polinomial local, além de explicitar questões numéricas relevantes, como o condicionamento do sistema de ajuste.

Nessa mesma linha, Razdan e Bae [13] propõem um método baseado no ajuste de superfícies de Bézier biquadráticas para estimar curvaturas em malhas triangulares. Os autores argumentam que essa representação oferece maior flexibilidade geométrica do que os ajustes quadráticos explícitos e permite incorporar equações de suavização, o que pode ser vantajoso em dados ruidosos ou triangulações irregulares. Assim, o trabalho evidencia que, mesmo dentro da classe dos métodos de ajuste local, a escolha da representação da superfície aproximante influencia diretamente a estabilidade e a qualidade da estimativa.

Já Petitjean [26] apresenta uma revisão abrangente sobre métodos de recuperação de quádricas em malhas triangulares, situando a estimação local de curvatura em um contexto mais amplo de recuperação geométrica, segmentação e engenharia reversa. O autor destaca o papel central das superfícies quadráticas — como esferas, cilindros, cones, elipsoides e paraboloides — como modelos geométricos simples e recorrentes em problemas de reconstrução e descrição de forma. Para a presente revisão, esse trabalho é importante porque mostra que o ajuste local de superfícies não constitui apenas uma estratégia numérica para cálculo de curvatura, mas integra uma tradição mais ampla de modelagem geométrica baseada em aproximações analíticas locais.

Uma formulação particularmente relevante para o presente trabalho é a das superfícies implícitas. Goldman [27] reúne expressões fechadas para curvatura média e curvatura Gaussiana de superfícies definidas implicitamente por $F(x, y, z) = 0$, mostrando como essas grandezas podem ser obtidas diretamente a partir do gradiente e da Hessiana da função implícita. O autor também explicita a relação entre formulações paramétricas e implícitas, o que torna o trabalho conceitualmente importante para a interpretação de métodos que recorrem a representações locais contínuas da superfície. Para esta revisão, sua principal contribuição está em fornecer uma base analítica clara para o cálculo de quantidades diferenciais em superfícies representadas de forma implícita.

Como exemplo de abordagem intermediária entre ajuste local e métodos discretos, Mao et al. [14] propõem um método de estimação de curvatura baseado em triângulos definidos por vértices e normais nodais, no qual cada face triangular plana é interpolada por um *patch* triangular cúbico de Bézier. A curvatura é inicialmente computada por triângulo e, em seguida, agregada nos vértices por média ponderada das faces adjacentes. Essa formulação é relevante para esta revisão porque mostra que o uso explícito de normais nodais pode enriquecer a reconstrução local da geometria sem recorrer diretamente a um ajuste local em mínimos quadrados, situando-se em uma faixa intermediária entre discretização direta e reconstrução local parametrizada.

Para este TCC, o ponto central dessa literatura é que métodos baseados em ajuste local tendem a capturar de maneira mais direta a geometria suave subjacente da superfície, mas sua precisão e estabilidade dependem criticamente da representação adotada para a superfície aproximante, do condicionamento numérico do problema e da extensão da vizinhança utilizada. Essa observação oferece sustentação teórica direta para a análise do AQFC com diferentes anéis de vizinhança.

2.3.3 Comparação entre Métodos, Robustez e Sensibilidade à Vizinhança

A literatura comparativa mostra que não existe um estimador de curvatura universalmente superior em todos os cenários. Gatzke e Grimm [15, 16] realizam comparações sistemáticas entre diferentes métodos e mostram que o desempenho

depende de fatores como a geometria da superfície, a resolução da malha, a regularidade da triangulação, a valência local (número de arestas incidentes a um vértice) e o ruído presente nos dados. Em particular, os autores observam que métodos discretos são computacionalmente atraentes, mas sua sensibilidade a ruído, valência e irregularidade da malha limita sua confiabilidade em triangulações menos regulares. Em contrapartida, métodos baseados em ajuste local com vizinhanças mais amplas tendem a apresentar melhor desempenho global, embora com maior custo computacional e sem ganho de precisão garantido para vizinhanças excessivamente grandes.

Essa mesma linha de raciocínio aparece em Rusinkiewicz [23] e é aprofundada por Kalogerakis et al. [28], que discutem explicitamente a robustez da estimativa frente a ruído, amostragem não uniforme e valores discrepantes. Os autores mostram que operadores clássicos de 1-anel funcionam adequadamente em malhas bem condicionadas, mas deixam de ser robustos quando a tesselação é irregular ou ruidosa. Também argumentam que operadores com regiões de suporte maiores tendem a ser mais estáveis, porém podem propagar extremos de curvatura além de fronteiras de feição e introduzir suavização excessiva. Como resposta a esse problema, propõem uma formulação robusta baseada em reponderação iterativa e adaptação automática da vizinhança efetiva.

Estudos comparativos diretamente alinhados ao desenho experimental deste trabalho podem ser encontrados em Surazhsky et al. [29] e Magid et al. [30]. Surazhsky et al. comparam cinco algoritmos de estimativa de curvatura e suas modificações em malhas triangulares obtidas como tesselações de superfícies NURBS sintéticas (Non-Uniform Rational B-Splines – superfícies paramétricas geradas artificialmente para teste, com geometria de referência conhecida), confrontando os resultados com os valores analíticos das superfícies de origem. Os autores mostram que o desempenho depende da grandeza geométrica considerada: o ajuste paraboloidal apresentou os melhores resultados para curvatura média, enquanto o esquema baseado em Gauss–Bonnet, no qual a curvatura Gaussiana é estimada a partir do defeito angular na vizinhança do vértice, se destacou na estimativa da curvatura Gaussiana. Já Magid et al. ampliam essa análise para incluir também dados provenientes de varredura, confirmando que diferentes algoritmos respondem de maneira distinta conforme a

superfície, a discretização e o tipo de dado disponível. Além disso, mostram que, em superfícies sintéticas, tanto o método de ajuste parabólico quanto o esquema de Gauss–Bonnet apresentam comportamento convergente com o refinamento da malha. Em conjunto, esses estudos reforçam a ideia de que não há um método universalmente superior e sustentam diretamente a proposta deste trabalho de comparar métodos distintos em diferentes geometrias e níveis de refinamento.

Em uma perspectiva mais recente, Kronenberger, Wirjadi e Hagen [31] apresentam uma comparação empírica entre estimadores de curvatura aplicados a imagens volumétricas e malhas triangulares, avaliando qualidade e custo computacional em geometrias com representação analítica conhecida. Os autores concluem que, em escala comparável de discretização, os métodos baseados em malha tendem a permitir estimativas mais precisas, embora o desempenho em tempo dependa da densidade superficial da geometria considerada. Para o presente trabalho, esse resultado é relevante por reforçar a competitividade das abordagens baseadas em malha quando se dispõe de superfícies trianguladas de boa qualidade geométrica.

No caso dos operadores do tipo Laplaciano, Sorkine [32] destaca que sua importância no processamento geométrico não decorre apenas do cálculo de curvatura, mas do fato de codificarem informação diferencial local de forma linear e eficiente, sustentando tarefas como edição de malhas, suavização, compressão e interpolação. Ao mesmo tempo, a autora distingue formulações puramente topológicas de discretizações geométricas, como as ponderações cotangentes, mostrando que diferentes definições do operador Laplaciano incorporam diferentes informações sobre a malha. Essa observação é particularmente relevante aqui, pois reforça que a qualidade de um método baseado em Laplace–Beltrami depende não apenas da teoria contínua subjacente, mas da construção discreta específica empregada.

Finalmente, Váša et al. [33] mostram, em um experimento amplo com diferentes tipos de malhas e estimadores, que nenhuma abordagem fornece resultados confiáveis em todas as situações. A partir dessa constatação, os autores investigam estatísticas de malha que permitam antecipar qual estimador tende a apresentar melhor desempenho em determinado conjunto de dados e propõem inclusive um *meta-estimador* baseado nessa ideia, isto é, um procedimento que seleciona automaticamente, com base em propriedades da malha, qual estimador de curvatura tende

a ser mais adequado para cada caso. Essa contribuição é particularmente importante porque reforça a interpretação de que a escolha do estimador não pode ser dissociada das propriedades da malha sobre a qual ele opera.

Em síntese, a literatura comparativa converge para quatro dimensões principais de avaliação: acurácia em relação à solução analítica, robustez a ruído, sensibilidade à irregularidade da malha e custo computacional. Para o presente trabalho, essa síntese é especialmente importante porque mostra que a comparação entre AQFC e Laplace–Beltrami não deve ser interpretada apenas como confronto entre dois valores de curvatura média, mas como comparação entre duas estratégias distintas de discretização da geometria diferencial, ambas sensíveis, em graus diferentes, à qualidade da malha e à escala local de informação.

2.3.4 Qualidade de Malha e Influência da Discretização

A literatura sobre qualidade de malha fornece um complemento essencial à discussão anterior, pois mostra que o desempenho de métodos numéricos não depende apenas da formulação matemática empregada, mas também da qualidade geométrica da discretização sobre a qual essa formulação opera. Knupp [34] organiza métricas de qualidade em um arcabouço algébrico baseado na matriz Jacobiana do elemento e em matrizes relacionadas, mostrando como essas métricas podem ser construídas, classificadas e analisadas de forma unificada. Um aspecto importante dessa formulação é o uso de um elemento de referência ideal, em relação ao qual a qualidade do elemento real pode ser avaliada por meio de propriedades geométricas codificadas na Jacobiana.

Em uma visão mais ampla, Sorgente et al. [35] mostram que a noção de qualidade de malha depende inevitavelmente do contexto de aplicação e da finalidade da discretização. Os autores apresentam uma revisão abrangente de indicadores geométricos, algébricos e orientados ao desempenho numérico, distinguindo ainda medidas locais e globais de qualidade. Também destacam que esses indicadores são usados não apenas para avaliar malhas existentes, mas também para orientar processos de geração, refinamento e otimização de malhas. Para o presente trabalho, essa perspectiva é importante porque reforça que tamanho de elemento e qualidade geométrica são aspectos relacionados, mas não equivalentes.

No caso específico de malhas triangulares, Farin [36] discute medidas de forma associadas à proximidade de um triângulo em relação à configuração equilátera, incluindo ângulo máximo, ângulo mínimo, razão entre circunrádio e inrádio e medidas de alongamento. Já Shewchuk [37] mostra que algoritmos de refinamento de Delaunay permitem controlar simultaneamente limites angulares, comprimentos de aresta, cardinalidade da malha e gradação entre elementos pequenos e grandes, sendo por isso particularmente relevantes em aplicações de interpolação, elementos finitos e volumes finitos. Em conjunto, essas contribuições reforçam que refinamento e qualidade de malha não são sinônimos: uma malha mais refinada pode continuar geometricamente desfavorável se a forma dos elementos permanecer inadequada.

Para a estimativa de curvatura, essa conclusão tem implicação direta. Em discretizações baseadas no operador Laplace–Beltrami, pesos geométricos e áreas associadas aos vértices tornam a estimativa sensível à forma dos triângulos e à regularidade da triangulação. Já em métodos de ajuste quadrático local, a distribuição espacial dos pontos da vizinhança influencia tanto o condicionamento numérico do problema quanto a fidelidade da superfície reconstruída. Assim, a análise dos erros observados neste trabalho deve considerar que parte deles pode decorrer não apenas da formulação do estimador, mas também da maneira como a malha representa a superfície contínua subjacente.

Em síntese, a literatura mostra que a avaliação de métodos de estimativa de curvatura em malhas triangulares deve considerar conjuntamente pelo menos quatro fatores: a formulação do método, a escala da vizinhança, a qualidade geométrica da malha e o refinamento da discretização. Essa conclusão fornece base teórica direta para a proposta deste trabalho, que compara um método baseado em operador discreto e um método baseado em ajuste quadrático local em diferentes superfícies analíticas, níveis de refinamento e extensões de vizinhança.

Capítulo 3

Metodologia

3.1 Caracterização da Pesquisa

Este trabalho caracteriza-se como uma pesquisa aplicada, de abordagem quantitativa e natureza computacional. O estudo consiste na implementação e avaliação comparativa de métodos numéricos para estimativa da curvatura média em malhas triangulares, sendo eles: o ajuste quadrático algébrico (*Algebraic Quadric Fitting Curvature* – AQFC) e a discretização FEM do operador Laplace–Beltrami (LB).

As geometrias analíticas consideradas neste estudo tiveram suas malhas triangulares geradas com o software livre *Gmsh* [38]. A partir dessas malhas, foi utilizado o método AQFC, proposto por Makovník e Chalmovianský [9], para estimar a curvatura média em todos os vértices da malha triangular. O método AQFC foi integralmente implementado em Python por meio de código desenvolvido pelo próprio autor deste trabalho.

Além da formulação original, foi investigada a influência da ampliação da vizinhança utilizada no ajuste quadrático local, considerando vizinhanças de um, dois e três anéis de vértices adjacentes, com o objetivo de analisar o impacto dessa estratégia na precisão das estimativas de curvatura.

Para fins de comparação, foi utilizada a estimativa de curvatura baseada na discretização do operador Laplace–Beltrami, conforme a formulação apresentada por Santos [4].

As implementações foram realizadas utilizando bibliotecas científicas da linguagem Python, como NumPy e SciPy. Os resultados foram processados e exportados

no formato VTK (*Visualization Toolkit*), permitindo a visualização dos campos de curvatura e das métricas de erro no software ParaView.

A validação dos métodos foi conduzida por meio da comparação entre as estimativas numéricas e os valores analíticos de referência das geometrias com curvatura conhecida (esfera, elipsoide, toro e cilindro). Os erros foram quantificados a partir da diferença entre a curvatura estimada e a curvatura analítica, considerando diferentes níveis de refinamento de malha.

Adicionalmente, métricas de qualidade dos elementos triangulares foram utilizadas para investigar a influência da distorção geométrica no erro numérico e subsidiar uma proposta preliminar de abordagem híbrida entre os métodos avaliados.

3.2 Configuração experimental

Os experimentos foram realizados em quatro geometrias: uma **esfera** ($r = 0.5$), um **elipsoide** ($a = 10.0$, $b = c = 4.0$), um **toro** ($R = 0.5$, $r = 0.4$) e um **cilindro** ($r = 0.5$ e $L = 2.0$), como pode ser observado na figura 3.1.

Para cada geometria, através do software Gmsh, criaram-se cinco malhas com diferentes graus de refinamento, a partir da variação do parâmetro h que é o comprimento médio das arestas do elemento triangular. Vale ressaltar também que para a geometria do elipsoide, outras malhas foram geradas utilizando um novo código para criação da geometria no Gmsh, já que o primeiro apresentou uma distorção significativa em um certo ponto da malha. O novo método corrigiu a distorção e melhorou a regularidade dos elementos. Essa reconstrução foi denominada "**elipsoide 2**" para diferenciação. A malha original foi mantida nos testes por apresentar resultados comparativos relevantes com a nova versão, sendo chamada de "**elipsoide 1**".

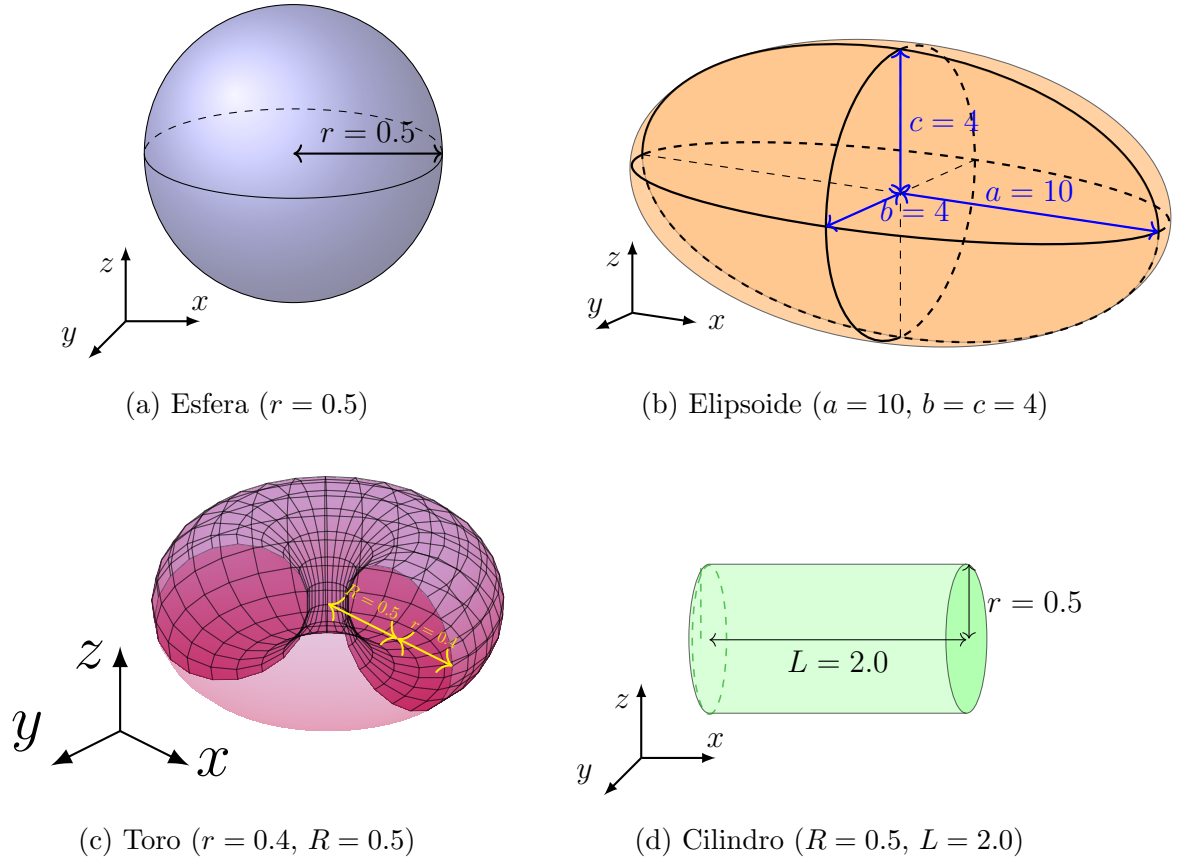


Figura 3.1: Geometrias Analisadas

Na figura 3.2 abaixo, segue uma demonstração comparativa de malhas geradas com cada método. Onde é possível observar que no **elipsoide 2** há uma divisão da malha em setores, o que facilita para a uniformização de seus elementos, e que algumas pequenas distorções concentram-se nas junções desses setores.

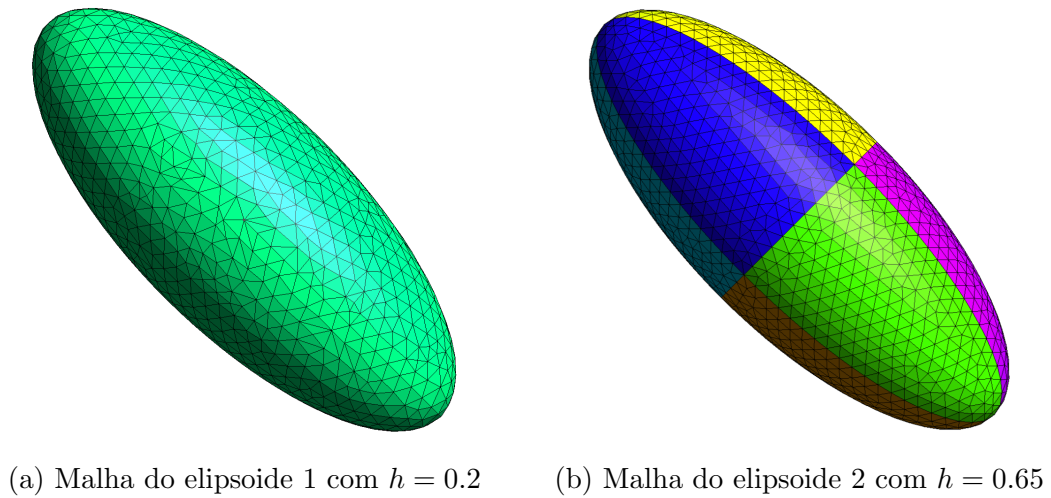


Figura 3.2: Comparação entre dois métodos de criação para as malhas do elipsoide

3.3 Métodos Utilizados

Nesta seção, são discutidas as bases teóricas dos métodos utilizados para a estimativa de curvatura em superfícies. São apresentados, em detalhes, o método de Ajuste Quadrático Local de Curvatura (AQFC) e o método baseado no operador Laplace–Beltrami (LB), destacando suas formulações e princípios fundamentais.

3.3.1 Ajuste Quadrático Local de Curvatura (AQFC)

O método *Algebraic Quadric Fitting Curvature* (AQFC), proposto por Makovník e Chalmovianský [9], insere-se na classe de métodos que estimam a curvatura a partir do ajuste local de uma superfície contínua à geometria discreta da malha.

Neste trabalho, adota-se a denominação *Ajuste Quadrático Local de Curvatura* (AQFC) como tradução livre do termo original, mantendo-se, contudo, a mesma formulação proposta pelos autores.

Mais especificamente, o método ajusta localmente uma superfície quádrlica implícita a uma vizinhança do vértice de interesse, utilizando os vértices e suas normais associadas. Os coeficientes da quádrlica são determinados por meio da minimização simultânea das distâncias algébricas ao quadrado e dos desvios entre as normais de entrada e o gradiente da superfície ajustada. A curvatura é então estimada como a curvatura da superfície ajustada, avaliada na projeção ortogonal do vértice sobre a quádrlica. A seguir, descrevem-se as etapas do procedimento.

1. Seleção da Vizinhança e Dados de Entrada:

Seja (V, n) o vértice-normal processado da malha, onde V representa a posição e n a normal associada. Para esse vértice, define-se uma vizinhança N composta por um certo número de vértices adjacentes, cada um acompanhado de sua respectiva normal (V_i, n_i) para $i = 1, \dots, m$.

2. Ajuste Quadrático:

O método consiste em ajustar localmente uma superfície quadrática que aproxima a geometria da malha na vizinhança N .

(a) Definição da Superfície Implícita:

A superfície implícita é definida como o conjunto de nível nulo de uma função quadrática $q : \mathbb{R}^3 \rightarrow \mathbb{R}$, dada por:

$$q(x, y, z) = a_{11}x^2 + a_{22}y^2 + a_{33}z^2 + a_{12}xy + a_{13}xz + a_{23}yz + a_{14}x + a_{24}y + a_{34}z + a_{44} \quad (3.1)$$

onde $a_{11}, \dots, a_{44}$ são coeficientes a serem determinados. Assim, a superfície implícita $Z(q)$ é

$$\mathcal{Z}(q) = \{(x, y, z)^\top \in \mathbb{R}^3 \mid q(x, y, z) = 0\} \quad (3.2)$$

(b) Formulação do Problema de Mínimos Quadrados:

Para ajustar a superfície quadrática à vizinhança $N = \{(V_i, n_i)\}_{i=1}^m$, define-se a função objetivo que combina:

- Os termos correspondentes às distâncias algébricas ao quadrado ponderadas $q(V_i)^2$ entre os vértices V_i e a superfície $Z(q)$;
- Os desvios entre os gradientes da função quadrática $\nabla q(V_i)$ e as normais de entrada n_i .

Essa função é dada por:

$$f(a_{11}, \dots, a_{44}) = \sum_{i=1}^m (\omega_{V_i} q(V_i)^2 + \omega_{n_i} \|n_i - \nabla q(V_i)\|^2) \quad (3.3)$$

onde os pesos ω_{V_i} e ω_{n_i} são escolhidos de forma a dar maior importância aos pontos próximos a V e à conformidade das normais. No artigo, os autores adotam os seguintes pesos:

$$\omega_{V_i} = \frac{1}{e^{\|V - V_i\|^4}} \quad (3.4a)$$

$$\omega_{n_i} = \frac{10^{-4}}{e^{\left\| \frac{n}{\|n\|} - \frac{n_i}{\|n_i\|} \right\|^2}} \quad (3.4b)$$

3. Resolução do Sistema Linear:

Como a função f é quadrática em relação aos coeficientes, seu mínimo global ocorre onde todas as derivadas parciais se anulam. Assumindo que a Hessiana é positiva, determinam-se os coeficientes resolvendo o sistema

$$\frac{\partial f}{\partial a_{ij}}(a_{11}, \dots, a_{44}) = 0, \quad \text{para } 1 \leq i \leq j \leq 4 \quad (3.5)$$

A resolução deste sistema linear fornece os valores de $a_{11}, \dots, a_{44}$ que melhor ajustam a superfície local à vizinhança N .

4. Superfície Implícita:

Após a determinação dos coeficientes a_{ij} , a função quadrática q define implicitamente uma superfície $Z(q)$, que representa a aproximação local da malha na vizinhança considerada.

5. Cálculo da Curvatura:

Uma vez que os coeficientes a_{ij} são determinados para cada ponto V da malha, a curvatura média $\bar{H}$ pode ser estimada pela seguinte equação:

$$\bar{H} = \frac{s(n, V)^\top \mathbf{H}_q(p(V)) s(n, V) - \|s(n, V)\|^2 \text{tr}(\mathbf{H}_q(p(V)))}{2\|s(n, V)\|^3}. \quad (3.6)$$

onde $\mathbf{H}_q$ denota a matriz Hessiana da função quadrática q . Para a função definida em (3.1), essa matriz é constante e dada por

$$\mathbf{H}_q = \begin{bmatrix} \frac{\partial^2 q}{\partial x^2} & \frac{\partial^2 q}{\partial x \partial y} & \frac{\partial^2 q}{\partial x \partial z} \\ \frac{\partial^2 q}{\partial y \partial x} & \frac{\partial^2 q}{\partial y^2} & \frac{\partial^2 q}{\partial y \partial z} \\ \frac{\partial^2 q}{\partial z \partial x} & \frac{\partial^2 q}{\partial z \partial y} & \frac{\partial^2 q}{\partial z^2} \end{bmatrix} = \begin{bmatrix} 2a_{11} & a_{12} & a_{13} \\ a_{12} & 2a_{22} & a_{23} \\ a_{13} & a_{23} & 2a_{33} \end{bmatrix} \quad (3.7)$$

e $s(\mathbf{n}, V)$ corresponde ao gradiente de q em $p(V)$, com comprimento corrigido pela norma de $\mathbf{n}$, dado por:

$$s(\mathbf{n}, V) = \|\mathbf{n}\| \frac{\nabla q(p(V))}{\|\nabla q(p(V))\|}. \quad (3.8)$$

e $p(V)$ é o ponto da superfície implícita $Z(q)$ que mais se aproxima do ponto original V , que pode ser expresso por:

$$p(V) = \arg \min_{W \in Z(q)} \|V - W\|. \quad (3.9)$$

A curvatura não é avaliada diretamente no vértice V , mas em sua projeção ortogonal $p(V)$ sobre a quádrlica ajustada. Essa escolha é adotada no método original porque $p(V)$ pertence à superfície implícita $Z(q)$, de modo que o gradiente $\nabla q(p(V))$ não sofre a distorção associada ao deslocamento entre o vértice da malha e a superfície ajustada. Como as expressões da curvatura média e da curvatura Gaussiana envolvem potências da norma do gradiente no denominador, erros nessa quantidade podem ser amplificados quando a avaliação é feita fora da superfície ajustada.

3.3.2 Laplace–Beltrami (LB)

Como método de comparação, adota-se a discretização do operador Laplace–Beltrami por meio da formulação em elementos finitos (*FEM*), conforme apresentada por Santos [4].

Inicialmente, consideremos uma superfície suave e fechada $\Gamma \subset \mathbb{R}^3$, parametrizada pelo vetor posição $\mathbf{x}$.

Definimos o espaço:

$$\mathbb{V}_\Gamma = H^1(\Gamma)^m = \left\{ \mathbf{v} = (v_1, \dots, v_m) : v_i \in H^1(\Gamma), \forall i = 1, \dots, m \right\} \quad (3.10)$$

onde $H^1(\Gamma)$ é o espaço de Sobolev definido sobre a superfície, dado por:

$$H^1(\Gamma) = \left\{ v \in L^2(\Gamma) : \nabla_s v \in L^2(\Gamma)^3 \right\} \quad (3.11)$$

sendo $\nabla_s v$ o gradiente tangencial à superfície Γ .

O espaço $L^2(\Gamma)$ é definido como:

$$L^2(\Gamma) = \left\{ v : \Gamma \rightarrow \mathbb{R}, \int_\Gamma v^2 d\Gamma < \infty \right\} \quad (3.12)$$

um espaço de dimensão infinita, caracterizado pela integral de Lebesgue sobre Γ . Note que $\mathbb{V}_\Gamma = H^1(\Gamma)^m$ é o produto cartesiano de m espaços $H^1(\Gamma)$. Como a superfície Γ é fechada, tem-se $\partial\Gamma = \emptyset$, não havendo necessidade de impor condições de contorno.

No contexto do operador Laplace–Beltrami, toma-se $m = 3$, de modo que as funções vetoriais $\mathbf{x}$ (vetor posição) e as funções teste $\mathbf{w}$ pertencem a $\mathbb{V}_\Gamma = H^1(\Gamma)^3$. Por simplicidade, denotamos $V = H^1(\Gamma)^3$.

Sobre essa superfície, o operador Laplace–Beltrami é utilizado para relacionar a geometria da malha ao vetor curvatura:

$$\mathbf{C}(\mathbf{x}) = \kappa \mathbf{n} = \nabla_s^2 \mathbf{x} = \nabla_s \cdot (\nabla_s \mathbf{x}), \quad (3.13)$$

em que $\mathbf{x}$ é o vetor posição dos pontos da superfície, ∇_s denota o operador gradiente sobre superfícies, $\mathbf{n}$ é o vetor normal unitário e κ representa a curvatura escalar associada ao vetor curvatura.

Para obter uma formulação adequada à discretização numérica, aplica-se a forma fraca do operador, obtida por integração por partes na superfície, o que reduz a ordem diferencial. Considerando uma função teste vetorial $\mathbf{w} \in V$ e integrando sobre Γ , tem-se:

$$-\int_{\Gamma} \nabla_s^2 \mathbf{x} \cdot \mathbf{w} \, d\Gamma = \int_{\Gamma} \nabla_s \mathbf{x} : \nabla_s \mathbf{w} \, d\Gamma - \int_{\partial\Gamma} (\nabla_s \mathbf{x} \cdot \mathbf{w}) \cdot \mathbf{n} \, d(\partial\Gamma). \quad (3.14)$$

Como as superfícies consideradas são fechadas ($\partial\Gamma = \emptyset$), o termo de contorno é nulo. Assim, a formulação variacional consiste em encontrar $\mathbf{x} \in V$ tal que, para todo $\mathbf{w} \in V$:

$$\int_{\Gamma} \nabla_s^2 \mathbf{x} \cdot \mathbf{w} \, d\Gamma = - \int_{\Gamma} \nabla_s \mathbf{x} : \nabla_s \mathbf{w} \, d\Gamma. \quad (3.15)$$

A discretização é realizada em cada elemento triangular da malha interfacial. Para isso, introduz-se inicialmente um sistema de coordenadas locais bidimensional no plano do triângulo. No método de Galerkin, tanto as coordenadas do elemento quanto as funções teste são representadas pelas mesmas funções de forma lineares. Para um elemento genérico, escrevem-se as aproximações:

$$x = \sum_{i=1}^3 x_i^e N_i, \quad y = \sum_{i=1}^3 y_i^e N_i, \quad (3.16)$$

onde N_i são as funções de forma lineares do triângulo e (x_i^e, y_i^e) são as coordenadas nodais do elemento no sistema local bidimensional. De forma análoga, as funções teste são representadas pelas mesmas funções de forma, conforme a formulação de Galerkin adotada:

$$w_x = \sum_{j=1}^3 N_j, \quad w_y = \sum_{j=1}^3 N_j. \quad (3.17)$$

Substituindo essas aproximações na forma fraca, o termo envolvendo os gradientes superficiais passa a ser escrito em termos dos produtos das derivadas das funções

de forma nas duas direções locais do elemento. Dessa forma, obtém-se uma expressão envolvendo integrais do tipo:

$$\int_{\Gamma} \nabla_s \mathbf{x} : \nabla_s \mathbf{w} d\Gamma = \int_{\Gamma} \left(x_i \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + x_i \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} + y_i \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + y_i \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right) d\Gamma, \quad (3.18)$$

evidenciando que a discretização depende diretamente das derivadas das funções de forma nas coordenadas locais do triângulo. Para simplificar a notação, definem-se os termos elementares:

$$c_{\xi\xi} = \int_{\Gamma} \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} d\Gamma, \quad c_{\eta\eta} = \int_{\Gamma} \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} d\Gamma, \quad (3.19)$$

cujas soma define o operador Laplace–Beltrami discreto no elemento:

$$C = c_{\xi\xi} + c_{\eta\eta}. \quad (3.20)$$

Os termos $c_{\xi\xi}$ e $c_{\eta\eta}$ representam, portanto, as matrizes que compõem o operador Laplaciano do elemento triangular linear. Embora a notação utilize os índices ξ e η para identificar as duas direções tangenciais locais do elemento, as derivadas são escritas em relação a x e y , que representam as coordenadas locais bidimensionais no plano do triângulo.

Entretanto, esse operador C é inicialmente definido em um sistema de coordenadas locais e, por isso, não pode ser aplicado diretamente à superfície inserida no espaço tridimensional. Torna-se necessária uma transformação entre o sistema bidimensional local do elemento e as coordenadas cartesianas globais da malha. Para um triângulo genérico com vértices i , j e k , define-se inicialmente a direção tangente unitária $\boldsymbol{\xi}$:

$$\boldsymbol{\xi} = \frac{\mathbf{x}_j - \mathbf{x}_i}{\|\mathbf{x}_j - \mathbf{x}_i\|}, \quad (3.21)$$

ao longo de uma das arestas do triângulo. A segunda direção tangente unitária $\boldsymbol{\eta}$, ortogonal a $\boldsymbol{\xi}$ e contida no plano do triângulo, é:

$$\boldsymbol{\eta} = \frac{[(\mathbf{x}_j - \mathbf{x}_i) \times (\mathbf{x}_k - \mathbf{x}_i)] \times (\mathbf{x}_j - \mathbf{x}_i)}{\|[(\mathbf{x}_j - \mathbf{x}_i) \times (\mathbf{x}_k - \mathbf{x}_i)] \times (\mathbf{x}_j - \mathbf{x}_i)\|}. \quad (3.22)$$

Nessa construção, o primeiro produto vetorial gera um vetor normal ao plano do triângulo e o segundo gera um vetor perpendicular a essa normal e à direção $\boldsymbol{\xi}$, permanecendo, portanto, no plano do elemento.

Uma vez definida essa transformação geométrica, o operador discreto pode ser aplicado separadamente às coordenadas cartesianas nodais da malha, produzindo os componentes do vetor curvatura:

$$\mathbf{C}(\mathbf{x}) = (C_x, C_y, C_z), \quad C_x = C x, \quad C_y = C y, \quad C_z = C z, \quad (3.23)$$

em que x , y e z representam os vetores das coordenadas cartesianas nodais.

A magnitude do vetor curvatura discreto é então:

$$\mathbf{C}(\mathbf{x})_t = \sqrt{C_x^2 + C_y^2 + C_z^2}. \quad (3.24)$$

Para obter valores nodais consistentes, a magnitude do vetor curvatura deve ser dividida pela área associada a cada nó. Assim, a curvatura escalar nodal é obtida por:

$$\kappa = \frac{\mathbf{C}(\mathbf{x})_t}{A_i}, \quad (3.25)$$

em que A_i é a área nodal associada ao nó i .

Como neste trabalho a comparação entre métodos é realizada em termos de curvatura média, usa-se a relação $\kappa = 2H$. Desse modo, a curvatura média nodal pode ser escrita como:

$$H = \frac{\mathbf{C}(\mathbf{x})_t}{2A_i}. \quad (3.26)$$

O cálculo da área nodal é tratado separadamente por Santos [4], sendo baseado na área de Voronoi associada ao vértice e incluindo uma correção específica para triângulos obtusos, de forma a evitar que a área atribuída ao nó ultrapasse a área do elemento quando o circuncentro se localiza fora do triângulo. Esse procedimento foi reproduzido na Seção 2.2.3.

Dessa forma, o método de Laplace–Beltrami empregado neste trabalho estima a curvatura média a partir da aplicação do operador discreto à geometria nodal da malha superficial, seguida da normalização pela área nodal correspondente.

3.4 Procedimentos Metodológicos

Nesta seção descreve-se como foram implementados computacionalmente os métodos AQFC 3.3.1, LB 3.3.2, assim como outras funções necessárias para os cálculos da curvatura média em cada vértice da malha.

3.4.1 Pré-processamento: Cálculo de Vizinhanças e Normais

Antes de calcular a curvatura, o algoritmo realiza etapas de pré-processamento para determinar as vizinhanças dos nós e suas normais. A função `Build_Neighborhoods` identifica os nós vizinhos de primeira, segunda e terceira ordem, bem como os elementos (triângulos) adjacentes a cada nó. A função `Calculate Normals` calcula as normais para cada vértice a partir de contribuições normalizadas das suas respectivas faces pertencentes.

Cálculo das Normais

1. Para cada face triangular da malha, a normal em cada ponto da face é calculada usando o produto vetorial entre suas arestas adjacentes.
2. A normais calculadas são unitarizadas e então somada às normais de cada um dos seus respectivos vértices que compõem a face.
3. Após iterar por todas as faces, cada vértice terá acumulado a soma das normais de suas faces adjacentes.
4. Finalmente, o vetor normal de cada vértice é normalizado (dividido por seu módulo) para se tornar um vetor unitário.

Algoritmo 1 Cálculo dos Vetores Normais

- 1: **procedure** `CALCULATE.NORMALS`(V , IEN)
- 2: $\triangleright V$ é uma lista com as coordenadas (x, y, z) de cada vértice.
- 3: $\triangleright IEN$ é a matriz de conectividade, uma lista de triângulos, onde cada face é uma lista de 3 índices de vértices.

```

4:    $npoints \leftarrow$  número de vértices
5:    $Normals \leftarrow$  inicializar vetor de  $npoints$  vetores (0,0,0)
6:   for cada elemento  $e$  em  $IEN$  com vértices  $v_1, v_2, v_3$  do
7:      $aresta_{12} \leftarrow V[v_2] - V[v_1]$ 
8:      $aresta_{13} \leftarrow V[v_3] - V[v_1]$ 
9:      $normal\_face \leftarrow \text{ProdutoVetorial}(aresta_{12}, aresta_{13})$ 
10:     $normal\_face \leftarrow \frac{normal\_face}{||normal\_face||}$ 
11:     $Normals[v_1] \leftarrow Normals[v_1] + normal\_face$ 
12:     $Normals[v_2] \leftarrow Normals[v_2] + normal\_face$ 
13:     $Normals[v_3] \leftarrow Normals[v_3] + normal\_face$ 
14:  end for
15:  for  $i$  de 0 até  $(npoints - 1)$  do
16:     $Normals[i] \leftarrow \frac{Normals[i]}{||Normals[i]||}$ 
17:  end for
18:  return  $Normals$ 
19: end procedure

```

Obtenção das Vizinhanças

1. Percorrer todas as faces (elementos) da malha para construir:
 - a lista de elementos incidentes a cada vértice (faces que contém o vértice);
 - a vizinhança de 1^a ordem: o conjunto de vértices conectados diretamente por aresta (vizinhos imediatos).
2. A partir da vizinhança de 1^a ordem (conjuntos), construir a vizinhança de 2^a ordem como a união das vizinhanças de 1^a ordem dos nós da 1^a ordem, removendo os nós que já pertencem à 1^a ordem e o próprio nó.
3. Da mesma forma, construir a vizinhança de 3^a ordem como a união das vizinhanças de 1^a ordem dos nós da 2^a ordem, removendo os nós das ordens inferiores e o próprio nó.
4. Recomenda-se usar **conjuntos** enquanto constrói vizinhanças (evita duplicatas e torna a lógica de união/interseção direta). Ao final, converte-se cada conjunto em lista.

Algoritmo 2 Cálculo de Nós e Elementos Vizinhos

```
1: procedure BUILD_NEIGHBORHOODS( $IEN$ )
2:                                      $\triangleright IEN$  : Matriz de Conectividade
3:    $num\_vertices \leftarrow \max(IEN) + 1$                                       $\triangleright$  Número total de vértices
4:   Inicializar  $nos\_vizinhos[i] \leftarrow []$  para  $i = 0, \dots, num\_vertices - 1$ 
5:   Inicializar  $elementos\_vizinhos[i] \leftarrow []$  para  $i = 0, \dots, num\_vertices - 1$ 
6:   for cada elemento  $elemento$  com índice  $elem\_idx$  em  $IEN$  do
7:      $elem\_nodes \leftarrow$  lista dos nós de  $elemento$ 
8:     for cada nó  $no$  em  $elem\_nodes$  do
9:       Adicionar  $elem\_idx$  em  $elementos\_vizinhos[no]$ 
10:      for cada  $outro\_no$  em  $elem\_nodes$  do
11:        if  $outro\_no \neq no$  e  $outro\_no$  não está em  $nos\_vizinhos[no]$  then
12:          adicionar  $outro\_no$  a  $nos\_vizinhos[no]$ 
13:        end if
14:      end for
15:    end for
16:  end for
17:   $vizinhos\_geral \leftarrow$  cópia de  $nos\_vizinhos$                                       $\triangleright 1^a$  ordem pronta

18:  — Construção da 2a vizinhança —
19:  Inicializar lista  $vizinhos\_segundo\_geral$ 
20:  for  $i = 0$  até  $(num\_vertices - 1)$  do
21:     $primeiro \leftarrow$  conjunto( $vizinhos\_geral[i]$ )
22:     $segundo \leftarrow$  conjunto vazio
23:    for cada  $j$  em  $primeiro$  do
24:       $segundo \leftarrow segundo \cup vizinhos\_geral[j]$ 
25:    end for
26:     $segundo \leftarrow segundo - primeiro$ 
27:    Remover  $i$  de  $segundo$ 
28:    Adicionar lista( $segundo$ ) em  $vizinhos\_segundo\_geral$ 
29:  end for
```

```

30:  — Construção da 3ª vizinhança —
31:  Inicializar lista vizinhos_terceiro_geral
32:  for  $i = 0$  até  $(num\_vertices - 1)$  do
33:       $primeiro \leftarrow conjunto(vizinhos\_geral[i])$ 
34:       $segundo \leftarrow conjunto(vizinhos\_segundo\_geral[i])$ 
35:       $terceiro\_ordem \leftarrow$  conjunto vazio
36:      for cada  $v2$  em segundo do
37:           $terceiro \leftarrow terceiro \cup vizinhos\_geral[v2]$ 
38:      end for
39:       $terceiro \leftarrow terceiro - primeiro$ 
40:       $terceiro \leftarrow terceiro - segundo$ 
41:      Remover  $i$  de terceiro
42:      Adicionar lista(terceiro) em vizinhos_terceiro_geral
43:  end for
44:  return (elementos_vizinhos,
45:          vizinhos_geral,
46:          vizinhos_segundo_geral,
47:          vizinhos_terceiro_geral)
48: end procedure

```

3.4.2 Ajuste Quadrático Local de Curvatura (AQFC)

Esta seção descreve a formulação matemática do método AQFC e a forma como ela foi implementada neste trabalho, incluindo a construção do sistema linear associado ao ajuste da quádrlica e a aplicação do método de otimização para a determinação da curvatura.

1. A Função Objetivo

A função objetivo f a ser minimizada para cada nó V é um somatório sobre os seus m vizinhos (V_i, n_i) . Esta função acumula dois termos: um referente a distâncias algébricas quadráticas ponderadas entre os vértices da vizinhança e a superfície implícita $Z(q)$ e outro aos desvios ponderados entre os gradientes da quádrlica ajustada e os vetores normais associados aos vértices da vizinhança.

A função f é dada por:

$$f(a_{11}, \dots, a_{44}) = \sum_{i=1}^m (\omega_{V_i} q(V_i)^2 + \omega_{n_i} \|n_i - \nabla q(V_i)\|^2)$$

Onde:

- $V_i = (x_i, y_i, z_i)$ são as coordenadas do i -ésimo nó vizinho.
- $n_i = (n_{ix}, n_{iy}, n_{iz})$ é a normal do i -ésimo nó vizinho.
- ω_{V_i} e ω_{n_i} são os pesos associados aos termos de vértice e normal, respectivamente. Estes pesos são calculados entre o nó central e o nó vizinho, sendo:

$$\omega_{V_i} = e^{-(\|V - V_i\|^4)}$$
$$\omega_{n_i} = 10^{-4} e^{-\left(\left\|\frac{n}{\|n\|} - \frac{n_i}{\|n_i\|}\right\|^2\right)}$$

- a_{ij} são 10 coeficientes da quádrlica que queremos encontrar.

A quádrlica é definida por:

$$q(x, y, z) = a_{11}x^2 + a_{22}y^2 + a_{33}z^2 + a_{12}xy + a_{13}xz + a_{23}yz + a_{14}x + a_{24}y + a_{34}z + a_{44}$$

2. Expressando $q(V_i)$ e $\nabla q(V_i)$ em Termos dos Coeficientes

Para facilitar a derivação, expressamos $q(V_i)$ e $\nabla q(V_i)$ como funções lineares dos coeficientes a_{ij} .

(a) Representação de $q(x, y, z)$

A função da quádrlica $q(x, y, z)$ pode ser reescrita como um produto escalar entre um vetor de termos polinomiais $c(x, y, z)$ e o vetor de coeficientes a :

$$q(x, y, z) = c(x, y, z) \cdot a$$

Onde:

- $c(x, y, z) = [x^2, y^2, z^2, xy, xz, yz, x, y, z, 1]$ é um vetor linha de 1x10.
- $a = [a_{11}, a_{22}, a_{33}, a_{12}, a_{13}, a_{23}, a_{14}, a_{24}, a_{34}, a_{44}]^T$ é o vetor coluna de 10x1 dos coeficientes.

(b) Representação de $\nabla q(x, y, z)$

O gradiente de $q(x, y, z)$ é um vetor de três componentes, cada uma sendo a derivada parcial de q em relação a x , y e z , respectivamente:

$$\nabla q(x, y, z) = \begin{bmatrix} \frac{\partial q}{\partial x} \\ \frac{\partial q}{\partial y} \\ \frac{\partial q}{\partial z} \end{bmatrix} = \begin{bmatrix} 2a_{11}x + a_{12}y + a_{13}z + a_{14} \\ 2a_{22}y + a_{12}x + a_{23}z + a_{24} \\ 2a_{33}z + a_{13}x + a_{23}y + a_{34} \end{bmatrix}$$

Podemos expressar $\nabla q(x, y, z)$ em forma matricial como o produto de uma matriz $G(x, y, z)$ (3x10) e o vetor de coeficientes a (10x1):

$$\nabla q(x, y, z) = G(x, y, z) \cdot a$$

Onde $G(x, y, z)$ é dada por:

$$G(x, y, z) = \begin{bmatrix} 2x & 0 & 0 & y & z & 0 & 1 & 0 & 0 & 0 \\ 0 & 2y & 0 & x & 0 & z & 0 & 1 & 0 & 0 \\ 0 & 0 & 2z & 0 & x & y & 0 & 0 & 1 & 0 \end{bmatrix}$$

3. Minimizando a Função Objetivo: Derivadas Parciais

Para encontrar os coeficientes a_{ij} que minimizam f , calculamos a derivada parcial de f em relação a cada coeficiente a_k (onde k varia de 1 a 10, correspondendo aos 10 coeficientes) e igualamos a zero. Isso nos dará um sistema de 10 equações lineares para os 10 coeficientes.

$$\frac{\partial f}{\partial a_k} = 0$$

Aplicando a regra da cadeia à função f :

$$\frac{\partial f}{\partial a_k} = \sum_{i=1}^m \left(\omega_{V_i} \frac{\partial}{\partial a_k} (q(V_i)^2) + \omega_{n_i} \frac{\partial}{\partial a_k} (\|n_i - \nabla q(V_i)\|^2) \right) = 0$$

Expandindo os termos:

- **Termo do Vértice:**

$$\frac{\partial}{\partial a_k} (q(V_i)^2) = 2q(V_i) \frac{\partial q(V_i)}{\partial a_k}$$

Como $q(V_i) = c(V_i) \cdot a$, a derivada de $q(V_i)$ em relação a a_k é simplesmente o k -ésimo elemento do vetor $c(V_i)$, que denotamos por $c_{i,k}$.

$$\frac{\partial q(V_i)}{\partial a_k} = c_{i,k}$$

Portanto, o termo do ponto se torna:

$$2\omega_{V_i} q(V_i) c_{i,k}$$

- **Termo da Normal:**

Seja $R_i = n_i - \nabla q(V_i)$. Queremos derivar $\|R_i\|^2 = R_i \cdot R_i$.

$$\frac{\partial}{\partial a_k} (\|n_i - \nabla q(V_i)\|^2) = 2(n_i - \nabla q(V_i)) \cdot \frac{\partial}{\partial a_k} (-\nabla q(V_i)) = -2(n_i - \nabla q(V_i)) \cdot \frac{\partial \nabla q(V_i)}{\partial a_k}$$

Como $\nabla q(V_i) = G(V_i) \cdot a$, a derivada de $\nabla q(V_i)$ em relação a a_k é a k -ésima coluna da matriz $G(V_i)$, que denotamos por $G_{i,k}$.

$$\frac{\partial \nabla q(V_i)}{\partial a_k} = G_{i,k}$$

Portanto, o termo da normal se torna:

$$-2\omega_{n_i}(n_i - \nabla q(V_i)) \cdot G_{i,k}$$

Combinando os termos e dividindo por 2 (que não afeta a solução, pois estamos igualando a zero):

$$\sum_{i=1}^m (\omega_{V_i} q(V_i) c_{i,k} - \omega_{n_i} (n_i - \nabla q(V_i)) \cdot G_{i,k}) = 0$$

Substituindo $q(V_i) = c(V_i) \cdot a$ e $\nabla q(V_i) = G(V_i) \cdot a$:

$$\sum_{i=1}^m (\omega_{V_i} (c(V_i) \cdot a) c_{i,k} - \omega_{n_i} (n_i - G(V_i) \cdot a) \cdot G_{i,k}) = 0$$

Rearranjando para agrupar os termos que dependem de a de um lado e os termos independentes do outro:

$$\sum_{i=1}^m (\omega_{V_i} (c(V_i) \cdot a) c_{i,k} + \omega_{n_i} (G(V_i) \cdot a) \cdot G_{i,k}) = \sum_{i=1}^m \omega_{n_i} n_i \cdot G_{i,k}$$

Esta equação representa a k -ésima linha do nosso sistema linear $Ax = b$.

4. Montagem do Sistema Linear $Ax = b$

O sistema linear $Ax = b$ é formado por uma matriz A (10x10) e um vetor b (10x1), onde x é o vetor de coeficientes a que buscamos.

(a) Construção da Matriz A

A matriz A é a soma das contribuições de cada vizinho i para cada termo da função objetivo. Cada termo é uma matriz 10x10.

$$A = \sum_{i=1}^m (\omega_{V_i} (c(V_i)^T c(V_i)) + \omega_{n_i} (G(V_i)^T G(V_i)))$$

Onde:

- $c(V_i)^T c(V_i)$ é o produto externo do vetor $c(V_i)$ por ele mesmo, resultando em uma matriz 10x10.

- $G(V_i)^T G(V_i)$ é o produto da transposta da matriz $G(V_i)$ (10x3) pela própria matriz $G(V_i)$ (3x10), resultando em uma matriz 10x10.

(b) **Construção do Vetor b**

O vetor b é a soma das contribuições de cada vizinho i para o lado direito da equação, que contém apenas os termos independentes dos coeficientes a .

$$b = \sum_{i=1}^m \omega_{n_i} (G(V_i)^T n_i)$$

Onde:

- $G(V_i)^T n_i$ é o produto da transposta da matriz $G(V_i)$ (10x3) pelo vetor normal n_i (3x1), resultando em um vetor 10x1.

5. Otimização para Projeção do Ponto

Após a determinação dos coeficientes da quádrlica, obtém-se uma função implícita $q(x, y, z) = 0$ que define uma superfície quadrática ajustada aos pontos da vizinhança. Como essa superfície é obtida por um processo de ajuste, ela não necessariamente interpola os pontos V_i , e, em particular, o ponto central V não pertence, em geral, à superfície $Z(q)$.

Dessa forma, as propriedades geométricas devem ser avaliadas em um ponto pertencente à superfície ajustada. Para isso, determina-se o ponto $p(V)$ correspondente à projeção ortogonal de V sobre a quádrlica. A utilização do ponto projetado $p(V)$ é fundamental para evitar erros na avaliação das quantidades geométricas, uma vez que o cálculo direto no ponto V introduziria distorções no gradiente da função implícita, as quais são amplificadas nas expressões de curvatura.

Então o próximo passo é encontrar o ponto $p(V)$, que é o ponto da superfície ajustada que está mais próximo do ponto V da malha. Isso é feito através de um problema de otimização:

$$\min_{p(V)} \|V - p(V)\|^2 \quad \text{sujeito a} \quad q(p(V)) = 0$$

Em outras palavras, substitui-se os coeficientes a_{ij} locais na equação 3.1, o problema é restrito a condição $q(x, y, z) = 0$ e o método de otimização deve realizar suas iterações buscando os valores de (x, y, z) que satisfazem esta condição. O ponto inicial para a otimização é o próprio ponto central V e os valores encontrados serão as coordenadas do ponto $p(V)$.

Método de Otimização: Sequential Least Squares Programming (SLSQP)

O método utilizado foi o **SLSQP** (Sequential Least Squares Programming) que é um algoritmo de otimização não linear que lida com funções objetivo e restrições que podem ser não lineares. Ele é particularmente adequado para problemas de otimização de pequena e média escala com restrições de igualdade e desigualdade. O **SLSQP** opera em uma abordagem iterativa, onde a cada iteração, o problema não linear original é aproximado por uma sequência de problemas de programação quadrática (QP) ou programação linear (LP), que são mais fáceis de resolver. Os passos gerais do ‘SLSQP’ incluem:

- (a) **Inicialização:** Começa com um ponto inicial x_0 (no nosso caso, o ponto central V).
- (b) **Linearização:** Em cada iteração, a função objetivo e as funções de restrição são linearizadas em torno do ponto atual. Isso transforma o problema não linear em um problema de programação quadrática (QP) local.
- (c) **Resolução do QP:** O problema QP linearizado é resolvido para encontrar uma direção de busca d_k .
- (d) **Busca Linear:** Uma busca linear é realizada ao longo da direção d_k para encontrar um novo ponto x_{k+1} que melhore a função objetivo e satisfaça as restrições, ou pelo menos reduza a violação das restrições.
- (e) **Atualização:** O ponto é atualizado, e o processo se repete até que um critério de convergência seja satisfeito (por exemplo, o gradiente da função objetivo é próximo de zero e as restrições são satisfeitas dentro de uma tolerância).

Uma característica importante do **SLSQP** é sua capacidade de lidar com restrições de igualdade e desigualdade, o que o torna versátil para uma ampla

gama de problemas. No contexto deste trabalho, a restrição $q(P_V) = 0$ é uma restrição de igualdade crucial para garantir que o ponto projetado esteja na superfície da quádrlica.

6. Cálculo da Curvatura Média

A curvatura média H é calculada no ponto projetado $p(V)$ usando a fórmula para superfícies implícitas, que depende do gradiente ∇q e da matriz Hessiana $\mathbf{H}_q$ da quádrlica.

Primeiramente, é calculado o vetor $\mathbf{s}(n, V)$, que é o gradiente $\nabla q(p(V))$ reescalado pelo módulo da normal do nó central $\|n\|$:

$$\mathbf{s}(n, V) = \|n\| \frac{\nabla q(p(V))}{\|\nabla q(p(V))\|}$$

A matriz Hessiana $\mathbf{H}_q$ é a matriz das segundas derivadas parciais de q , que é constante para uma quádrlica:

$$\mathbf{H}_q = \begin{bmatrix} 2a_{11} & a_{12} & a_{13} \\ a_{12} & 2a_{22} & a_{23} \\ a_{13} & a_{23} & 2a_{33} \end{bmatrix}$$

A curvatura média H é então calculada pela fórmula:

$$H = \frac{\mathbf{s}(n, V)^T \mathbf{H}_q(p(V)) \mathbf{s}(n, V) - \|\mathbf{s}(n, V)\|^2 \text{tr}(\mathbf{H}_q(p(V)))}{2\|\mathbf{s}(n, V)\|^3}$$

onde $\text{tr}(\mathbf{H}_q)$ é o traço da matriz Hessiana.

Pseudocódigo AQFC

Algoritmo 3 Pseudocódigo AQFC — Cálculo de Curvatura via Ajuste Quadrático

- 1: // Defina as funções numéricas q , ∇q e $H(q)$
- 2: **procedure** $\mathbf{q}(x, y, z, a)$ ▷ Calcula o valor da quádrlica $q(x, y, z)$ dados os coeficientes a
- 3: **return** $a[0] x^2 + a[1] y^2 + a[2] z^2 + a[3] xy + a[4] xz + a[5] yz + a[6] x + a[7] y + a[8] z + a[9]$
- 4: **end procedure**
- 5: **procedure** $\nabla q(x, y, z, a)$ ▷ Calcula o vetor gradiente de $q(x, y, z)$ dados os coeficientes a

```

6:    $dq_{dx} \leftarrow 2a[0]x + a[3]y + a[4]z + a[6]$ 
7:    $dq_{dy} \leftarrow 2a[1]y + a[3]x + a[5]z + a[7]$ 
8:    $dq_{dz} \leftarrow 2a[2]z + a[4]x + a[5]y + a[8]$ 
9:   return  $[dq_{dx}, dq_{dy}, dq_{dz}]$ 
10: end procedure
11: procedure  $Hq(a)$  ▷ Calcula a matriz Hessiana de  $q(x, y, z)$  dados os
    coeficientes  $a$ 
12:    $d2q_{dx2} \leftarrow 2a[0]$ 
13:    $d2q_{dy2} \leftarrow 2a[1]$ 
14:    $d2q_{dz2} \leftarrow 2a[2]$ 
15:    $d2q_{dxdy} \leftarrow a[3]$ 
16:    $d2q_{dx dz} \leftarrow a[4]$ 
17:    $d2q_{dy dz} \leftarrow a[5]$ 
18:    $H \leftarrow \begin{bmatrix} d2q_{dx2} & d2q_{dxdy} & d2q_{dx dz} \\ d2q_{dxdy} & d2q_{dy2} & d2q_{dy dz} \\ d2q_{dx dz} & d2q_{dy dz} & d2q_{dz2} \end{bmatrix}$ 
19:   return  $H$ 
20: end procedure
21: // Código Principal
22: procedure  $AQFC\_CURVATURE(V, neighbors, normals)$ 
23:   ▷  $V$ : Coordenadas  $(x, y, z)$  dos Vértices  $V$ 
24:   ▷  $neighbors$ : Lista com os nós vizinhos
25:   ▷  $normals$ : Lista com as normais
26:    $n \leftarrow normals$ 
27:   for cada nó  $V$  do
28:     Inicialize a matriz  $A$  (10x10) e o vetor  $b$  (10x1) com zeros
29:      $A \leftarrow \mathbf{0}_{10 \times 10}$ 
30:      $b \leftarrow \mathbf{0}_{10}$ 
31:     for cada vizinho  $V_i$  de  $V$  do
32:       Calcule os pesos  $\omega_{V_i}$  e  $\omega_{n_i}$ 
33:        $\omega_{Vi} \leftarrow \exp(-\|V - V_i\|^4)$ 
34:        $\omega_{ni} \leftarrow 10^{-4} \exp(-\left\|\frac{n}{\|n\|} - \frac{n_i}{\|n_i\|}\right\|^2)$ 
35:       Defina o vetor  $c(V_i)$  e a matriz  $G(V_i)$ 

```

```

36:       $c(V\_i) \leftarrow [xi^2, yi^2, zi^2, xi\ yi, xi\ zi, yi\ zi, xi, yi, zi, 1.0]$ 
37:       $G(V\_i) \leftarrow \begin{bmatrix} 2\ xi & 0 & 0 & yi & zi & 0 & 1 & 0 & 0 & 0 \\ 0 & 2\ yi & 0 & xi & 0 & zi & 0 & 1 & 0 & 0 \\ 0 & 0 & 2\ zi & 0 & xi & yi & 0 & 0 & 1 & 0 \end{bmatrix}$ 
38:      Acumule as contribuições em  $A$  e  $b$ :
39:       $A \leftarrow A + \omega_{V_i}(c(V_i)^T c(V_i)) + \omega_{n_i}(G(V_i)^T G(V_i))$ 
40:       $b \leftarrow b + \omega_{n_i}(G(V_i)^T n_i)$ 
41:  end for
42:  Resolva o sistema linear  $Ax = b$  para obter os coeficientes  $a$ 
43:  Defina a função restrição  $q(x, y, z, a) = 0$  usando os coeficientes  $a$  encontrados
44:  Utilize um algoritmo de otimização para encontrar o ponto da superfície que satisfaça a restrição e utilize o ponto original  $V$  como chute inicial.
45:  Armazene o ponto encontrado  $PV$ .
46:  Calcule a curvatura média  $H$  usando  $\nabla q$  e  $Hq$  com os coeficientes  $a$ 
47:  // Cálculo da curvatura média  $H$ 
48:   $grad\_at\_PV \leftarrow \nabla q(PV[0], PV[1], PV[2], a)$ 
49:   $norm\_grad \leftarrow \|\nabla q(PV[0], PV[1], PV[2], a)\|$ 
50:   $s \leftarrow \frac{\|n\| \nabla q(PV[0], PV[1], PV[2], a)}{norm\_grad} \triangleright$  vetor  $s$  na direção do gradiente normalizado
51:   $hess\_num \leftarrow \text{HESS\_Q\_FUNC\_NUMERIC}(a)$ 
52:   $H \leftarrow \frac{s^T hess\_num s}{2 \|s\|^3} - \frac{\|s\|^2 \text{trace}(hess\_num)}{2 \|s\|^3}$ 
53:  Armazene a curvatura média calculada  $H$ .
54:  end for
55:  return Curvatura  $H$  para cada ponto da malha.
56: end procedure

```

3.4.3 Operador Laplace-Beltrami Discreto (LB)

O código implementa numericamente a teoria apresentada em 3.3.2, aplicando o operador Laplace-Beltrami discretizado sobre uma malha triangular definida por um conjunto de vértices (X, Y, Z) e pela matriz de conectividade (IEN) . O objetivo é obter a curvatura média em cada vértice da malha.

A formulação segue a discretização por elementos finitos, onde a curvatura é aproximada a partir da multiplicação do operador Laplaciano pela posição dos vértices. Além disso, uma etapa adicional foi incorporada para corrigir as áreas de influência em triângulos obtusos, conforme sugerido em [4], garantindo maior precisão numérica.

1. Construção da Matriz Laplaciana e Áreas de Influência:

O algoritmo percorre todos os triângulos da malha para calcular:

- (a) **Matriz do Laplaciano local:** Para cada triângulo, é construída uma matriz 3×3 usando projeções vetoriais das arestas no plano local.
- (b) **Áreas de influência locais:** São calculadas as áreas mistas de Voronoi/baricêntricas associadas a cada vértice do triângulo.
- (c) **Correção para triângulos obtusos:** Caso o triângulo seja obtuso, as áreas de influência são ajustadas proporcionalmente às arestas, conforme visto em 2.2.3.
- (d) **Armazenamento em formato de tripletos:** As entradas das matrizes locais são armazenadas no formato $(i, j, valor)$, que posteriormente é usado para montar a matriz Laplaciana global esparsa.

2. Aplicação do Operador e Cálculo da Curvatura Média:

- (a) **Montagem da matriz global:** A partir dos tripletos, constrói-se a matriz esparsa global $\mathbf{K}$.
- (b) **Aplicação do operador:** O operador é aplicado aos vetores de coordenadas (X, Y, Z) dos vértices:

$$\mathbf{k}_x = -\mathbf{KX}, \quad \mathbf{k}_y = -\mathbf{KY}, \quad \mathbf{k}_z = -\mathbf{KZ}$$

- (c) **Magnitude do vetor de curvatura:** A magnitude do vetor resultante é dada por:

$$\|\kappa\| = \sqrt{k_x^2 + k_y^2 + k_z^2}$$

- (d) **Cálculo da curvatura média:** A curvatura média em cada vértice é obtida por:

$$H_i = \frac{\|\mathbf{K}_i\|}{2A_i}$$

onde A_i é a área de influência do vértice i .

- (e) **Determinação do sinal:** O sinal da curvatura é definido pelo produto escalar entre o vetor de curvatura (k_x, k_y, k_z) e a normal $\mathbf{n}$ no vértice:

$$\text{se } (\kappa \cdot \mathbf{n}) > 0, \quad \text{então } \kappa_i \leftarrow -\kappa_i$$

Pseudocódigo LB

Algoritmo 4 Cálculo de Curvatura Média via Laplace-Beltrami Discreto

1: ▷ **Função para Área Circuncêntrica**

2: **procedure** CALC_A(no_1, no_2, no_3)

3: Calcular lados a, b, c do triângulo entre os vértices

4: Calcular pesos p_a, p_b, p_c com:

$$p_a = a^2(b^2 + c^2 - a^2)$$

$$p_b = b^2(a^2 + c^2 - b^2)$$

$$p_c = c^2(a^2 + b^2 - c^2)$$

5: $sum \leftarrow p_a + p_b + p_c$

6: Calcular centro $C = (p_a no_1 + p_b no_2 + p_c no_3) / sum$

7: Calcular pontos médios m_1, m_2 e vetores t_1, t_2

8: $P_{12} \leftarrow (no_1 + no_2) / 2$

9: $P_{13} \leftarrow (no_1 + no_3) / 2$

10: $m_1 \leftarrow C - P_{13}$

11: $m_2 \leftarrow C - P_{12}$

12: $dt_1 \leftarrow \|no_3 - no_1\|$

13: $t_1 \leftarrow (no_3 - no_1) / \|dt_1\|$

14: $dt_2 \leftarrow \|no_2 - no_1\|$

```

15:    $t_2 \leftarrow (no_2 - no_1) / \|dt_2\|$ 
16:    $t_1 \leftarrow t_1 dt_1 / 2$ 
17:    $t_2 \leftarrow t_2 dt_2 / 2$ 
18:    $A \leftarrow 0.5 \times (\|t_1 \times m_1\| + \|t_2 \times m_2\|)$ 
19:   return  $A$ 
20: end procedure

21:                                      $\triangleright$  Cálculo da Matriz Local e Áreas de Influência
22: procedure CALCULARRIGIDEZLOCALEAREAS( $V, IEN$ )
23:                                      $\triangleright V$ : lista de coordenadas  $(x, y, z)$  dos vértices
24:                                      $\triangleright IEN$ : matriz de conectividade (lista de triângulos com 3 índices de
vértices)
25:    $K\_tripletos \leftarrow$  lista vazia
26:    $AreasInfluencia \leftarrow$  vetor de zeros com tamanho  $|V|$ 
27:   for cada triângulo  $T = (no_1, no_2, no_3)$  em  $IEN$  do
28:                                      $\triangleright no_i$ : coordenadas  $(x, y, z)$  dos vértices do triângulo
29:      $ji \leftarrow no_2 - no_1$ 
30:      $ik \leftarrow no_1 - no_3$ 
31:      $kj \leftarrow no_3 - no_2$ 
32:      $ksi \leftarrow \text{Normalizar}(ji)$ 
33:      $pvet1 \leftarrow ji \times ik$ 
34:      $pvet2 \leftarrow pvet1 \times ji$ 
35:      $eta \leftarrow \text{Normalizar}(pvet2)$ 
36:                                      $\triangleright$  Componentes geométricos
37:      $b_i \leftarrow kj \cdot eta$ 
38:      $b_j \leftarrow ik \cdot eta$ 
39:      $b_k \leftarrow ji \cdot eta$ 
40:      $c_i \leftarrow kj \cdot ksi$ 
41:      $c_j \leftarrow ik \cdot ksi$ 
42:      $c_k \leftarrow ji \cdot ksi$ 
43:      $area \leftarrow 0.5 \times \|pvet1\|$ 
44:                                      $\triangleright$  Áreas circuncêntricas de cada vértice
45:      $a_1 \leftarrow \text{CALC\_A}(no_1, no_2, no_3)$ 

```



```

46:       $a_2 \leftarrow \text{CALC\_A}(no_2, no_3, no_1)$ 
47:       $a_3 \leftarrow \text{CALC\_A}(no_3, no_1, no_2)$ 
48:                                     ▷ Correção para triângulos obtusos
49:      if  $(a_1 + a_2 + a_3) > \text{area}$  then
50:                                     ▷ Comprimento das arestas
51:           $la \leftarrow \|kj\|$ 
52:           $lb \leftarrow \|ik\|$ 
53:           $lc \leftarrow \|ji\|$ 
54:          Corrigir area para os dois vértices que estão diretamente conectados
          à maior aresta  $a_i$ , como:
55:           $a_i \leftarrow \frac{\text{aresta menor conectada ao vértice } i}{\text{soma das duas arestas menores}} (\text{area} - a_i \text{ da aresta de maior angulo})$ 
56:      end if
57:      Somar áreas para seus respectivos vértices em AreasInfluencia
58:                                     ▷ Construção das matrizes locais
59:      for  $p$  in  $[0, 1, 2]$  do
60:          for  $q$  in  $[0, 1, 2]$  do
61:               $K_b \leftarrow (b_p b_q) / (4 \text{area})$                                      ▷ Sendo:  $b_0 = b_i, b_1 = b_j, b_2 = b_k$ .
62:               $K_c \leftarrow (c_p c_q) / (4 \text{area})$                                      ▷ Sendo:  $c_0 = c_i, c_1 = c_j, c_2 = c_k$ .
63:               $K_{local} \leftarrow K_b + K_c$ 
64:              Armazenar  $p, q$  e  $K_{local}$  em  $K\_tripletos$ 
65:          end for
66:      end for
67:  end for
68:  return  $(K\_tripletos, AreasInfluencia)$ 
69: end procedure

70:                                     ▷ Função principal
71: procedure LAPLACEBELTRAMICURVATURE( $V, IEN, Normais$ )
72:                                     ▷  $V$ : lista de coordenadas  $(x, y, z)$  dos vértices
73:                                     ▷  $IEN$ : matriz de conectividade (lista de triângulos com 3 índices de
          vértices)
74:                                     ▷  $Normais$ : vetor de normais médias por vértice
75:  Obter  $K\_tripletos$  e  $AreasInfluencia$  pela função criada

```

```

76:    $K \leftarrow \text{MONTARMATRIZESPARGA}(K\_tripletos, \quad (\text{numero\_de\_V} \times$ 
       $\text{numero\_de\_V}))$ 
77:    $X, Y, Z \leftarrow \text{componentes de } V$ 
78:    $Kx \leftarrow -K \times X$ 
79:    $Ky \leftarrow -K \times Y$ 
80:    $Kz \leftarrow -K \times Z$ 
81:    $H \leftarrow \text{vetor de zeros com tamanho } V$ 
82:   for  $i = 0$  até  $|\text{numero\_de\_V}| - 1$  do
83:      $Kappa \leftarrow (Kx[i], Ky[i], Kz[i])$ 
84:      $mod \leftarrow \text{Norma}(Kappa)$ 
85:     if  $\text{AreasInfluencia}[i] \neq 0$  then
86:        $\kappa \leftarrow mod / \text{AreasInfluencia}[i]$ 
87:     else
88:        $\kappa \leftarrow 0$ 
89:     end if
90:                                      $\triangleright$  Determinação do sinal
91:     if  $\text{ProdutoEscalar}(Kappa, \text{Normais}[i]) > 0$  then
92:        $\kappa \leftarrow -\kappa$ 
93:     end if
94:      $H[i] \leftarrow \kappa / 2$ 
95:   end for
96:   return  $H$ 
97: end procedure

```

3.5 Curvatura Analítica de Referência

As equações conhecidas para calcular os valores analíticos das curvaturas médias das geometrias de referência foram:

Esfera

Equação da superfície:

$$x^2 + y^2 + z^2 = R^2 \quad \implies \quad F(x, y, z) = x^2 + y^2 + z^2 - R^2 = 0 \quad (3.27)$$

Curvatura média:

$$H = \frac{1}{R} \quad (3.28)$$

Parâmetro: $R = 0.5$

Valores extremos:

$$H_{\min} = H_{\max} = 2.0$$

Elipsoide

Equação da superfície:

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 + \left(\frac{z}{c}\right)^2 = 1 \quad \implies \quad F(x, y, z) = \left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 + \left(\frac{z}{c}\right)^2 - 1 = 0 \quad (3.29)$$

Curvatura média:

$$p = \sqrt{\frac{x^2}{a^4} + \frac{y^2}{b^4} + \frac{z^2}{c^4}} \quad (3.30)$$

$$H = \frac{1}{2p} \left(\frac{1}{a^2} + \frac{1}{b^2} + \frac{1}{c^2} \right) - \frac{1}{2p^3} \left(\frac{x^2}{a^6} + \frac{y^2}{b^6} + \frac{z^2}{c^6} \right) \quad (3.31)$$

Parâmetros: $a = 10.0$ $b = 4.0$ $c = 4.0$

Valores extremos:

$$H_{\min} \approx 0.145 \quad H_{\max} \approx 0.625$$

Toro

Equação da superfície:

$$\left(\sqrt{x^2 + y^2} - R\right)^2 + z^2 = r^2 \implies F(x, y, z) = \left(\sqrt{x^2 + y^2} - R\right)^2 + z^2 - r^2 = 0 \quad (3.32)$$

Curvatura média:

$$\rho = \sqrt{x^2 + y^2} \quad (3.33)$$

$$H = \frac{2\rho - R}{2r\rho} \quad (3.34)$$

Parâmetros: $R = 0.5$ $r = 0.4$

Valores extremos:

$$H_{\min} \approx -3.750 \quad H_{\max} \approx 1.806$$

Cilindro

Equação da superfície:

$$x^2 + y^2 = r^2 \implies F(x, y, z) = x^2 + y^2 - r^2 = 0 \quad (3.35)$$

Curvaturas principais:

$$\kappa_1 = \frac{1}{r}, \quad \kappa_2 = 0 \quad (3.36)$$

Curvatura média:

$$H = \frac{1}{2}(\kappa_1 + \kappa_2) = \frac{1}{2} \left(\frac{1}{r} + 0 \right) = \frac{1}{2r} \quad (3.37)$$

Observação: A curvatura média está definida apenas em pontos regulares da superfície, onde esta é diferenciável. Em regiões de borda, a superfície não é suave, e portanto não está definida nesses pontos.

Parâmetro: $r = 0.5$

Valores extremos:

$$H_{\max} = 1.0 \quad H_{\min} = 0$$

3.6 Métricas de Erro

Em problemas de métodos numéricos e simulação computacional, a avaliação da precisão de um método requer a comparação entre a solução numérica e uma solução de referência considerada exata ou analítica. Dessa forma, foram adotadas três métricas clássicas de erro, capazes de caracterizar diferentes aspectos da discrepância entre as curvaturas médias estimadas numericamente e os valores analíticos de referência.

Sejam $\kappa_{n,i}$ os valores estimados numericamente, $\kappa_{a,i}$ os valores analíticos, e N o número de pontos válidos. As métricas definidas são:

Erro Relativo na Norma L_2

A norma L_2 é uma das métricas mais utilizadas na análise de erro de métodos numéricos, pois fornece uma medida global da discrepância quadrática entre duas soluções. Em discretizações numéricas, a versão relativa dessa norma é frequentemente empregada para permitir comparações independentes da escala do problema.

$$e_{L2} = \sqrt{\frac{\sum_{i=1}^N (\kappa_{n,i} - \kappa_{a,i})^2}{\sum_{i=1}^N \kappa_{a,i}^2}} \quad (3.38)$$

Erro Relativo Médio

O erro relativo médio mede a discrepância percentual média ponto a ponto entre as soluções numérica e analítica, o que fornece uma interpretação direta da precisão média local do método

$$e_{\text{mean}} = \frac{1}{N} \sum_{i=1}^N \left| \frac{\kappa_{n,i} - \kappa_{a,i}}{\kappa_{a,i}} \right| \quad (3.39)$$

Erro Relativo Máximo

O erro relativo máximo corresponde ao maior desvio relativo observado no domínio e está associado à norma L_∞ . Essa métrica é fundamental para identificar regiões críticas onde o método apresenta pior desempenho.

$$e_{\max} = \max_{1 \leq i \leq N} \left| \frac{\kappa_{n,i} - \kappa_{a,i}}{\kappa_{a,i}} \right| \quad (3.40)$$

Em regiões onde a curvatura analítica apresenta valores próximos de zero ($|\kappa_{a,i}| < \varepsilon$), as métricas baseadas em erro relativo tornam-se numericamente instáveis devido à divisão por valores muito pequenos. Nesses casos, adotou-se uma das seguintes estratégias, dependendo da análise realizada:

- utilização do erro absoluto;
- marcação do valor como **NaN** (*Not a Number*), indicando que o erro relativo não é definido naquele ponto.

A utilização de **NaN** permite preservar a consistência estatística das métricas, evitando que valores indefinidos distorçam as médias e os máximos globais.

3.7 Métricas de Qualidade da Malha

A avaliação da qualidade da malha é uma etapa fundamental em qualquer análise numérica baseada em elementos discretos, e quando o objetivo é estimar propriedades locais de curvatura sobre uma superfície triangular, a regularidade dos elementos que compõem a malha torna-se determinante para a confiabilidade dos métodos empregados. Assim, o uso de métricas de qualidade permite quantificar a distorção geométrica dos triângulos e identificar regiões onde o refinamento ou o uso de métodos mais robustos de cálculo de curvatura se faz necessário.

Razão Incentro/Circuncentro (r/R)

A razão entre o raio do círculo inscrito r e o raio do círculo circunscrito R , comumente denotada por ρ , é uma métrica clássica de qualidade de triângulos. Ela quantifica quão próximo o elemento está da forma ideal (triângulo equilátero) [35, 34]. Para um triângulo com arestas de comprimento a, b, c e área A define-se:

$$s = \frac{a + b + c}{2} \quad (\text{semi-perímetro}) \quad (3.41)$$

$$r = \frac{A}{s} \quad (\text{raio inscrito}) \quad (3.42)$$

$$R = \frac{a b c}{4A} \quad (\text{raio circunscrito}) \quad (3.43)$$

$$\rho = \frac{r}{R} \quad (3.44)$$

Interpretação:

- $\rho = 0.5$ para triângulos equiláteros (melhor caso).
- $\rho \rightarrow 0$ para triângulos degenerados (um ângulo muito pequeno ou elemento muito alongado).

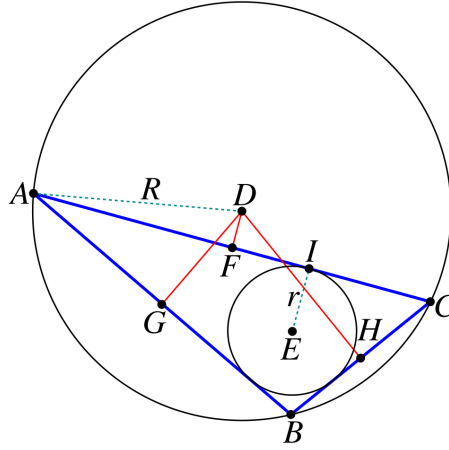


Figura 3.3: Raio inscrito r e circunscrito R de um triângulo (esquema ilustrativo).

Distorção Angular (*Angular Skewness*)

A distorção angular mede o desvio dos ângulos internos de um triângulo em relação ao ângulo ideal de 60° . Uma formulação comum é:

$$\text{skew} = \max\left(\frac{\theta_{\max} - \theta_{\text{equi}}}{180^\circ - \theta_{\text{equi}}}, \frac{\theta_{\text{equi}} - \theta_{\min}}{\theta_{\text{equi}} - 0^\circ}\right) \quad \text{com } \theta_{\text{equi}} = 60^\circ.$$

O valor varia em $[0, 1]$: 0 para triângulo equilátero, 1 para triângulo degenerado. Esta métrica é explicitada em manuais de meshing e ferramentas comerciais [39].

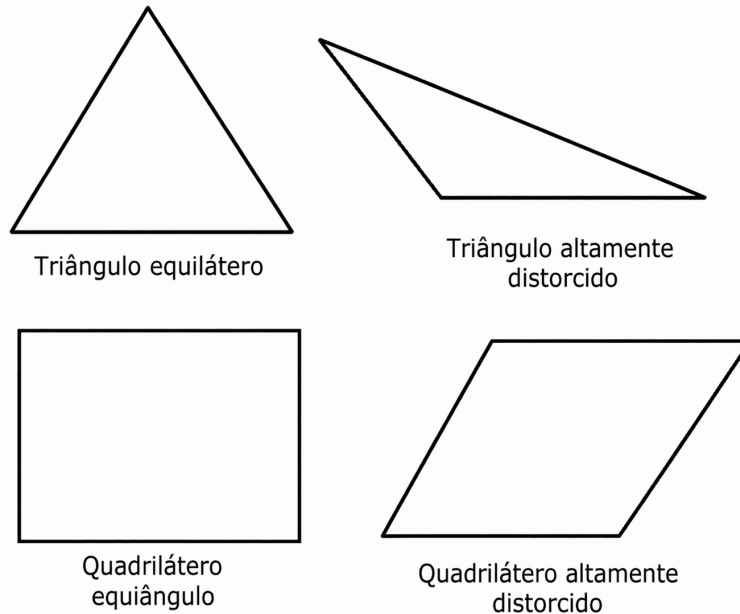


Figura 3.4: Comparação entre elementos equiláteros e elementos distorcidos.

Razão de Aspecto

Uma definição prática e simples de razão de aspecto é a razão entre a maior e a menor aresta do triângulo:

$$AR = \frac{\text{aresta}_{\max}}{\text{aresta}_{\min}}.$$

Outras definições equivalentes relacionam as normais nas arestas e outras variantes algebraicas também são usadas na literatura de qualidade de malha [36, 40].

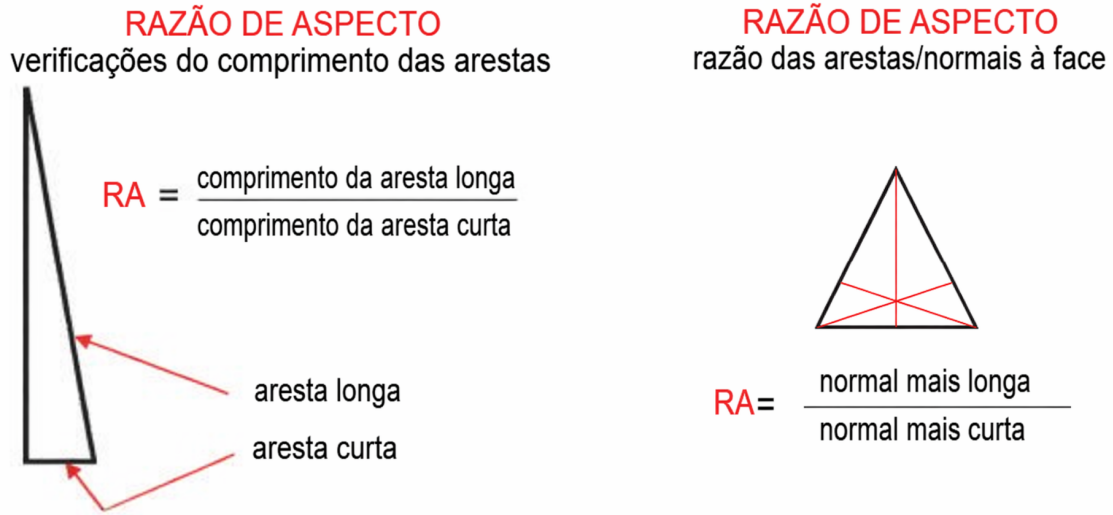


Figura 3.5: Exemplos de diferentes definições de razões de aspecto em triângulos.

Interpretação:

- $AR = 1$: triângulo equilátero (ideal).
- $AR \gg 1$: elemento alongado, potencialmente problemático.

Qualidade Composta

Embora métricas individuais sejam eficazes para caracterizar aspectos específicos da geometria dos elementos, é frequentemente conveniente dispor de um único escalar que sintetize diferentes critérios de qualidade da malha. Nesse contexto, define-se uma métrica composta baseada nas grandezas previamente introduzidas.

Sejam:

- $\overline{\text{skew}}$: média da distorção angular entre os elementos associados ao nó;

- $\min(\rho = r/R)$: menor razão incentro/circuncentro entre os elementos associados ao nó (destaca os piores casos);
- $\max(AR)$: maior razão de aspecto entre os elementos associados ao nó;
- $\max(\delta_A)$: desvio relativo máximo das áreas dos elementos adjacentes a um nó em relação à média local.

A grandeza δ é definida, para cada nó i , como:

$$\max(\delta_{A_i}) = \frac{\max_j |A_j - \bar{A}_i|}{\bar{A}_i}, \quad (3.45)$$

onde A_j são as áreas dos elementos vizinhos ao nó e $\bar{A}_i$ é a média dessas áreas.

A qualidade composta é definida como:

$$Q = w_1 (1 - [\overline{\text{skew}}]_{[0,1]}) + w_2 [\min(\rho)]_{[0,0.5]} + w_3 \left(\frac{1}{1 + [\max(AR)]_{[1,10]}} \right) + w_4 (1 - [\delta_A]_{[0,1]}), \quad (3.46)$$

onde $[\text{métrica}]_{[a,b]}$ representa a limitação dos valores de cada métrica dentro do intervalo de a até b .

Os pesos w_i permitem ajustar a importância relativa de cada métrica. Neste trabalho, foram adotados os seguintes valores:

$$w_1 = 0.4, \quad w_2 = 0.3, \quad w_3 = 0.2, \quad w_4 = 0.1,$$

com $\sum_{i=1}^4 w_i = 1$.

A construção da métrica garante que valores maiores de Q correspondam a elementos de melhor qualidade:

- $1 - \overline{\text{skew}}$: penaliza alta distorção angular, quanto menor $\overline{\text{skew}}$ melhor;
- $\min(\rho)$: favorece triângulos próximos do equilátero, quanto maior ρ melhor;
- $\frac{1}{1 + \max(AR)}$: reduz o impacto de elementos alongados, quanto menor AR melhor;
- $1 - \max(\delta_A)$: penaliza a não-uniformidade das áreas no entorno do nó (se tiver alguma área com o dobro do tamanho da média, o termo é anulado), quanto menor δ_A melhor.

Para permitir a comparação entre diferentes malhas, a métrica é normalizada para o intervalo $[0, 1]$:

$$Q_{\text{norm}} = \frac{Q - Q_{\min}}{Q_{\max} - Q_{\min} + \varepsilon}, \quad (3.47)$$

onde ε é um parâmetro pequeno (10^{-12}) introduzido para evitar divisão por zero.

Interpretação:

- $Q_{\text{norm}} = 1$ para região da malha com boa qualidade local.
- $Q_{\text{norm}} \rightarrow 0$ para região de baixa qualidade local.

Capítulo 4

Resultados e Discussões

4.1 Configuração dos Testes e Métricas de Avaliação

Esta seção apresenta os resultados quantitativos da estimativa de curvatura média em malhas triangulares por dois grupos de métodos: (i) a discretização do operador Laplace–Beltrami (LB) e (ii) o Ajuste Quadrático de Curvatura (AQFC) com três tamanhos de vizinhança, definidos por $k = 1, 2, 3$ anéis (AQFC1, AQFC2 e AQFC3). Os métodos foram aplicados às geometrias Esfera, Elipsoide 1, Elipsoide 2, Toro e Cilindro, com diversas malhas variando do fator de tamanho de elemento h , de modo a avaliar simultaneamente acurácia e custo computacional em função do refinamento.

Nesta análise, adotam-se como métricas principais: (a) o erro relativo na norma L_2 , por representar o comportamento global, e (b) o erro relativo máximo, por caracterizar o pior caso local. A métrica de erro relativo médio é apresentada como suporte à interpretação dos resultados. As tabelas completas com informações das malhas, tempos de execução e métricas de erro encontram-se no Apêndice B.

Tratamento de Regiões Singulares e Valores de Referência Próximos de Zero

Em geometrias com singularidades ou referência analítica próxima de zero, a avaliação por erro relativo pode ser altamente impactada, o que gera uma necessidade de filtrar os dados.

Assim, no toro, vértices onde havia o valor de curvatura analítica próxima a zero (região onde troca a concavidade), foram marcados como zero literal e ignorados nos cálculos dos erros, como pode ser visto na tabela B.13.

Já para o cilindro, a análise tornou-se mais delicada, pois, nas bordas, a curvatura teórica tende ao infinito e esses pontos precisam ser desconsiderados na avaliação do erro. Como o método AQFC depende fortemente das normais e da configuração geométrica da vizinhança de cada nó, os valores das normais nos nós de borda afetam também os nós adjacentes. Assim, não apenas os pontos de borda são removidos, mas também aqueles cuja vizinhança utilizada no ajuste quadrático inclui tais pontos.

Esse comportamento não se manifesta no método baseado no operador Laplace–Beltrami, que apresenta menor dependência da vizinhança. Trata-se, portanto, de uma limitação inerente ao AQFC, já que o ajuste quadrático deixa de ser adequado em regiões onde a superfície não é localmente contínua.

Dessa forma, para garantir uma comparação justa, os resultados do método LB foram salvos desconsiderando exatamente os mesmos pontos removidos nas estimativas AQFC com vizinhanças equivalentes. São chamados de LB1, LB2 e LB3, aqueles que consideram os mesmos pontos válidos referentes ao AQFC1, AQFC2, e AQFC3, respectivamente. O LB refere-se aos dados onde somente as bordas do cilindro foram desconsideradas e os outros são referentes ao seu respectivo número do AQFC, de acordo com os pontos vizinhança das bordas também ignorados. O gráfico 4.1 mostra esse percentual de pontos ignorados no cilindro para cada método, e de acordo com o refino da malha.

Ainda assim, essa filtragem teve impacto mínimo nos resultados do LB, como evidenciado nos gráficos de erros que serão apresentados adiante.

CILINDRO - Percentual de Pontos Ignorados x Malha h

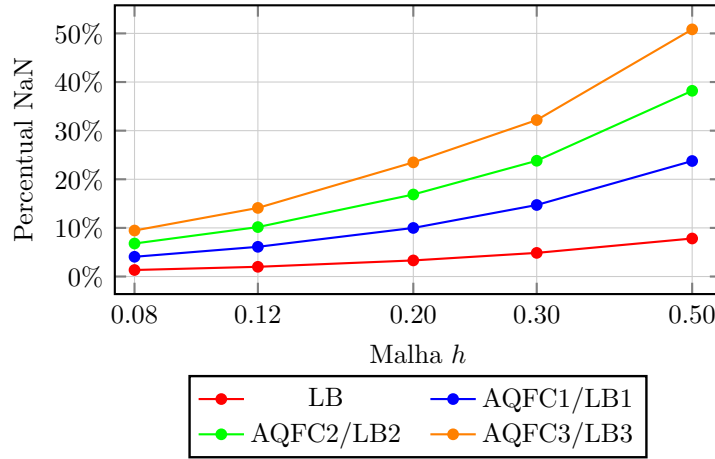


Figura 4.1: CILINDRO: Percentual de Pontos Ignorados x Malha h

4.2 Desempenho Computacional

A avaliação do desempenho computacional dos métodos constitui etapa fundamental para a análise comparativa, uma vez que a escolha de um estimador de curvatura não depende exclusivamente da precisão obtida, mas também do custo associado ao seu cálculo. Nesta seção, analisa-se o comportamento temporal dos métodos Laplace–Beltrami (LB) e Ajuste Quadrático de Curvatura (AQFC) em função do refinamento da malha e do tamanho da vizinhança considerada no AQFC.

Os tempos reportados referem-se exclusivamente à etapa de cálculo da curvatura média, considerando que as normais e as estruturas de vizinhança foram previamente computadas. Essa padronização experimental permitiu maior eficiência na coleta dos dados experimentais, assim como isolar o efeito da ampliação da vizinhança no custo do ajuste quadrático do AQFC, evitando que o tempo de construção das vizinhanças interferisse diretamente na comparação.

Ressalta-se que essa escolha tende a favorecer proporcionalmente os métodos AQFC, uma vez que estes dependem explicitamente da definição e manipulação das vizinhanças, enquanto o método LB utiliza apenas as normais previamente calculadas. Em experimentos complementares, nos quais o tempo total de execução foi considerado (incluindo a construção das vizinhanças), observou-se um tempo semelhante entre o LB e o AQFC1, enquanto o AQFC2 e AQFC3 mantiveram os mesmos deslocamentos relativos (dos resultados apresentados) em relação ao AQFC1.

Os resultados referentes ao desempenho computacional dos métodos são apresentados na Figura 4.2.

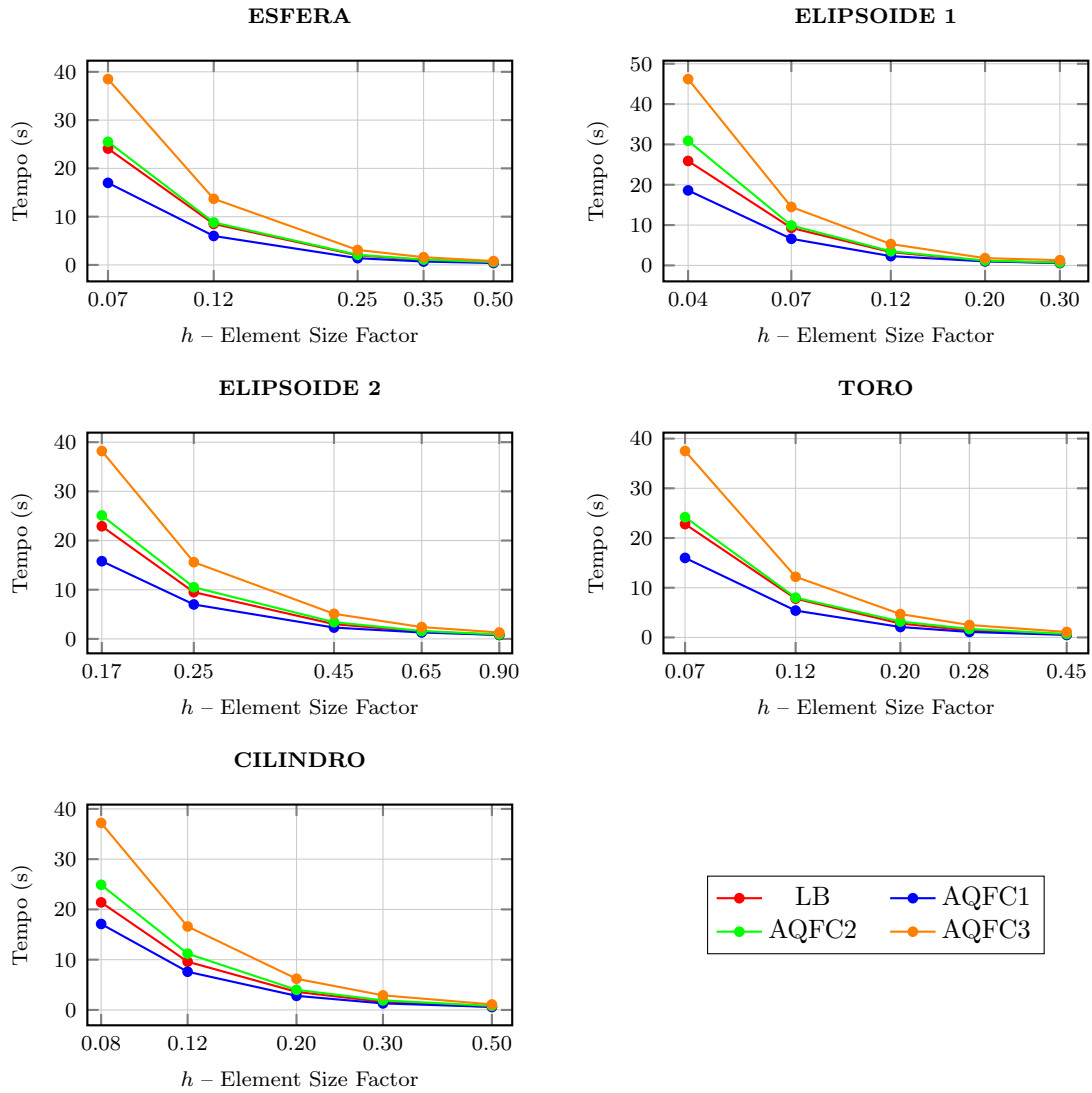


Figura 4.2: Tempo de execução em função de h para cada geometria analisada (legenda comum aos métodos).

Para todas as geometrias analisadas, observa-se **crescimento do tempo computacional à medida que a malha se torna mais refinada**, isto é, conforme o fator de tamanho de elemento h diminui. Esse comportamento decorre do aumento no número de vértices e elementos envolvidos no cálculo, que, em superfícies bidimensionais discretizadas, cresce aproximadamente de forma inversamente proporcional a h^2 .

Os perfis temporais dos métodos apresentam comportamento qualitativamente semelhante, diferindo principalmente por fatores multiplicativos associados à com-

plexidade local de cada estratégia. O tempo do método LB exibe crescimento regular e previsível. Embora os AQFCs acompanhem tendência similar, observa-se um **aumento progressivo do custo à medida que o tamanho da vizinhança considerada no ajuste quadrático é ampliado**, devido ao maior número de vértices envolvidos na montagem e resolução do sistema linear associado ao ajuste local.

Dessa forma, a avaliação definitiva da eficiência relativa entre os métodos não pode basear-se exclusivamente no tempo de execução. A análise conjunta com os indicadores de precisão, apresentados nas seções seguintes, é fundamental para contextualizar o desempenho computacional dentro do panorama completo da comparação.

4.3 Acurácia Global: Erro Relativo na Norma L_2

A Figura 4.3 apresenta o erro relativo na norma L_2 em função do refino da malha, permitindo comparar o desempenho global dos estimadores entre geometrias e níveis de vizinhança do AQFC.

Na esfera, os métodos AQFC2 e AQFC3 apresentam os menores erros para praticamente todos os níveis de h , enquanto o LB mantém valores variados. Outra observação, é que o erro do AQFC2 e AQFC3 não reduziu com o refino da malha, sugerindo um possível limite de precisão numérica do método.

No elipsoide 1, o desempenho do LB deteriora-se significativamente, com erros globais elevados e comportamento oscilatório ao longo de h . Em contraste, os métodos AQFC mantêm tendência de redução do erro com o refinamento, sendo o AQFC2 o que apresenta melhor equilíbrio entre estabilidade e precisão.

Quando se utiliza o elipsoide 2, que teve a malha regularizada, o método LB torna-se mais competitivo, evidenciando sua forte dependência da qualidade da triangulação. Ainda assim, AQFC2 e AQFC3 mantêm desempenho superior nas malhas mais refinadas.

No toro, o LB apresenta erro aproximadamente constante ao longo do refinamento, enquanto o AQFC demonstra melhora progressiva com a redução de h , especialmente para vizinhanças maiores. Já no cilindro, mesmo após filtragem das

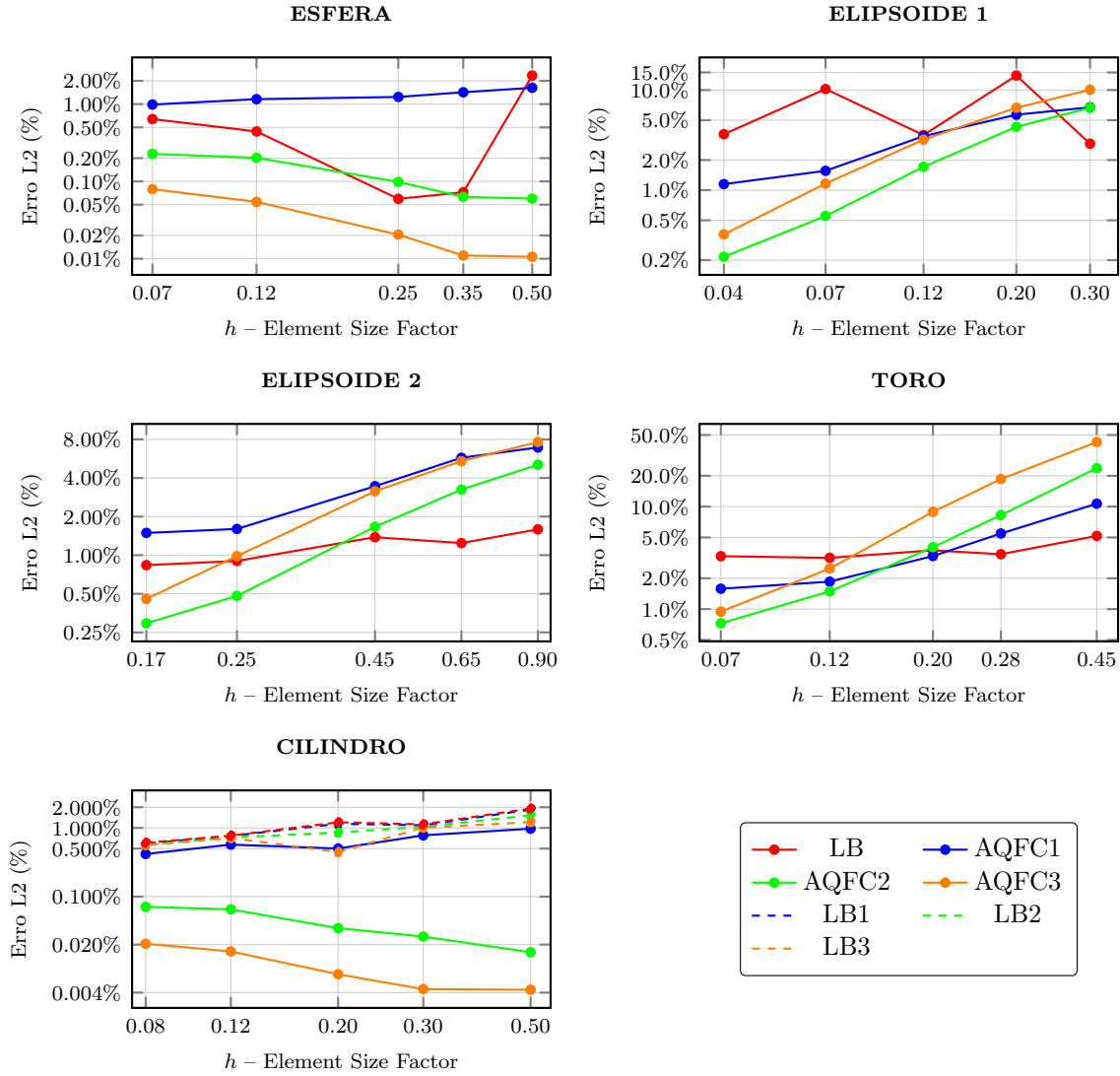


Figura 4.3: Erro relativo da norma L_2 em função de h para cada geometria analisada (legenda comum aos métodos).

regiões críticas, o AQFC2 e o AQFC3 apresentam erros significativamente inferiores aos do LB.

De maneira geral, os resultados indicam que os métodos AQFC2 e AQFC3 fornecem maior robustez e precisão global, particularmente em malhas com irregularidades, enquanto o LB mostra-se mais sensível à qualidade geométrica da discretização.

4.4 Acurácia Local: Erro Relativo Máximo

A Figura 4.4 apresenta o erro relativo máximo, que caracteriza o pior caso local e é especialmente informativo para avaliar sensibilidade a irregularidades, distorções e regiões geometricamente críticas. Diferentemente da norma L_2 , essa métrica é dominada por picos pontuais e tende a evidenciar instabilidades localizadas.

Em todas as geometrias, o método LB apresentou os erros mais elevados e comportamento menos regular com o refinamento, particularmente no elipsoide 1, onde os valores máximos atingem ordens de grandeza muito superiores aos observados para o AQFC. Já no elipsoide 2, observa-se redução significativa dos picos do LB em comparação ao caso irregular, reforçando a relação direta entre qualidade da malha e estabilidade local do método.

Os métodos AQFC mostram redução progressiva dos picos à medida que a vizinhança é ampliada. Esse comportamento está associado ao efeito de regularização implícita do ajuste por mínimos quadrados, que redistribui o erro entre os elementos vizinhos e reduz concentrações pontuais.

Novamente, a análise do erro máximo, sugere que o LB é mais suscetível a instabilidades localizadas, enquanto o AQFC, especialmente com vizinhanças ampliadas, apresenta maior robustez frente a irregularidades geométricas.

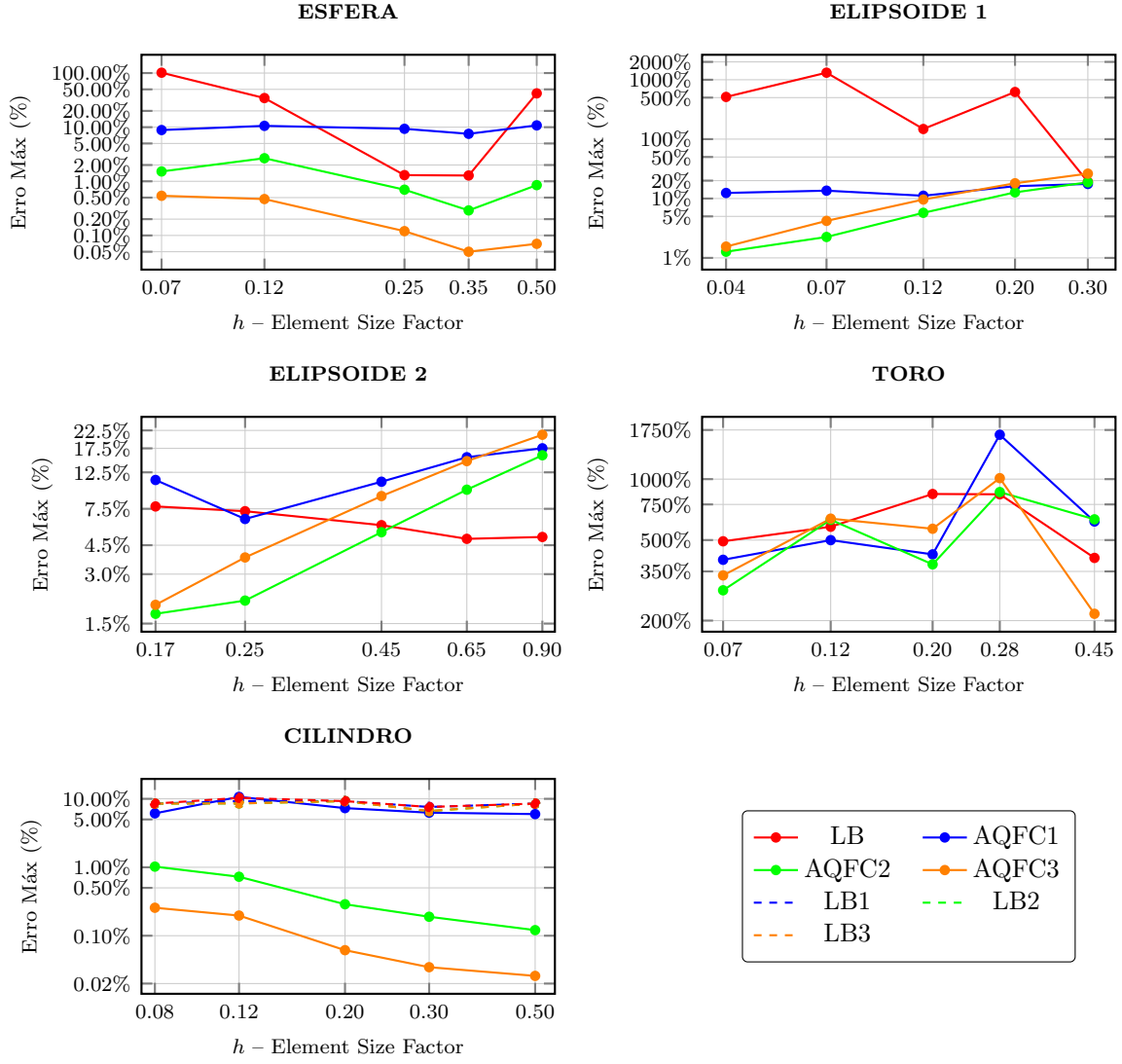


Figura 4.4: Erro relativo máximo em função de h para cada geometria analisada (legenda comum aos métodos).

4.5 Erro Relativo Médio

A Figura 4.5 apresenta o erro relativo médio, essa métrica complementa a análise ao descrever o comportamento local de forma direta. Novamente, observa-se que o aumento de vizinhança no AQFC reduz a variabilidade do erro em geometrias mais irregulares, enquanto o LB apresenta melhor desempenho no erro médio da esfera e desempenho competitivo nas outras geometrias.

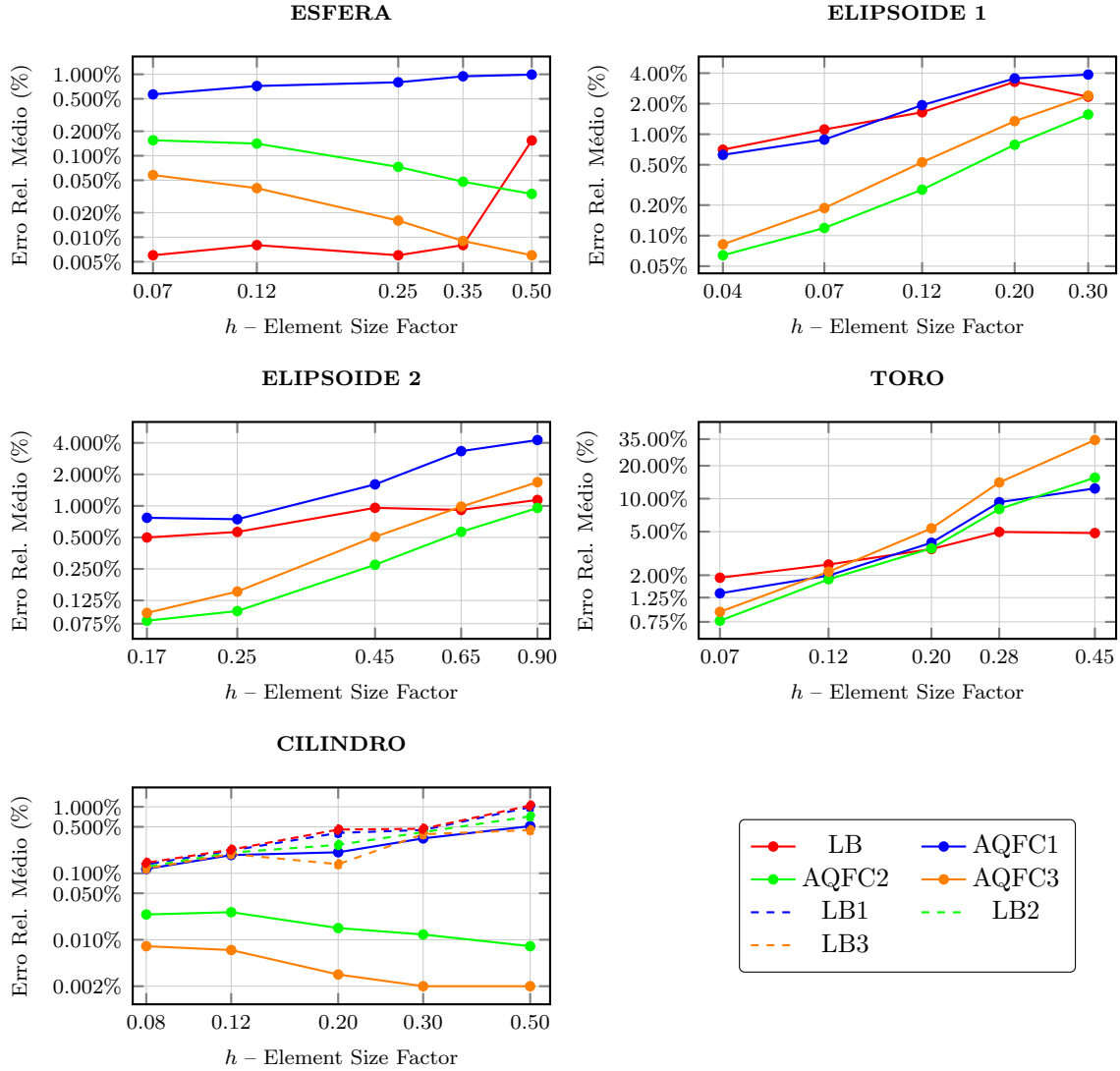


Figura 4.5: Erro relativo médio em função de h para cada geometria analisada (legenda comum aos métodos).

4.6 Exemplos Visuais e Padrões Espaciais do Erro

Para complementar a análise quantitativa apresentada anteriormente, são exibidas visualizações no ParaView dos campos de curvatura e dos mapas de erro em malhas representativas. Essas figuras permitem identificar padrões espaciais, localizar regiões críticas e relacionar qualitativamente o comportamento do erro às características geométricas da superfície e à qualidade da triangulação.

Ressalta-se que as escalas de cores utilizadas nas visualizações não são idênticas entre todas as comparações. Essa escolha foi adotada para destacar de forma mais clara as regiões de interesse em cada caso, pois uma escala global comum poderia reduzir o contraste e dificultar a visualização dos padrões espaciais do erro. Dessa forma, a análise das figuras deve considerar também os valores apresentados nas barras de escala, sendo as visualizações utilizadas como complemento qualitativo às métricas quantitativas discutidas anteriormente.

4.6.1 Curvaturas

Nas figuras 4.6, 4.7, 4.8, 4.9 e 4.10 são apresentadas, para cada geometria analisada, os mapas de curvatura média em malhas com refinamento intermediário. Em cada caso, são exibidos lado a lado: (i) a curvatura média analítica, (ii) a curvatura média estimada pelo operador Laplace–Beltrami e (iii) a curvatura média estimada pelo método AQFC com dois anéis de vizinhança.

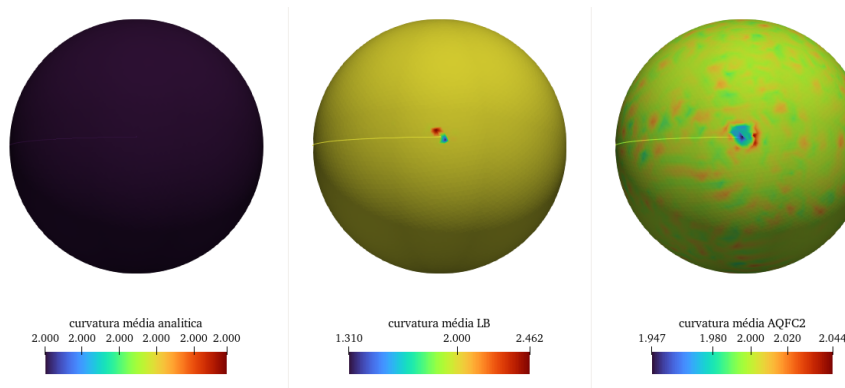


Figura 4.6: Curvatura da esfera na malha com $h = 0.12$, comparando lado a lado: (i) a curvatura média analítica, (ii) a curvatura média estimada pelo operador Laplace–Beltrami e (iii) a curvatura média estimada pelo método AQFC com dois anéis de vizinhança.

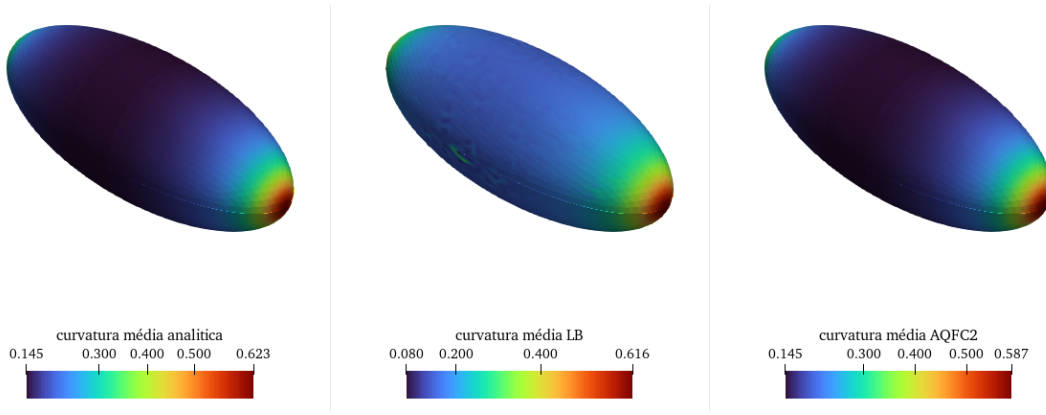


Figura 4.7: Curvatura do elipsoide 1 na malha com $h = 0.12$, comparando lado a lado: (i) a curvatura média analítica, (ii) a curvatura média estimada pelo operador Laplace–Beltrami e (iii) a curvatura média estimada pelo método AQFC com dois anéis de vizinhança.

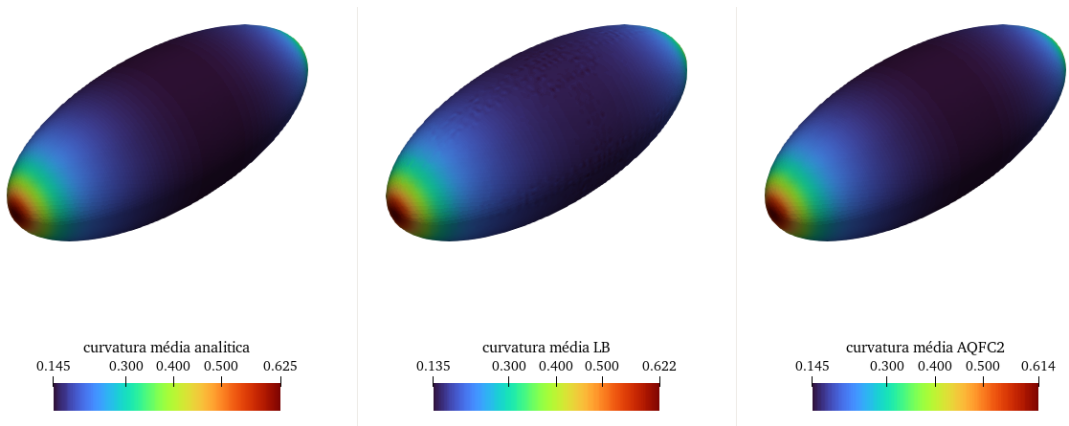


Figura 4.8: Curvatura do elipsoide 2 na malha com $h = 0.25$, comparando lado a lado: (i) a curvatura média analítica, (ii) a curvatura média estimada pelo operador Laplace–Beltrami e (iii) a curvatura média estimada pelo método AQFC com dois anéis de vizinhança.

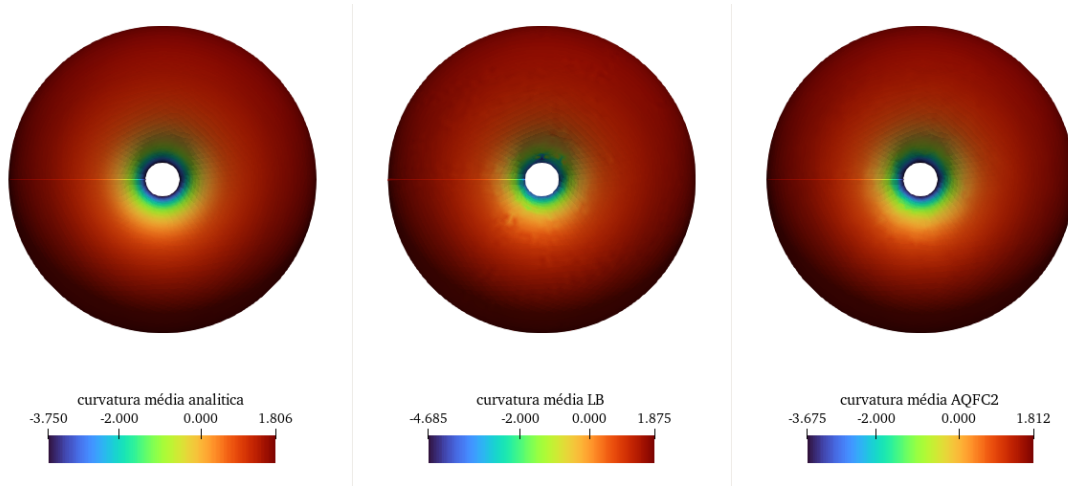


Figura 4.9: Curvatura do toro na malha com $h = 0.12$, comparando lado a lado: (i) a curvatura média analítica, (ii) a curvatura média estimada pelo operador Laplace–Beltrami e (iii) a curvatura média estimada pelo método AQFC com dois anéis de vizinhança.

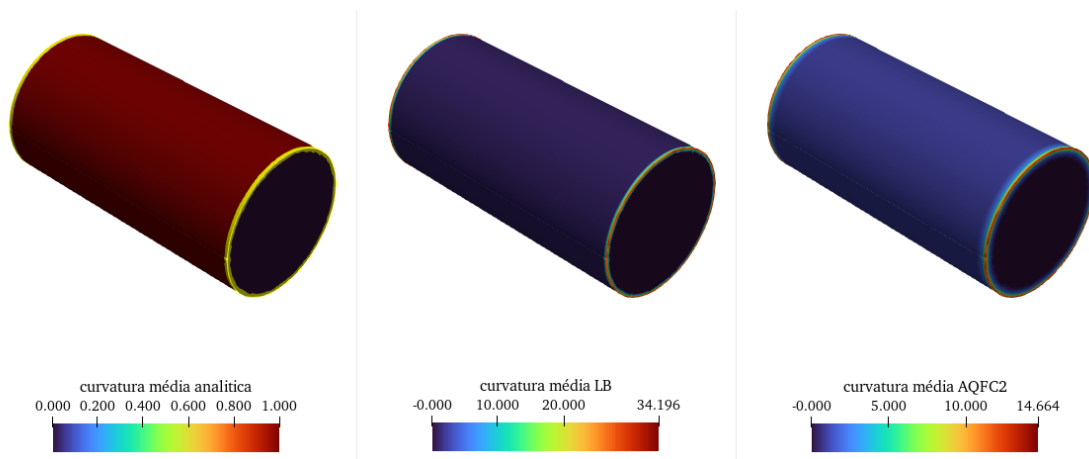


Figura 4.10: Curvatura do cilindro na malha com $h = 0.12$, comparando lado a lado: (i) a curvatura média analítica, (ii) a curvatura média estimada pelo operador Laplace–Beltrami e (iii) a curvatura média estimada pelo método AQFC com dois anéis de vizinhança.

4.6.2 Comparação de Malhas: Elipsoide 1 x Elipsoide 2

Devido às irregularidades observadas na malha do elipsoide 1, foi necessário reformular o arquivo `.geo` com o objetivo de gerar uma nova discretização com melhor qualidade geométrica dos elementos.

Para caracterizar essas diferenças, foram analisadas métricas clássicas de qualidade de malha: **razão de aspecto**, **razão r/R** e **distorção angular (skewness)**. Adicionalmente, considera-se neste trabalho uma métrica denominada **qualidade composta**, definida a partir da combinação ponderada das métricas anteriores.

Inicialmente, apresentam-se os mapas espaciais dessas métricas para uma malha do elipsoide 1 e outra gerada após a reformulação do arquivo `.geo`, o elipsoide 2.

Na malha do elipsoide 1 (Figura 4.11), observam-se regiões com valores elevados das métricas de qualidade distribuídas de forma irregular ao longo da superfície. Esse comportamento indica a presença de elementos com elevado alongamento ou distorção angular, caracterizando uma triangulação geometricamente menos uniforme.

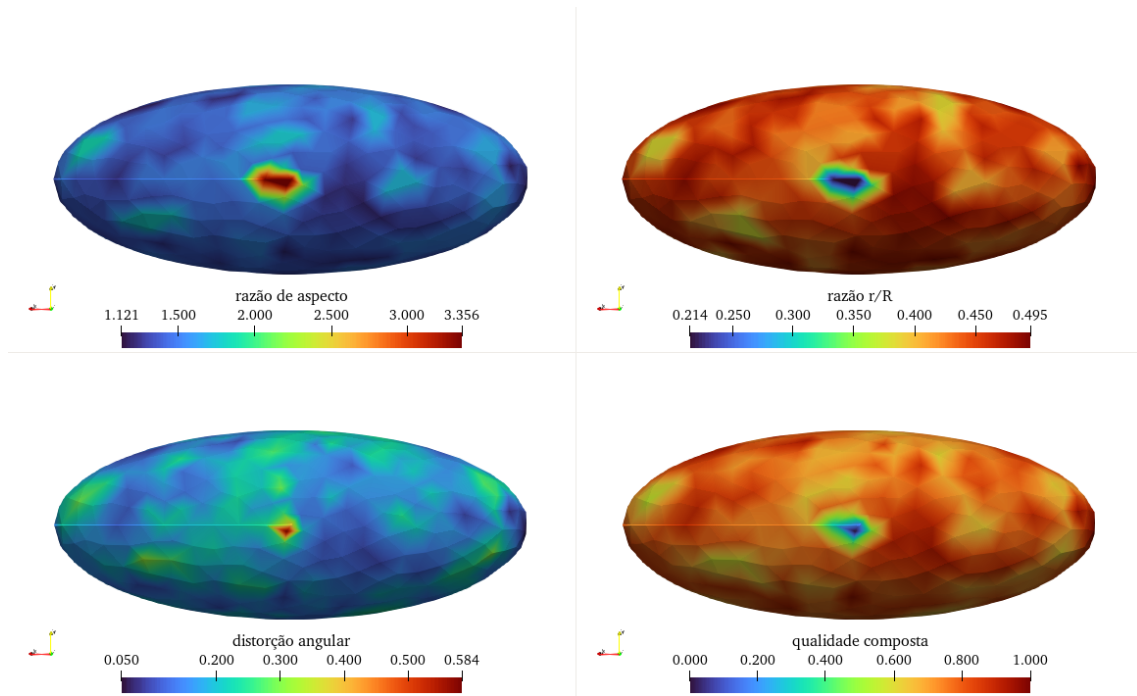


Figura 4.11: Malha do elipsoide 1 com $h = 0.30$. Métricas de qualidade da malha plotadas lado a lado: razão de aspecto (valor ideal = 1.0), razão r/R (valor ideal = 0.5), distorção angular (valor ideal = 0.0) e qualidade composta (valor ideal = 1.0), respectivamente.

Por sua vez, na malha do elipsoide 2 (Figura 4.12), as regiões de maior distorção apresentam um padrão de distribuição espacial organizado, concentrando-se principalmente nas interfaces entre subdivisões da malha, como na Figura 3.2.

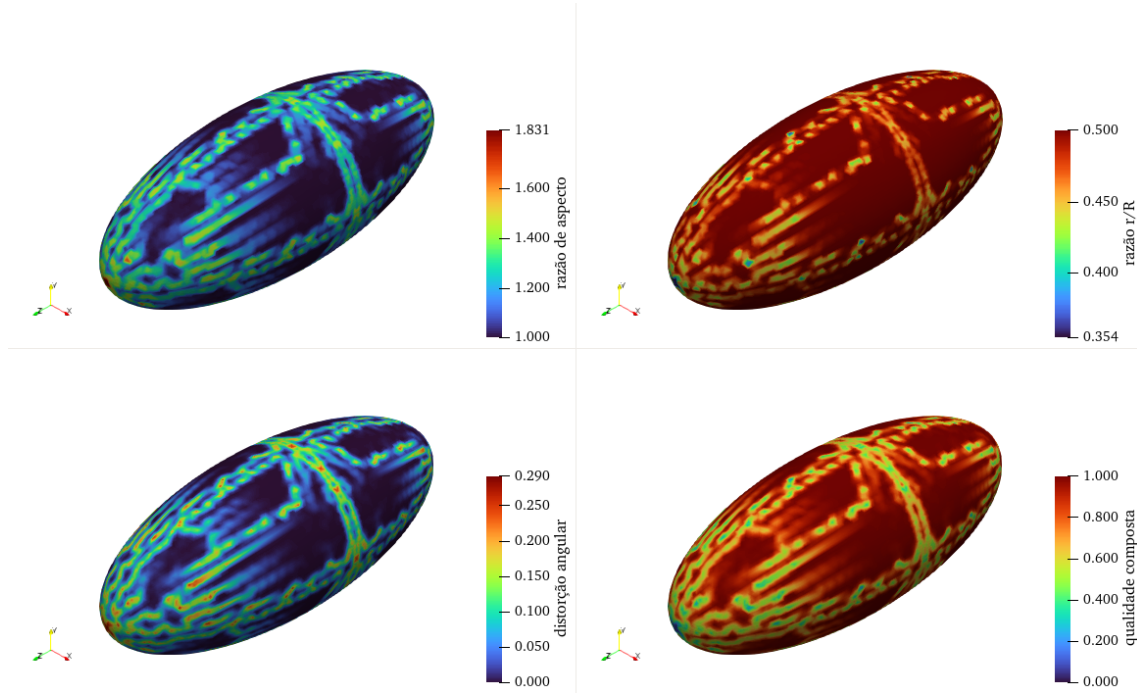


Figura 4.12: Malha do elipsoide 2 com $h = 0.17$. Métricas de qualidade da malha plotadas lado a lado: razão de aspecto (valor ideal = 1.0), razão r/R (valor ideal = 0.5), distorção angular (valor ideal = 0.0) e qualidade composta (valor ideal = 1.0), respectivamente.

Embora os mapas de qualidade permitam identificar essas diferenças visualmente, uma avaliação quantitativa é necessária para caracterizar de forma objetiva o nível de irregularidade geométrica presente em cada malha. Para esse fim, foram calculadas estatísticas globais das métricas de qualidade dos exemplos visuais selecionados, conforme apresentado nas Tabelas 4.1 e 4.2.

MÉTRICA	MÉDIA	DP	MÍN	P5	P50	P95	MÁX
Razão de Aspecto	1,45686	0,19864	1,12113	1,24832	1,42675	1,70223	3,35620
Distorção Angular	0,17616	0,05004	0,05041	0,10207	0,17071	0,25216	0,58356
Razão r/R	0,44169	0,03260	0,21443	0,39693	0,44664	0,48133	0,49483
Qualidade Composta	0,75861	0,09706	0,00000	0,61572	0,76662	0,90048	1,00000

Tabela 4.1: Estatísticas globais das métricas de qualidade para a malha do elipsoide 1 com $h = 0.30$. Reportam-se média, desvio-padrão (DP), mínimo, percentis (P5, P50/mediana, P95) e máximo.

MÉTRICA	MÉDIA	DP	MÍN	P5	P50	P95	MÁX
Razão de Aspecto	1,13316	0,14261	1,00021	1,00130	1,07555	1,40787	1,83084
Distorção Angular	0,05258	0,05888	0,00005	0,00053	0,02880	0,17488	0,29030
Razão r/R	0,48634	0,02140	0,35412	0,43964	0,49776	0,50000	0,50000
Qualidade Composta	0,83092	0,18490	0,00000	0,46355	0,91015	0,99851	1,00000

Tabela 4.2: Estatísticas globais das métricas de qualidade para a malha do elipsoide 2 com $h = 0.17$. Reportam-se média, desvio-padrão (DP), mínimo, percentis (P5, P50/mediana, P95) e máximo.

No caso da malha do elipsoide 1, as estatísticas refletem uma condição de maior heterogeneidade geométrica. A razão de aspecto apresenta média de aproximadamente 1,46 e valor máximo de 3,36, indicando a ocorrência de elementos relativamente alongados. De modo consistente, a distorção angular apresenta média de 0,176 e valor máximo de 0,584, sugerindo a presença de regiões com pior regularidade angular. A razão r/R reforça essa interpretação, com valor mínimo de aproximadamente 0,21, compatível com a ocorrência de triângulos de baixa qualidade geométrica. Em conjunto, esses indicadores são coerentes com o padrão visual observado no mapa, no qual as distorções aparecem distribuídas de forma irregular ao longo da superfície.

Para a malha do elipsoide 2, as estatísticas indicam uma distribuição geométrica mais regular na maior parte da superfície. A razão de aspecto apresenta média

de aproximadamente 1,13 e máximo de 1,83, enquanto a distorção angular assume média de 0,053, com valores reduzidos na maior parte dos elementos. A razão r/R apresenta mediana próxima de 0,50, sugerindo predominância de triângulos com boa regularidade geométrica. De forma consistente, a qualidade composta apresenta mediana elevada e percentil superior muito próximo de 1, indicando que grande parte da superfície é formada por elementos com boa regularidade. Esses valores indicam que, de modo geral, os elementos da malha apresentam melhor regularidade geométrica, resultado consistente com o mapa correspondente.

Sob essa perspectiva, o ponto central desta análise é evidenciar que os padrões geométricos observados visualmente podem ser descritos de forma consistente por meio das métricas de qualidade adotadas. Essa caracterização é relevante para a discussão subsequente dos mapas de erro, uma vez que permite examinar, ainda que qualitativamente, como regiões de maior irregularidade geométrica podem estar associadas a maiores erros nas estimativas de curvatura.

4.6.3 Relação entre a Qualidade da Malha e os Erros

A análise visual das Figuras 4.13–4.19 evidencia correlação espacial entre as métricas de qualidade da malha e os padrões de erro na estimativa da curvatura média.

Na Figura 4.13, observa-se que a região com maior distorção da razão r/R coincide com a região de maior erro absoluto no método LB. Essa correspondência indica que o operador discreto é particularmente sensível a distorções geométricas locais, refletindo diretamente a qualidade da malha na estimativa da curvatura.

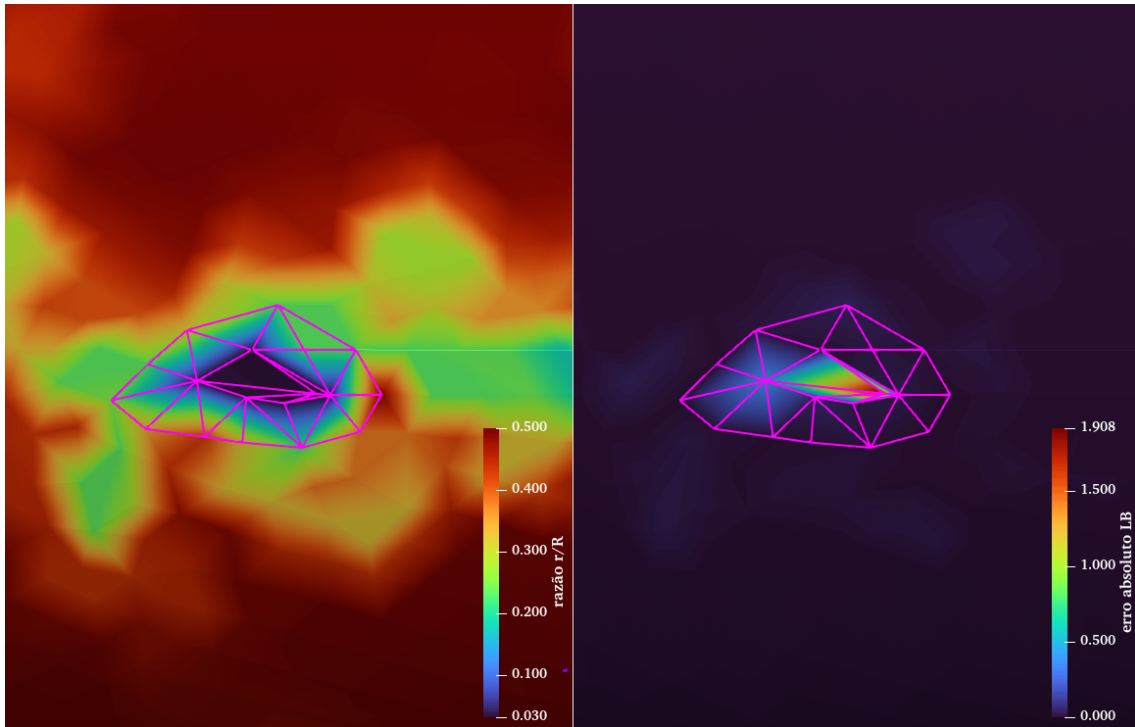


Figura 4.13: Malha do elipsoide 1 com $h = 0.07$: comparação da mesma região ampliada, apresentando a razão r/R (valor ideal = 0.5) e o erro absoluto da curvatura média calculada pelo método LB.

Essa relação torna-se ainda mais clara ao comparar, lado a lado, os mapas de qualidade e os mapas de erro para os diferentes métodos (Figuras 4.14 e 4.15). O erro do LB concentra-se predominantemente nas mesmas regiões onde as métricas de qualidade indicam maior distorção, reproduzindo padrão espacial semelhante ao das irregularidades da malha.

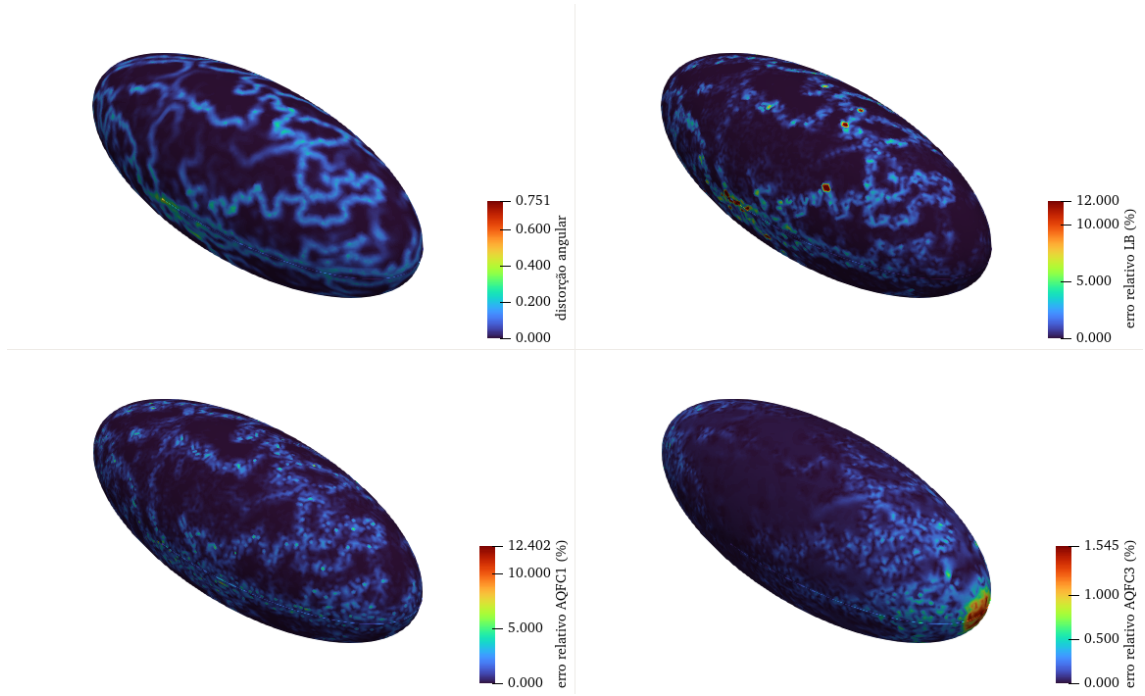


Figura 4.14: Malha do elipsoide 1 com $h = 0.04$: comparação entre os mapas de erro relativo percentual da curvatura média (LB, AQFC1 e AQFC3) e a distorção angular (valor ideal = 0.0).

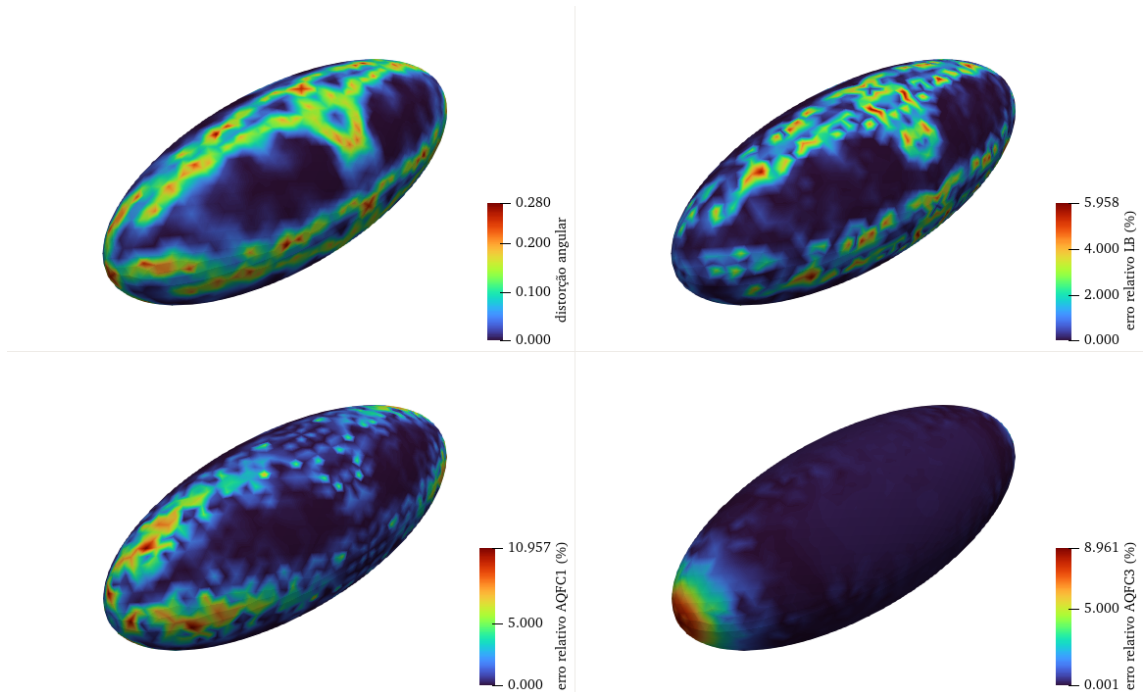


Figura 4.15: Malha do elipsoide 2 com $h = 0.45$: comparação entre os mapas de erro relativo percentual da curvatura média (LB, AQFC1 e AQFC3) e a distorção angular (valor ideal = 0.0).

Para o AQFC com vizinhança reduzida, observa-se comportamento semelhante ao do LB, porém com menor intensidade dos picos. À medida que a vizinhança é ampliada, os erros tornam-se espacialmente mais distribuídos, reduzindo a concentração pontual e promovendo maior estabilidade local da estimativa.

Esse efeito é evidenciado nas Figuras 4.16 e 4.17, nas quais o aumento de vizinhança reduz significativamente os valores máximos locais, embora possa introduzir uma suavização excessiva em regiões de variação de curvatura acentuada, como pode ser observado nas extremidades do elipsoide.

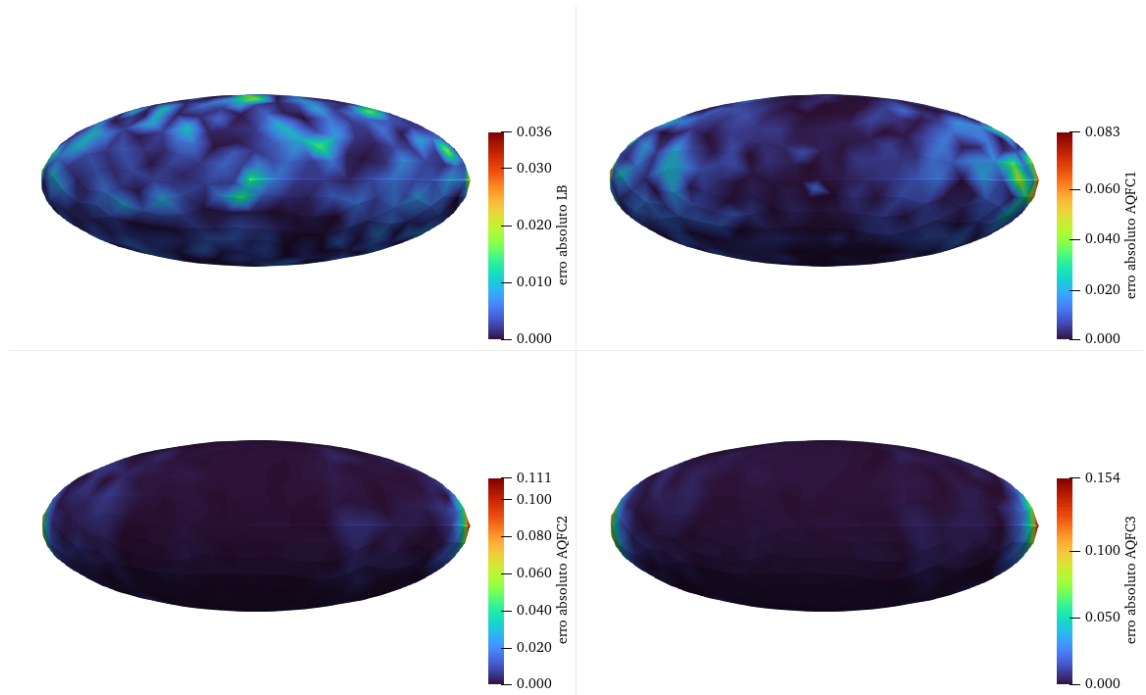


Figura 4.16: Malha do elipsoide 1 com $h = 0.30$: comparação entre os mapas de erro absoluto da curvatura média (LB e AQFC com 1, 2 e 3 vizinhanças).

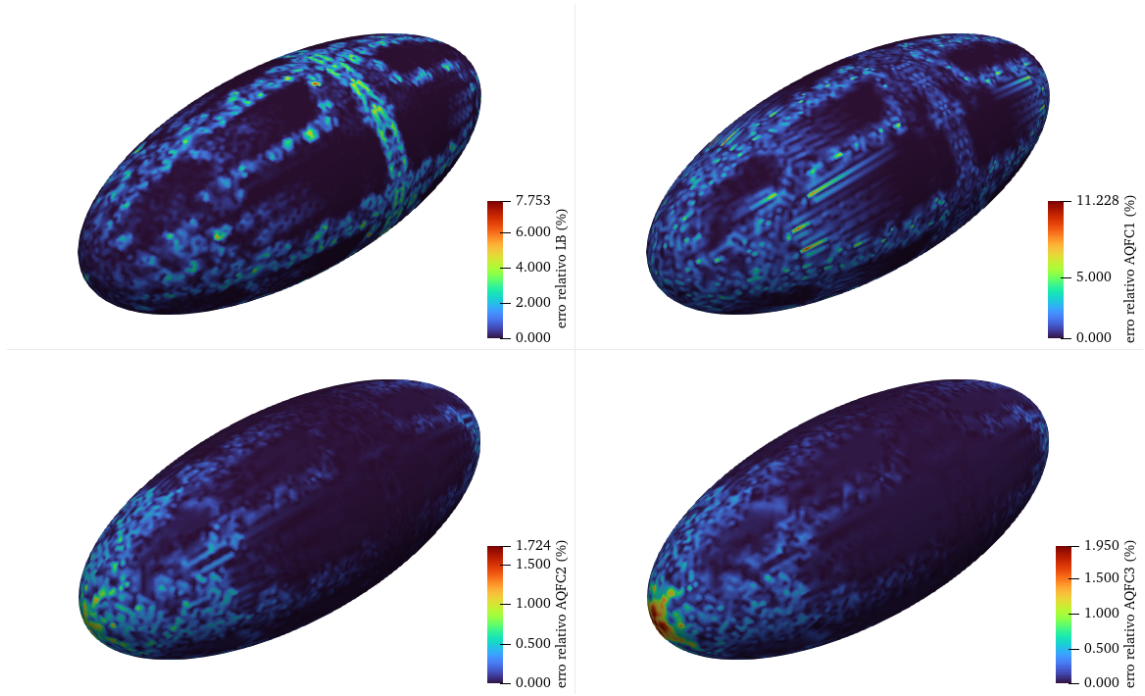


Figura 4.17: Malha do elipsoide 2 com $h = 0.17$: comparação entre os mapas de erro relativo percentual da curvatura média (LB e AQFC com 1, 2 e 3 vizinhanças).

A Figura 4.18 ilustra de forma clara o mecanismo de redistribuição do erro com o aumento da vizinhança. No método LB, o erro concentra-se em um vértice específico. No AQFC1, essa concentração desloca-se ligeiramente; no AQFC2 e AQFC3, o erro torna-se progressivamente mais difuso, distribuindo-se entre os elementos vizinhos. Além disso, pela análise da escala de cores, observa-se uma redução global na magnitude dos erros à medida que a vizinhança é ampliada, evidenciando o efeito suavizador do aumento do domínio de ajuste.

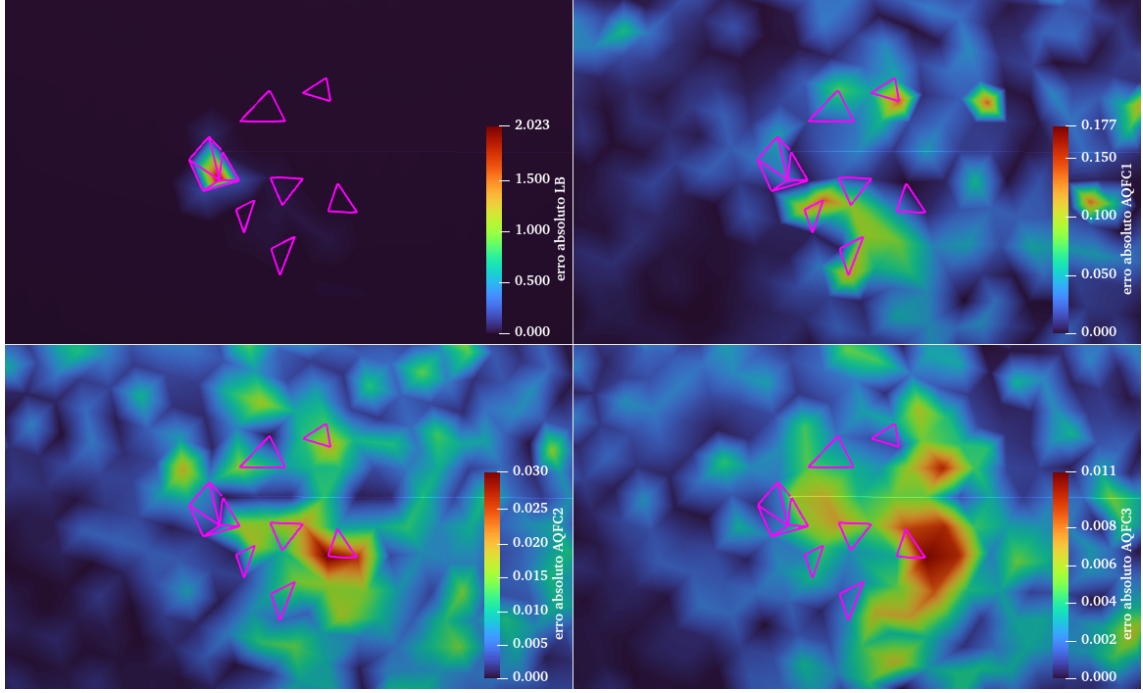


Figura 4.18: Malha da esfera com $h = 0.07$: comparação ampliada entre os mapas de erro absoluto da curvatura média para os métodos LB e AQFC (1, 2 e 3 vizinhanças). A escala de cores (barra lateral) representa a magnitude do erro absoluto, evidenciando a diminuição dos valores à medida que a vizinhança é ampliada.

Esse comportamento pode ser interpretado como uma redistribuição espacial do erro associada ao aumento do domínio de ajuste no método AQFC. Ao incorporar maior número de vértices no problema de mínimos quadrados, o ajuste torna-se menos sensível a perturbações locais isoladas, resultando em maior estabilidade.

Fenômeno semelhante pode ser observado no cilindro (Figura 4.19), onde o AQFC2 apresenta erros significativamente menores e mais distribuídos do que o LB.

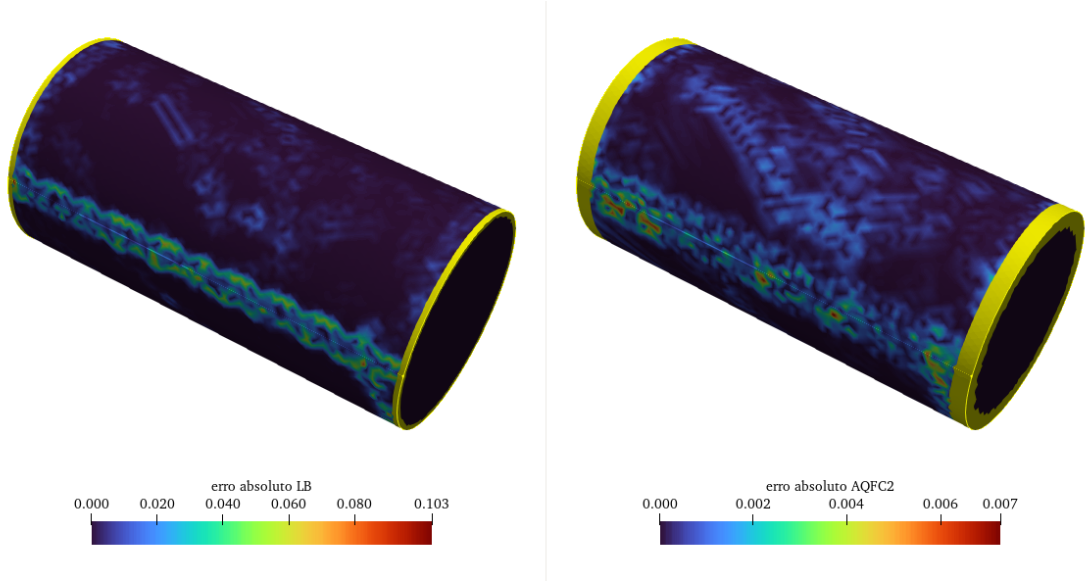


Figura 4.19: Malha do cilindro com $h = 0.12$: comparação entre os mapas de erro absoluto da curvatura média para os métodos LB e AQFC com 2 vizinhanças.

De forma consolidada, as evidências visuais confirmam que a qualidade da malha influencia diretamente o desempenho dos estimadores.

Nesse contexto, torna-se necessário estabelecer um compromisso entre o grau de refinamento da malha (h) e o tamanho da vizinhança adotada no ajuste quadrático, considerando o tipo de geometria analisada, a variação espacial da curvatura e a relação entre custo computacional e precisão desejada. Em geometrias mais uniformes, com variações suaves de curvatura, a ampliação da vizinhança pode ser empregada sem necessidade de refinamento excessivo. Por outro lado, em regiões com variações abruptas de curvatura, o refino da malha torna-se essencial, uma vez que a simples ampliação da vizinhança não é suficiente para capturar adequadamente o comportamento geométrico local.

Sob essa perspectiva, o método LB apresenta forte correlação entre distorções geométricas e concentração de erro, enquanto o AQFC, especialmente com vizinhanças ampliadas, demonstra maior robustez frente a irregularidades locais, ao custo de uma suavização progressiva do campo estimado.

4.7 Estudos de Sensibilidade

Com base nos resultados quantitativos e qualitativos apresentados nas seções anteriores, esta seção reúne e interpreta, de forma integrada, os principais fatores que influenciam o desempenho dos métodos de estimativa de curvatura analisados. Em vez de discutir isoladamente cada gráfico ou mapa, busca-se aqui sintetizar os efeitos dos parâmetros mais relevantes do problema: o tamanho da vizinhança adotada no método AQFC, a qualidade geométrica e o nível de refinamento da malha.

Essa abordagem é particularmente importante porque o erro numérico observado não depende de um único fator de forma isolada. Ao contrário, ele resulta da interação entre a representação geométrica da superfície, a regularidade dos elementos, a escala espacial capturada pela vizinhança de ajuste e a própria natureza do operador numérico empregado. Assim, a análise de sensibilidade permite compreender não apenas qual método apresenta menor erro em determinado caso, mas também sob quais condições esse desempenho tende a se manter, deteriorar ou se tornar mais robusto.

4.7.1 Sensibilidade ao Tamanho da Vizinhança no AQFC

O tamanho da vizinhança adotada no ajuste quadrático constitui um dos parâmetros mais importantes do método AQFC, pois controla diretamente a extensão espacial da informação geométrica utilizada na estimativa da curvatura. Em termos práticos, o aumento do número de anéis de vizinhança amplia o conjunto de pontos empregados no problema de mínimos quadrados, tornando o ajuste menos sensível a perturbações locais isoladas. Por outro lado, essa ampliação também pode reduzir a fidelidade do método à geometria estritamente local da superfície.

Os resultados obtidos mostram que a ampliação da vizinhança tende, em geral, a reduzir a ocorrência de picos locais de erro. Esse comportamento é evidenciado pela análise do gráfico de erro máximo (Figura 4.4), no qual, para uma mesma malha, os métodos AQFC2 e AQFC3 apresentam erros inferiores aos observados para o AQFC1. Evidências adicionais podem ser observadas nos mapas de erro, como na Figura 4.18, onde se nota que o AQFC com vizinhanças ampliadas produz campos de

erro mais distribuídos espacialmente e menos concentrados em vértices específicos. Esse efeito pode ser interpretado como uma forma de regularização implícita do ajuste, uma vez que a contribuição de múltiplos pontos vizinhos reduz a influência de flutuações puramente locais.

Entretanto, essa tendência não implica que vizinhanças maiores sejam sempre preferíveis. Em diversas situações, observou-se que o aumento da vizinhança pode introduzir um efeito de suavização que reduz a capacidade do método de representar com fidelidade variações geométricas mais acentuadas da superfície. Esse comportamento pode ser observado, por exemplo, no caso do cilindro, em que, devido às singularidades nas bordas, foi necessário ignorar não apenas os vértices de fronteira, mas também aqueles cujas vizinhanças incluíam tais pontos. Situação semelhante ocorre no toro, onde os erros globais das vizinhanças ampliadas passam a ser inferiores aos do AQFC1 apenas a partir de níveis intermediários de refinamento da malha.

Nos elipsoides, esse efeito também pode ser identificado nos mapas de erro das Figuras 4.16 e 4.17, nos quais se observa uma tendência de concentração dos maiores erros nas extremidades da geometria, regiões associadas a maiores variações de curvatura, à medida que a vizinhança é ampliada. Além disso, ao analisar os gráficos do erro L_2 (Figura 4.3), nota-se que, para as geometrias de curvatura variável, o AQFC3 apresenta tendência de superar o AQFC2 à medida que a malha se torna mais refinada, indicando que vizinhanças maiores precisam de discretizações mais finas, a depender da variação de curvatura no local. Em outras palavras, o ajuste passa a representar uma média geométrica sobre uma região da superfície, sendo necessário um nível de refinamento compatível com a vizinhança considerada para que os pontos representem adequadamente a geometria local.

Do ponto de vista quantitativo, os resultados indicam que o AQFC1 tende a ser mais sensível a perturbações locais e, conseqüentemente, mais suscetível a oscilações pontuais associadas à qualidade da malha. Por outro lado, o AQFC3, embora frequentemente apresente os menores picos de erro e maior regularidade espacial, também está associado a maior custo computacional e, em alguns casos, à suavização excessiva da estimativa. Nesse contexto, o AQFC2 mostrou, de modo geral, o melhor compromisso entre estabilidade numérica, fidelidade geométrica e

custo computacional, destacando-se como a configuração mais equilibrada entre as avaliadas.

Dessa forma, conclui-se que a sensibilidade do AQFC ao tamanho da vizinhança reflete um compromisso entre precisão geométrica e custo computacional. Em regiões com menor variação de curvatura, uma malha menos refinada associada a vizinhanças maiores pode ser suficiente para obter estimativas estáveis e precisas, com custo computacional relativamente reduzido. Por outro lado, em regiões com maior variação de curvatura, a utilização de vizinhanças ampliadas pode exigir níveis mais elevados de refinamento da malha para preservar a fidelidade geométrica da estimativa, o que implica aumento significativo do custo computacional. Nessas situações, a adoção de vizinhanças menores pode representar uma alternativa mais eficiente, caso o nível de precisão requerido permita essa simplificação.

4.7.2 Sensibilidade à Irregularidade da Malha

A qualidade geométrica da malha mostrou-se um fator decisivo na precisão dos estimadores de curvatura, sobretudo para o método baseado no operador Laplace–Beltrami. As comparações entre o elipsoide 1 e o elipsoide 2 evidenciaram de forma clara que a regularidade da triangulação exerce influência direta tanto sobre os níveis globais de erro quanto sobre a distribuição espacial das regiões críticas.

No caso do elipsoide 1, os mapas de qualidade revelaram a presença de elementos mais alongados e regiões com maior distorção angular distribuídas de maneira irregular ao longo da superfície. As estatísticas globais dessas métricas confirmaram essa heterogeneidade geométrica. Em correspondência, os mapas de erro mostraram que os maiores erros do método LB se concentram precisamente em regiões onde a malha apresenta pior qualidade (Figura 4.13), indicando forte correlação espacial entre distorção geométrica e degradação da estimativa numérica. Essa sensibilidade também se refletiu nas métricas globais, em especial no erro máximo, que assumiu valores muito superiores aos observados para os métodos AQFC.

A reformulação da malha no elipsoide 2 permitiu reduzir significativamente essas distorções, produzindo uma discretização mais uniforme. Como consequência, os erros do método LB tornaram-se substancialmente menores e mais regulares, o que reforça a interpretação de que esse método depende fortemente da qualidade local

dos elementos. Em outras palavras, o operador discreto Laplace–Beltrami mostrou elevada sensibilidade a irregularidades geométricas da malha, sendo particularmente afetado por triangulações não uniformes ou com distorções locais intensas.

Os métodos AQFC, por sua vez, demonstraram maior robustez frente a esse tipo de irregularidade. Embora também sejam influenciados pela qualidade geométrica da malha, o uso de um ajuste local por mínimos quadrados faz com que a informação de múltiplos vértices contribua para a estimativa, atenuando o impacto de perturbações associadas a elementos isolados de baixa qualidade. Esse comportamento foi observado tanto nos gráficos de erro quanto nos mapas espaciais, nos quais os campos estimados por AQFC tendem a apresentar menor concentração de erro nas regiões mais distorcidas da malha em relação ao LB.

Assim, os resultados mostram que a qualidade da triangulação não é apenas um aspecto auxiliar da discretização, mas um fator estrutural para o desempenho dos estimadores. Em particular, o método LB é dependente da regularidade da malha, ao passo que o AQFC apresenta maior tolerância a distorções locais, especialmente quando associado a vizinhanças ampliadas. Essa constatação é relevante do ponto de vista prático, pois indica que, em aplicações nas quais não se dispõe de malhas altamente regulares, métodos de ajuste local podem oferecer maior robustez numérica.

4.7.3 Influência do Refinamento da Malha

O refinamento da malha, representado neste trabalho pela redução do fator de tamanho de elemento h , influencia simultaneamente a representação geométrica da superfície, o número de vértices envolvidos no cálculo da curvatura e a escala espacial sobre a qual os estimadores operam. Em princípio, espera-se que o refino melhore a aproximação da geometria contínua e, conseqüentemente, reduza o erro numérico. No entanto, os resultados obtidos mostram que essa relação não é uniforme entre todos os métodos e geometrias analisados.

Do ponto de vista computacional, observou-se, para todas as geometrias, aumento monotônico do tempo de execução à medida que a malha se torna mais refinada. Esse comportamento está associado ao crescimento do número de vértices e elementos da discretização, o que amplia o custo total das operações locais de

estimativa de curvatura. Embora todos os métodos acompanhem essa tendência, o AQFC apresenta acréscimo progressivo de custo com o aumento da vizinhança, pois a montagem e a resolução do problema de ajuste local passam a envolver um número maior de pontos.

Sob o ponto de vista da precisão, o refino produziu comportamentos distintos conforme o método e a geometria considerados. Em alguns casos, como no AQFC aplicado ao elipsoide e ao toro, observou-se redução consistente das métricas de erro com a diminuição de h , indicando ganho efetivo de precisão. Em outros casos, entretanto, essa tendência não se manifestou de forma monotônica. Na esfera e no cilindro, por exemplo, os métodos com vizinhança ampliada já apresentavam erros muito baixos nas malhas mais grosseiras, e o refino adicional não resultou em melhora proporcional, sugerindo a presença de um limite prático de precisão numérica. Já no método LB, houve situações em que o erro global permaneceu aproximadamente constante com o refino ou apresentou comportamento irregular, especialmente em malhas de pior qualidade. Comportamento semelhante foi observado no erro máximo, enquanto o erro médio diminuiu com o refino.

O fato do erro médio diminuir com o refino, enquanto os outros erros não melhoram muito ou variam, indica que a maioria dos desvios ponto à ponto realmente estão diminuindo. Porém, em regiões localizadas, estão amplificando os picos de erro, já que o erro máximo aumenta e o erro L_2 fica mais instável, pelo fato dele ser mais afetado por maiores desvios.

Esses resultados indicam que o refinamento da malha, por si só, não garante convergência uniforme ou melhora sistemática da estimativa. O efeito do refino depende da forma como cada método utiliza a informação geométrica disponível e de como essa informação interage com a qualidade dos elementos e com o tamanho da vizinhança adotada. Em particular, quando a vizinhança é demasiadamente ampla para a escala local representada pela malha refinada, o método pode passar a incorporar informação além do necessário para descrever a geometria local, produzindo suavização adicional e limitando os ganhos de precisão.

Dessa forma, o refinamento deve ser interpretado como um parâmetro cuja influência é condicionada pelos demais fatores do problema. Em malhas suficientemente regulares e com escolha adequada da vizinhança, a redução de h tende a

beneficiar a estimativa. Entretanto, quando há distorções geométricas relevantes ou incompatibilidade entre a escala da malha e o tamanho da vizinhança, o efeito positivo do refino pode ser parcialmente reduzido ou até mesmo amplificar os erros.

4.7.4 Interação entre Refinamento, Vizinhança e Qualidade da Malha

As análises apresentadas nas subseções anteriores indicam que o desempenho dos estimadores de curvatura não depende de um único parâmetro isolado. Em vez disso, o erro observado resulta da interação entre três fatores principais: o refinamento da malha, o tamanho da vizinhança adotada no ajuste quadrático e a qualidade geométrica da triangulação. A partir dos resultados obtidos, podem ser destacadas as seguintes tendências gerais:

- **Interação entre refinamento da malha e tamanho da vizinhança.** O refinamento da malha altera a escala espacial representada pelos elementos e, conseqüentemente, a escala efetiva da vizinhança utilizada no AQFC. Em malhas mais grosseiras, vizinhanças ampliadas tendem a melhorar a estabilidade do ajuste ao incorporar maior quantidade de informação geométrica. Já em malhas mais refinadas, vizinhanças menores tornam-se suficientes para representar adequadamente a geometria local, enquanto vizinhanças excessivamente amplas podem introduzir suavização desnecessária e aumentar o custo computacional.
- **Influência da qualidade da malha na resposta ao refinamento.** A qualidade geométrica da triangulação condiciona o benefício obtido com o refinamento. Em malhas irregulares, o aumento do número de elementos não garante melhoria sistemática da estimativa, particularmente no método baseado no operador Laplace–Beltrami, cuja discretização é mais sensível a distorções geométricas. Em contrapartida, quando a malha apresenta maior regularidade, o efeito do refinamento tende a se manifestar de forma mais consistente nas métricas de erro.
- **Interação entre qualidade da malha e tamanho da vizinhança.** Em triangulações irregulares, vizinhanças pequenas tornam o ajuste mais sensível

à conectividade local e às distorções dos elementos adjacentes. A ampliação da vizinhança tende a reduzir essa sensibilidade, atuando como mecanismo de estabilização do ajuste. Entretanto, vizinhanças excessivamente amplas podem comprometer a representação de variações locais de curvatura, sobretudo em geometrias com mudanças mais acentuadas do campo geométrico.

De forma geral, os resultados indicam que o desempenho ótimo não está associado ao valor extremo de nenhum desses parâmetros isoladamente, mas sim ao equilíbrio entre eles. Malhas refinadas, porém geometricamente irregulares, podem não beneficiar adequadamente o método LB, enquanto vizinhanças muito amplas podem reduzir picos locais de erro ao custo de maior suavização e maior tempo de processamento. Nesse contexto, o AQFC com dois anéis de vizinhança destacou-se como a configuração mais equilibrada entre as analisadas, apresentando bom compromisso entre robustez, acurácia e eficiência computacional.

Capítulo 5

Conclusões e Trabalhos Futuros

5.1 Síntese das Contribuições

Neste trabalho foi desenvolvida e implementada, em Python, uma versão computacional do método *Algebraic Quadric Fitting Curvature* (AQFC), baseado na reconstrução local da superfície a partir de ajustes quadráticos. Em complemento, foi implementado o método clássico baseado no operador Laplace-Beltrami para o cálculo da curvatura média, possibilitando uma análise comparativa direta entre as duas abordagens.

Os algoritmos foram avaliados de forma sistemática em quatro geometrias de referência — esfera, cilindro, toro e elipsoide — considerando múltiplos níveis de refinamento da malha e diferentes tamanhos de vizinhança no cálculo do AQFC, correspondentes a um, dois e três anéis de vizinhos. Também foi utilizada uma segunda malha do elipsoide com construção otimizada.

A análise das curvas de erro mostrou que o método AQFC apresentou, de modo geral, o melhor desempenho global entre os métodos avaliados. Em particular, observou-se que o AQFC3 apresentou melhor desempenho nas malhas menos refinadas das geometrias de curvatura constante (esfera e cilindro), enquanto o AQFC2 apresentou melhores resultados nas malhas mais refinadas das geometrias com curvatura variável, como o toro e os elipsoides.

A comparação entre os mapas espaciais de erro de curvatura e as métricas de qualidade da malha evidenciou forte correlação entre regiões de maior distorção geométrica e os maiores erros observados no método baseado no operador Laplace–

Beltrami. Esses resultados confirmam a elevada sensibilidade desse método à regularidade da malha. O AQFC com vizinhança de um anel apresentou comportamento qualitativamente semelhante, porém com erros de menor magnitude. À medida que o tamanho da vizinhança foi ampliado, os erros passaram a se distribuir de forma mais homogênea ao longo da região considerada, reduzindo a sensibilidade do método às irregularidades locais da malha.

A análise da influência do tamanho da vizinhança no desempenho do AQFC indicou que vizinhanças maiores podem melhorar a precisão da estimativa em geometrias com variação de curvatura, desde que a malha apresente nível de refinamento suficiente para representar adequadamente a região considerada. No entanto, o aumento da vizinhança implica também maior custo computacional, uma vez que o ajuste quadrático passa a envolver um número maior de pontos.

Isso ocorre devido aos pontos adicionais considerados no ajuste. Por exemplo, em regiões pouco refinadas e com curvatura constante, como o tamanho da geometria permanece o mesmo, ao ampliar a vizinhança para três anéis a área representada torna-se maior e, como a curvatura não se altera, o ajuste tende a se adequar melhor aos pontos considerados.

Em contrapartida, ao refinarmos essa mesma região, o tamanho das arestas dos elementos diminui e o número de vizinhos aumenta. Como consequência, a área representada por uma mesma vizinhança torna-se menor, enquanto o número de pontos utilizados no ajuste aumenta. Nesse cenário, passam a existir mais pontos que não adicionam informação significativa ao ajuste, podendo inclusive torná-lo mais suscetível aos erros locais da malha.

Já em regiões de curvatura variável, considerar uma área de vizinhança maior não é suficiente se essa região não estiver adequadamente representada pela discretização da malha.

Esses resultados indicam que a escolha do tamanho da vizinhança no método AQFC deve considerar o compromisso entre precisão geométrica e custo computacional, além das características da geometria analisada e do nível de refinamento da malha.

Por fim, a comparação entre os métodos analisados permitiu identificar uma possível direção para o desenvolvimento futuro de abordagens híbridas adaptati-

vas para a estimativa de curvatura média em malhas triangulares. Tal estratégia poderia combinar a eficiência computacional do operador Laplace–Beltrami com a maior robustez geométrica do método AQFC, selecionando automaticamente o estimador mais adequado com base em métricas locais de qualidade da malha. Embora essa abordagem não tenha sido apresentada neste trabalho, os resultados obtidos fornecem uma base consistente para investigações futuras nessa direção.

5.2 Trabalhos Futuros

Embora as geometrias analisadas neste trabalho constituam casos clássicos de validação, escolhidos principalmente pela disponibilidade de soluções analíticas de curvatura, elas não representam a complexidade típica de interfaces presentes em escoamentos bifásicos reais, que costumam apresentar geometrias altamente irregulares e dinâmicas. Em particular, não foram consideradas superfícies com múltiplas curvaturas principais espacialmente variáveis de forma complexa, nem geometrias contendo furos, reconexões ou outras singularidades topológicas. Nesse sentido, a extensão dos testes para geometrias mais gerais, como superfícies de revolução com curvaturas espacialmente variáveis ou geometrias contendo furos e descontinuidades topológicas, poderia enriquecer significativamente as conclusões obtidas e ampliar a aplicabilidade dos resultados.

Adicionalmente, a análise desenvolvida considerou apenas três tamanhos fixos de vizinhança no método AQFC, correspondentes a um, dois e três anéis de vizinhos. Uma investigação mais detalhada, envolvendo estratégias de vizinhança adaptativa ou uma variação mais granular desse parâmetro, poderia revelar comportamentos mais sutis do método e possibilitar uma otimização contínua do compromisso entre precisão geométrica e custo computacional.

Outras extensões relevantes incluem o estudo sistemático da sensibilidade do método AQFC em relação aos pesos associados aos termos de posição e de normal, bem como a influência das tolerâncias de convergência do algoritmo de otimização SLSQP nos resultados finais. Também se mostra pertinente a avaliação do impacto de diferentes estratégias de estimação de normais na robustez e na precisão das estimativas de curvatura, especialmente em malhas irregulares.

Por fim, cabe registrar que, paralelamente às análises centrais deste trabalho, foram realizados testes exploratórios envolvendo uma estratégia híbrida de estimativa de curvatura. Esses ensaios tiveram caráter preliminar e não atingiram grau de maturidade suficiente para compor a metodologia principal nem para sustentar conclusões gerais. Ainda assim, por constituírem uma iniciativa diretamente relacionada aos desdobramentos desta pesquisa, seus principais aspectos e limitações são brevemente apresentados na subseção seguinte.

Abordagem Híbrida (Escopo e Testes Preliminares)

A metodologia híbrida proposta tem como objetivo combinar os métodos de Ajuste Quadrático e Laplace-Beltrami a partir de métricas de qualidade da malha, previamente discutidas. A ideia central consiste em atribuir, a cada ponto da malha, um valor ponderado com base na qualidade dos elementos em sua vizinhança, de modo a estabelecer um critério para a seleção do método mais adequado em cada região da interface.

Com o intuito de aumentar a precisão das estimativas de curvatura, foi definida uma métrica denominada *qualidade composta*, na qual diferentes métricas de qualidade da malha foram combinadas por meio de pesos atribuídos empiricamente e normalizadas em um intervalo entre 0 e 1. Entretanto, o desenvolvimento aprofundado dessa abordagem extrapolou o escopo inicialmente proposto para este trabalho e, por esse motivo, não foi explorado em maior detalhe.

Foram realizados testes preliminares utilizando valores de corte arbitrários para a métrica composta, os quais indicavam regiões da malha mais afastadas da condição ideal. Embora tais escolhas tenham produzido resultados promissores, com erros médios inferiores aos obtidos individualmente pelos métodos AQFC e Laplace-Beltrami, a robustez e a generalidade da abordagem ainda se mostraram insuficientes.

Uma alternativa considerada foi a aplicação de um procedimento de engenharia reversa, no qual, a partir de geometrias com curvatura analítica conhecida, seria possível selecionar, ponto a ponto, o método que apresentasse o menor desvio. A partir dessa análise, poderiam ser definidos valores de corte mais adequados para a métrica de qualidade, aumentando a taxa de acerto do critério híbrido. Contudo,

mesmo essa estratégia não conduziu a um nível satisfatório de precisão.

Diante dessas limitações, surgiu a possibilidade de combinar múltiplas condições e diferentes valores de corte, o que, apesar de potencialmente eficaz, implicaria um aumento significativo da complexidade do método e do esforço de implementação.

Por fim, considerou-se o uso de técnicas de *Machine Learning* para otimizar de forma automática esse conjunto de filtros e parâmetros. Ao avançar nessa direção, tornaram-se evidentes novas variáveis relevantes, como o grau de variação da curvatura local e a limitação imposta pelo número de vizinhos considerados no AQFC, as quais deveriam ser incorporadas como parâmetros adicionais do modelo. Embora promissora, a metodologia híbrida, em função do aumento substancial da complexidade e da extensão do estudo, essa abordagem foi considerada incompatível com o escopo e o prazo de um Trabalho de Conclusão de Curso, motivo pelo qual não foi incluída no presente trabalho.

Referências Bibliográficas

- [1] GAUTHIER, S., PUECH, W., BÉNIÈRE, R., et al., “Digitized 3D mesh segmentation based on curvature analysis”, *Electronic Imaging*, v. 29, pp. 33–38, 2017.
- [2] ANJOS, G. R. D., *A 3D ALE Finite Element Method for Two-Phase Flows with Phase Change*, Ph.D. Thesis, École Polytechnique Fédérale de Lausanne, Lausanne, Suisse, 2012, Tese de Doutorado.
- [3] MEYER, M., DESBRUN, M., SCHRÖDER, P., et al., “Discrete differential-geometry operators for triangulated 2-manifolds”, In: *Visualization and mathematics III*, chap. 2, pp. 35–57, Springer, 2003.
- [4] SANTOS, D. B. V., *Finite Element Simulation of Three-Dimensional Two-Phase Flows in Complex Geometries*, Master’s Thesis, Universidade Federal do Rio de Janeiro, COPPE, Rio de Janeiro, Brasil, 2024, Dissertação de Mestrado, Orientador: Gustavo Rabello dos Anjos.
- [5] BRATLAND, O., “Pipe Flow 2: Multi-phase Flow Assurance”, Disponível em: <https://www.drbratland.com/PipeFlow2/>, 2010, Acesso em: 22 set. 2025.
- [6] POMEAU, Y., *Surface tension: from fundamental principles to applications in liquids and in solids*, Ph.D. Thesis, Warsaw School of Statistical Physics, 2013.
- [7] BRACKBILL, J. U., KOTHE, D. B., ZEMACH, C., “A continuum method for modeling surface tension”, *Journal of computational physics*, v. 100, n. 2, pp. 335–354, 1992.

- [8] HARVIE, D. J., DAVIDSON, M., RUDMAN, M., “An analysis of parasitic current generation in volume of fluid simulations”, *Applied mathematical modelling*, v. 30, n. 10, pp. 1056–1066, 2006.
- [9] MAKOVNÍK, M., CHALMOVIANSKY, P., “Curvature estimation for meshes via algebraic quadric fitting”, *arXiv preprint arXiv:2304.08909*, 2023.
- [10] DO CARMO, M. P., *Differential geometry of curves and surfaces: revised and updated second edition*. Courier Dover Publications, 2016.
- [11] KREYSZIG, E., *Differential geometry*. v. 11. Courier Corporation, 1991.
- [12] BOTSCH, M., KOBELT, L., PAULY, M., et al., *Polygon mesh processing*. CRC press, 2010.
- [13] RAZDAN, A., BAE, M., “Curvature estimation scheme for triangle meshes using biquadratic Bézier patches”, *Computer-Aided Design*, v. 37, n. 14, pp. 1481–1491, 2005.
- [14] ZHIHONG, M., GUO, C., YANZHAO, M., et al., “Curvature estimation for meshes based on vertex normal triangles”, *Computer-Aided Design*, v. 43, n. 12, pp. 1561–1566, 2011.
- [15] GATZKE, T., GRIMM, C., *Improved curvature estimation on triangular meshes*, Tech. Rep. WUCSE-2004-9, All Computer Science and Engineering Research, 2004, Disponível em: <https://doi.org/10.7936/K7PV6HQZ>. Acesso em: dia 10/03/2026.
- [16] GATZKE, T. D., GRIMM, C. M., “Estimating curvature on triangular meshes”, *International journal of shape modeling*, v. 12, n. 01, pp. 1–28, 2006.
- [17] DESBRUN, M., MEYER, M., SCHRÖDER, P., et al., “Implicit fairing of irregular meshes using diffusion and curvature flow”. In: *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 317–324, 1999.
- [18] WARDETZKY, M., MATHUR, S., KÄLBERER, F., et al., “Discrete Laplace operators: no free lunch”. In: *Symposium on Geometry processing*, v. 33, p. 37, 2007.

- [19] XU, G., “Discrete Laplace–Beltrami operators and their convergence”, *Computer aided geometric design*, v. 21, n. 8, pp. 767–784, 2004.
- [20] BOBENKO, A. I., SPRINGBORN, B. A., “A discrete Laplace–Beltrami operator for simplicial surfaces”, *Discrete & Computational Geometry*, v. 38, n. 4, pp. 740–756, 2007.
- [21] TAUBIN, G., “Estimating the tensor of curvature of a surface from a polyhedral approximation”. In: *Proceedings of IEEE International Conference on Computer Vision*, pp. 902–907, 1995.
- [22] THEISEL, H., ROSSI, C., ZAYER, R., et al., “Normal based estimation of the curvature tensor for triangular meshes”. In: *12th Pacific Conference on Computer Graphics and Applications, 2004. PG 2004. Proceedings.*, pp. 288–297, 2004.
- [23] RUSINKIEWICZ, S., “Estimating curvatures and their derivatives on triangle meshes”. In: *Proceedings. 2nd International Symposium on 3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004.*, pp. 486–493, 2004.
- [24] HAMANN, B., “Curvature approximation for triangulated surfaces”, In: *Geometric modelling*, chap. 10, pp. 139–153, Springer, 1992.
- [25] CAZALS, F., POUGET, M., “Estimating differential quantities using polynomial fitting of osculating jets”, *Computer aided geometric design*, v. 22, n. 2, pp. 121–146, 2005.
- [26] PETITJEAN, S., “A survey of methods for recovering quadrics in triangle meshes”, *ACM Computing Surveys (CSUR)*, v. 34, n. 2, pp. 211–262, 2002.
- [27] GOLDMAN, R., “Curvature formulas for implicit curves and surfaces”, *Computer Aided Geometric Design*, v. 22, n. 7, pp. 632–658, 2005.
- [28] KALOGERAKIS, E., SIMARI, P., NOWROUZEZAHRAI, D., et al., “Robust statistical estimation of curvature on discretized surfaces”. In: *Symposium on Geometry Processing*, v. 13, pp. 110–114, 2007.

- [29] SURAZHISKY, T., MAGID, E., SOLDEA, O., et al., “A comparison of gaussian and mean curvatures estimation methods on triangular meshes”. In: *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, v. 1, pp. 1021–1026, 2003.
- [30] MAGID, E., SOLDEA, O., RIVLIN, E., “A comparison of Gaussian and mean curvature estimation methods on triangular meshes of range image data”, *Computer Vision and Image Understanding*, v. 107, n. 3, pp. 139–159, 2007.
- [31] KRONENBERGER, M., WIRJADI, O., HAGEN, H., “Empirical comparison of curvature estimators on volume images and triangle meshes”, *IEEE transactions on visualization and computer graphics*, v. 25, n. 10, pp. 3032–3041, 2018.
- [32] SORKINE, O., “Laplacian mesh processing”, *Eurographics (State of the Art Reports)*, v. 4, n. 4, pp. 1, 2005.
- [33] VÁŠA, L., VANĚČEK, P., PRANTL, M., et al., “Mesh Statistics for Robust Curvature Estimation”, *Computer Graphics Forum*, v. 35, n. 5, pp. 271–280, 2016.
- [34] KNUPP, P. M., “Algebraic mesh quality metrics”, *SIAM journal on scientific computing*, v. 23, n. 1, pp. 193–218, 2001.
- [35] SORGENTE, T., BIASOTTI, S., MANZINI, G., et al., “A survey of indicators for mesh quality assessment”, *Computer graphics forum*, v. 42, n. 2, pp. 461–483, 2023.
- [36] FARIN, G., “Shape measures for triangles”, *IEEE transactions on visualization and computer graphics*, v. 18, n. 1, pp. 43–46, 2010.
- [37] SHEWCHUK, J. R., “Delaunay refinement algorithms for triangular mesh generation”, *Computational geometry*, v. 22, n. 1-3, pp. 21–74, 2002.
- [38] GEUZAIN, C., REMACLE, J.-F., “Gmsh: A three-dimensional finite element mesh generator with built-in pre- and post-processing facilities”,

International Journal for Numerical Methods in Engineering, v. 79, n. 11, pp. 1309–1331, 2009.

- [39] ANSYS INC., *ANSYS Meshing User's Guide*, 2021.
- [40] VTK DEVELOPERS, “vtkMeshQuality: TriangleAspectRatio description”, Disponível em: <https://vtk.org/doc/nightly/html/classvtkMeshQuality.html>, 2024.
- [41] SPIVAK, M., *A Comprehensive Introduction to Differential Geometry, Volume One*. 3rd ed. Publish or Perish, Inc.: Houston, Texas, 1999.

Apêndice A

Códigos Utilizados

A.1 Curvatura por Laplace-Beltrami

```
# FUNCOES AUXILIARES
```

```
# Calculo das areas circuncentricas
```

```
def calc_A(no1_coords, no2_coords, no3_coords):
```

```
    # no1_coords, no2_coords, no3_coords sao arrays numpy [x, y, z]
```

```
    a = np.sqrt((no2_coords[0] - no3_coords[0]) ** 2 + (no2_coords[1] -  
        no3_coords[1]) ** 2 + (  
        no2_coords[2] - no3_coords[2]) ** 2)
```

```
    b = np.sqrt((no1_coords[0] - no3_coords[0]) ** 2 + (no1_coords[1] -  
        no3_coords[1]) ** 2 + (  
        no1_coords[2] - no3_coords[2]) ** 2)
```

```
    c = np.sqrt((no2_coords[0] - no1_coords[0]) ** 2 + (no2_coords[1] -  
        no1_coords[1]) ** 2 + (  
        no2_coords[2] - no1_coords[2]) ** 2)
```

```
    pa = a * a * (b * b + c * c - a * a)
```

```
    pb = b * b * (a * a + c * c - b * b)
```

```
    pc = c * c * (b * b + a * a - c * c)
```

```
    # Evitar divisao por zero se pa + pb + pc for muito pequeno
```

```

sum_pa_pb_pc = pa + pb + pc
if np.isclose(sum_pa_pb_pc, 0):
    # Lidar com o caso de degeneracao, por exemplo, retornando 0 ou
    # levantando um erro
    return 0.0 # Ou np.nan, dependendo do comportamento desejado

cx = (pa * no1_coords[0] + pb * no2_coords[0] + pc * no3_coords[0]) /
    sum_pa_pb_pc
cy = (pa * no1_coords[1] + pb * no2_coords[1] + pc * no3_coords[1]) /
    sum_pa_pb_pc
cz = (pa * no1_coords[2] + pb * no2_coords[2] + pc * no3_coords[2]) /
    sum_pa_pb_pc

p_med_13_x = (no1_coords[0] + no3_coords[0]) / 2
p_med_13_y = (no1_coords[1] + no3_coords[1]) / 2
p_med_13_z = (no1_coords[2] + no3_coords[2]) / 2
p_med_12_x = (no1_coords[0] + no2_coords[0]) / 2
p_med_12_y = (no1_coords[1] + no2_coords[1]) / 2
p_med_12_z = (no1_coords[2] + no2_coords[2]) / 2

m1 = np.array([cx - p_med_13_x, cy - p_med_13_y, cz - p_med_13_z])
m2 = np.array([cx - p_med_12_x, cy - p_med_12_y, cz - p_med_12_z])

dist_t1 = np.sqrt((no3_coords[0] - no1_coords[0]) ** 2 +
    (no3_coords[1] - no1_coords[1]) ** 2 + (
        no3_coords[2] - no1_coords[2]) ** 2)
# Evitar divisao por zero
t1 = np.array([(no3_coords[0] - no1_coords[0]) / dist_t1,
    (no3_coords[1] - no1_coords[1]) / dist_t1,
    (no3_coords[2] - no1_coords[2]) / dist_t1]) if dist_t1
    != 0 else np.zeros(3)

dist_t2 = np.sqrt((no2_coords[0] - no1_coords[0]) ** 2 +
    (no2_coords[1] - no1_coords[1]) ** 2 + (

```

```

        no2_coords[2] - no1_coords[2])) ** 2)
# Evitar divisao por zero
t2 = np.array([(no2_coords[0] - no1_coords[0]) / dist_t2,
               (no2_coords[1] - no1_coords[1]) / dist_t2,
               (no2_coords[2] - no1_coords[2]) / dist_t2]) if dist_t2
               != 0 else np.zeros(3)

t1 = t1 * dist_t1 / 2
t2 = t2 * dist_t2 / 2

vet1 = np.cross(t1, m1)
vet2 = np.cross(t2, m2)
mod_vet1 = np.linalg.norm(vet1)
mod_vet2 = np.linalg.norm(vet2)

return 0.5 * (mod_vet1 + mod_vet2)

% A area de Voronoi de um determinado no e obtida repetindo esse
    processo para cada triangulo ao qual ele pertence e somando os
    valores das areas

# Calculo das normais
def calc_nx_ny_nz(X, Y, Z, IEN, normal_inv=False):
    npoints = len(X)
    nx_vec = np.zeros(npoints)
    ny_vec = np.zeros(npoints)
    nz_vec = np.zeros(npoints)

    for el_nodes in IEN:
        no1_coords = np.array([X[el_nodes[0]], Y[el_nodes[0]],
                               Z[el_nodes[0]]])
        no2_coords = np.array([X[el_nodes[1]], Y[el_nodes[1]],
                               Z[el_nodes[1]]])

```

```

no3_coords = np.array([X[el_nodes[2]], Y[el_nodes[2]],
                       Z[el_nodes[2]]])

v1 = no2_coords - no1_coords
v2 = no3_coords - no1_coords

result = np.cross(v1, v2)
modulo = np.linalg.norm(result)

if modulo != 0:
    result /= modulo
else:
    result = np.zeros(3) # Caso degenerado, normal zero

nx_vec[el_nodes[0]] += result[0]
ny_vec[el_nodes[0]] += result[1]
nz_vec[el_nodes[0]] += result[2]

nx_vec[el_nodes[1]] += result[0]
ny_vec[el_nodes[1]] += result[1]
nz_vec[el_nodes[1]] += result[2]

nx_vec[el_nodes[2]] += result[0]
ny_vec[el_nodes[2]] += result[1]
nz_vec[el_nodes[2]] += result[2]

for i in range(npoints):
    modulo = np.sqrt(nx_vec[i] ** 2 + ny_vec[i] ** 2 + nz_vec[i] ** 2)
    if modulo != 0:
        if normal_inv:
            modulo = -modulo
        nx_vec[i] /= modulo
        ny_vec[i] /= modulo
        nz_vec[i] /= modulo

```

```

        # Se modulo for zero, as componentes ja sao zero, nao precisa
        dividir

    return nx_vec, ny_vec, nz_vec

# Criacao da matriz laplaciana esparsa
def calc_K_bolha(X, Y, Z, IEN):
    npoints = len(X)
    K_triplets = []
    area_vec = np.zeros(npoints)

    for el_nodes in IEN:
        no1_coords = np.array([X[el_nodes[0]], Y[el_nodes[0]],
                               Z[el_nodes[0]]])
        no2_coords = np.array([X[el_nodes[1]], Y[el_nodes[1]],
                               Z[el_nodes[1]]])
        no3_coords = np.array([X[el_nodes[2]], Y[el_nodes[2]],
                               Z[el_nodes[2]]])

        ji = no2_coords - no1_coords
        ik = no1_coords - no3_coords
        kj = no3_coords - no2_coords

        mod_ksi = np.linalg.norm(ji)
        ksi = ji / mod_ksi if mod_ksi != 0 else np.zeros(3)

        pvet1 = np.cross(ji, ik)
        pvet2 = np.cross(pvet1, ji)
        mod_eta = np.linalg.norm(pvet2)
        eta = pvet2 / mod_eta if mod_eta != 0 else np.zeros(3)

        bi = np.dot(kj, eta)
        bj = np.dot(ik, eta)

```

```

bk = np.dot(ji, eta)
ci = np.dot(kj, ksi)
cj = np.dot(ik, ksi)
ck = np.dot(ji, ksi)

area = 0.5 * np.linalg.norm(pvet1)

temp1 = calc_A(no1_coords, no2_coords, no3_coords)
temp2 = calc_A(no2_coords, no3_coords, no1_coords)
temp3 = calc_A(no3_coords, no1_coords, no2_coords)

# Correcao de area para triangulos obtusos
if (temp1 + temp2 + temp3) * 0.9999 > area:
    a_len = np.linalg.norm(no2_coords - no3_coords) # lado oposto
        ao no 1
    b_len = np.linalg.norm(no1_coords - no3_coords) # lado oposto
        ao no 2
    c_len = np.linalg.norm(no1_coords - no2_coords) # lado oposto
        ao no 3

    # no 1 obtuso <=> a_len e o maior
    if a_len >= b_len and a_len >= c_len:
        diff = area - temp1
        denom = b_len + c_len
        if denom != 0:
            temp2 = diff * (c_len / denom) # AJ = c/(b+c) * (A - AI)
            temp3 = diff * (b_len / denom) # AK = b/(b+c) * (A - AI)
        else:
            temp2 = temp3 = 0.0

    # no 2 obtuso <=> b_len e o maior
    elif b_len >= a_len and b_len >= c_len:
        diff = area - temp2
        denom = a_len + c_len

```



```

        if denom != 0:
            temp1 = diff * (c_len / denom)
            temp3 = diff * (a_len / denom)
        else:
            temp1 = temp3 = 0.0

# no 3 obtuso <=> c_len e o maior
else:
    diff = area - temp3
    denom = a_len + b_len
    if denom != 0:
        temp1 = diff * (b_len / denom)
        temp2 = diff * (a_len / denom)
    else:
        temp1 = temp2 = 0.0

area_vec[el_nodes[0]] += temp1
area_vec[el_nodes[1]] += temp2
area_vec[el_nodes[2]] += temp3

mat_k = np.zeros((3, 3))
mat_k[0, 0] = bi * bi
mat_k[0, 1] = bi * bj
mat_k[0, 2] = bi * bk
mat_k[1, 0] = bj * bi
mat_k[1, 1] = bj * bj
mat_k[1, 2] = bj * bk
mat_k[2, 0] = bk * bi
mat_k[2, 1] = bk * bj
mat_k[2, 2] = bk * bk
mat_k = mat_k / (4 * area) if area != 0 else np.zeros((3, 3))

mat_c = np.zeros((3, 3))
mat_c[0, 0] = ci * ci

```

```

    mat_c[0, 1] = ci * cj
    mat_c[0, 2] = ci * ck
    mat_c[1, 0] = cj * ci
    mat_c[1, 1] = cj * cj
    mat_c[1, 2] = cj * ck
    mat_c[2, 0] = ck * ci
    mat_c[2, 1] = ck * cj
    mat_c[2, 2] = ck * ck
    mat_c = mat_c / (4 * area) if area != 0 else np.zeros((3, 3))

    for p in range(3):
        for q in range(3):
            K_triplets.append((el_nodes[p], el_nodes[q], mat_k[p, q] +
                               mat_c[p, q]))
    return K_triplets, area_vec

# Funcao principal para calcular a curvatura
def laplace_beltrami(X, Y, Z, IEN):
    npoints = len(X)

    nx_vec, ny_vec, nz_vec = calc_nx_ny_nz(X, Y, Z, IEN)

    K_triplets, area_vec = calc_K_bolha(X, Y, Z, IEN)

    # Para construir a matriz esparsa, usaremos scipy.sparse
    rows = [t[0] for t in K_triplets]
    cols = [t[1] for t in K_triplets]
    data = [t[2] for t in K_triplets]

    K = coo_matrix((data, (rows, cols)), shape=(npoints, npoints)).tocsr()

    x_pos = X
    y_pos = Y

```

```

z_pos = Z

Kx_bolha = -K @ x_pos
Ky_bolha = -K @ y_pos
Kz_bolha = -K @ z_pos

mod_forca = np.sqrt(Kx_bolha ** 2 + Ky_bolha ** 2 + Kz_bolha ** 2)

# Evitar divisao por zero para kappa
kappa = np.zeros(npoints)
non_zero_area_indices = area_vec != 0
kappa[non_zero_area_indices] = mod_forca[non_zero_area_indices] /
    np.abs(area_vec[non_zero_area_indices])

for i in range(npoints):
    prod_vetorial = Kx_bolha[i] * nx_vec[i] + Ky_bolha[i] * ny_vec[i]
        + Kz_bolha[i] * nz_vec[i]
    if prod_vetorial > 0:
        kappa[i] = -kappa[i]

return kappa, mod_forca, area_vec, nx_vec, ny_vec, nz_vec

```

A.2 Curvatura por Ajuste Quadrático Local

```

# ALGEBRIC QUADRIC FITTING CURVATURE - AQFC

# Funcoes auxiliares
# Definicao da quadrica e seu gradiente e Hessiana em termos dos
    coeficientes 'a'
# Estes sao os 10 coeficientes que serao resolvidos pelo sistema linear
# a = [a11, a22, a33, a12, a13, a23, a14, a24, a34, a44]

```

```

def q_func_numeric(x, y, z, a):
    #  $q(x,y,z) = a_{11}x^2 + a_{22}y^2 + a_{33}z^2 + a_{12}xy + a_{13}xz + a_{23}yz +$ 
    #  $a_{14}x + a_{24}y + a_{34}z + a_{44}$ 
    return (
        a[0]*x**2 + a[1]*y**2 + a[2]*z**2 +
        a[3]*x*y + a[4]*x*z + a[5]*y*z +
        a[6]*x + a[7]*y + a[8]*z + a[9]
    )

def grad_q_func_numeric(x, y, z, a):
    # Derivadas parciais de q em relacao a x, y, z
    dq_dx = 2*a[0]*x + a[3]*y + a[4]*z + a[6]
    dq_dy = 2*a[1]*y + a[3]*x + a[5]*z + a[7]
    dq_dz = 2*a[2]*z + a[4]*x + a[5]*y + a[8]
    return np.array([dq_dx, dq_dy, dq_dz])

def hess_q_func_numeric(x, y, z, a):
    # Hessiana de q (matriz 3x3 de segundas derivadas parciais)
    #  $H_{qq} = \begin{bmatrix} d^2q_{dx^2} & d^2q_{dxdy} & d^2q_{dxdz} \\ d^2q_{dydx} & d^2q_{dy^2} & d^2q_{dydz} \\ d^2q_{dzdx} & d^2q_{dzdy} & d^2q_{dz^2} \end{bmatrix}$ 

    d2q_dx2 = 2*a[0]
    d2q_dy2 = 2*a[1]
    d2q_dz2 = 2*a[2]
    d2q_dxdy = a[3]
    d2q_dxdz = a[4]
    d2q_dydz = a[5]

    H = np.array([
        [d2q_dx2, d2q_dxdy, d2q_dxdz],
        [d2q_dxdy, d2q_dy2, d2q_dydz],
        [d2q_dxdz, d2q_dydz, d2q_dz2]
    ])

```

```

return H

# Funcao objetivo para a otimizacao (minimize)
def objetivo(xyz, V_central, a_coeffs):
    # Esta e a funcao que scipy.optimize.minimize tentara minimizar.
    # Ela deve retornar o valor de q(xyz)^2
    return q_func_numeric(xyz[0], xyz[1], xyz[2], a_coeffs)**2

# Funcao para calcular nos e elementos vizinhos
def elementos_vizinhanca(IEN):
    # Encontrar o numero total de vertices
    num_vertices = np.max(IEN) + 1 # +1 porque os indices geralmente
        comecam em 0

    # Inicializar as listas de vizinhos
    nos_vizinhos = [[] for _ in range(num_vertices)]
    elementos_vizinhos = [[] for _ in range(num_vertices)]

    # Percorre elementos e preenche elementos_vizinhos e nos_vizinhos (1
        ordem)
    for elem_idx, elemento in enumerate(IEN):
        # Garantir lista de ints
        elem_nodes = [int(x) for x in elemento]
        for no in elem_nodes:
            elementos_vizinhos[no].append(elem_idx)
            # Adiciona os outros nos do elemento como vizinhos (sem
                duplicar)
            for outro_no in elem_nodes:
                if outro_no != no and outro_no not in nos_vizinhos[no]:
                    nos_vizinhos[no].append(outro_no)

    # 1 vizinhanca pronta
    vizinhos_geral = [list(v) for v in nos_vizinhos]

```

```

# --- 2 vizinhanca ---
vizinhos_segundo_geral = []
for i in range(num_vertices):
    primeiro = set(vizinhos_geral[i]) # 1 ordem
    segundo = set()
    # Uniao dos vizinhos dos vizinhos (excluindo o proprio no)
    for v in primeiro:
        segundo.update(vizinhos_geral[v])
    # Remove 1 ordem e o proprio no
    segundo -= primeiro
    segundo.discard(i)
    vizinhos_segundo_geral.append(list(segundo))

# --- 3 vizinhanca ---
vizinhos_terceiro_geral = []
for i in range(num_vertices):
    primeiro = set(vizinhos_geral[i])
    segundo = set(vizinhos_segundo_geral[i])
    terceiro = set()
    # Pega vizinhos de 1 ordem dos nos do 2 guarda-chuva
    for v2 in segundo:
        terceiro.update(vizinhos_geral[v2])
    # Remove os nos que ja estao em 1 ou 2 ordem e o proprio no
    terceiro -= primeiro
    terceiro -= segundo
    terceiro.discard(i)
    vizinhos_terceiro_geral.append(list(terceiro))

return elementos_vizinhos, vizinhos_geral, vizinhos_segundo_geral,
       vizinhos_terceiro_geral

# Funcao para calcular normais resultantes

```

```

def calcular_normais_vertices(IEN, coordenadas, neighbors):
    # Calcular nos e elementos vizinhos
    elementos_vizinhos, nos_vizinhos, _, _ = neighbors

    # Numero de vertices
    num_vertices = coordenadas.shape[0]

    # Inicializar lista de normais dos vertices
    normais_vertices = [[] for _ in range(num_vertices)]

    # Calcular normais das faces
    normais_faces = {}
    for i, triangulo in enumerate(IEN):
        # Obter coordenadas dos vertices do triangulo
        v0 = coordenadas[triangulo[0]]
        v1 = coordenadas[triangulo[1]]
        v2 = coordenadas[triangulo[2]]

        # Calcular vetores das arestas
        edge1 = v1 - v0
        edge2 = v2 - v0

        # Calcular normal da face (produto vetorial)
        normal = np.cross(edge1, edge2)

        # Normalizar
        norm = np.linalg.norm(normal)

        normais_faces[i] = (normal / norm).tolist()

    # Para cada vertice
    for vertice_id in range(num_vertices):
        # Inicializar normal resultante

```

```

normal_resultante = np.zeros(3)

# Para cada elemento vizinho do vertice
for elem_idx in elementos_vizinhos[vertice_id]:
    # Obter o triangulo
    triangulo = IEN[elem_idx]

    # Encontrar a posicao do vertice no triangulo
    pos_vertice = np.where(triangulo == vertice_id)[0][0]

    # Obter coordenadas dos vertices do triangulo
    v = [coordenadas[int(triangulo[i])] for i in range(3)]

    # Calcular vetores das arestas adjacentes ao vertice
    if pos_vertice == 0:
        e1 = v[1] - v[0]
        e2 = v[2] - v[0]
    elif pos_vertice == 1:
        e1 = v[2] - v[1]
        e2 = v[0] - v[1]
    else: # pos_vertice == 2
        e1 = v[0] - v[2]
        e2 = v[1] - v[2]

    # Normalizar os vetores das arestas
    l1 = np.linalg.norm(e1)
    l2 = np.linalg.norm(e2)

    if l1 > 0 and l2 > 0:
        e1 = e1 / l1
        e2 = e2 / l2

    # Calcular o angulo entre as arestas
    cos_angle = np.clip(np.dot(e1, e2), -1.0, 1.0)

```



```

        angle = np.arccos(cos_angle)

        # Ponderar a normal da face pelo angulo
        normal_face = np.array(normais_faces[elem_idx])
        normal_resultante += angle * normal_face

    # Normalizar a normal resultante
    norm = np.linalg.norm(normal_resultante)

    # Converter para lista Python padrao
    normais_vertices[vertexe_id] = (normal_resultante / norm).tolist()

return normais_vertices

# Funcao principal primeira vizinhanca

def AQFC_curvature_NV(X, Y, Z, neighbors, normals):
    # Obtem informacoes das funcoes criadas
    elementos_geral, vizinhos_geral, vizinhos_segundo_geral,
        vizinhos_terceiro_geral = neighbors
    Ngeral = normals

    # Para estender a vizinhanca basta somar os vizinhos, ex:
    # 1V = vizinhos_geral
    # 2V = 1V + vizinhos_segundo_geral
    # 3V = 2V + vizinhos_terceiro_geral
    # Aqui e mais um exemplo, mas quando for rodar o codigo, nao precisa
    # chamar vizinhancas desnecessarias e ocupar espaco na memoria, pode
    # tambem comentar a funcao elementos_vizinhanca para nao calcular a
    # vizinhanca extendida se for de menor grau

```

```

npoints = len(X)

# Loop principal
pointsZq, DE_tds, Hs = [], [], []
# erros = [] # tuplas (indice, (x,y,z))

for no in range(npoints):
    # print(f"No {no}")
    V = np.array([X[no], Y[no], Z[no]]) # Ponto central
    N = np.array([Ngeral[no][0], Ngeral[no][1], Ngeral[no][2]])

    vizinhos = 1V[no] # ALTERAR AQUI A VIZINHANCA; 2V se quiser com a
                        # segunda e 3V se quiser a terceira

    # Inicialize A e b para o no atual
    A = np.zeros((10, 10))
    b = np.zeros(10)

    for i in vizinhos:
        # print(f"Vizinho {i} do no {no}")
        Vi = np.array([X[i], Y[i], Z[i]])
        Ni = np.array([Ngeral[i][0], Ngeral[i][1], Ngeral[i][2]])

        # Calcular pesos W_Vi e W_Ni
        W_Vi = math.exp(- ((np.linalg.norm(V - Vi)) ** 4))
        W_Ni = 1e-4 * math.exp(- ((np.linalg.norm(N - Ni)) ** 2))

        xi, yi, zi = Vi

        # Vetor c_i (1x10)
        c_i = np.array([[xi ** 2, yi ** 2, zi ** 2, xi * yi, xi * zi,
                        yi * zi, xi, yi, zi,
                        1.0]])

        # Matriz G_i (3x10)

```

```

G_i = np.array([
    [2 * xi, 0, 0, yi, zi, 0, 1, 0, 0, 0],
    [0, 2 * yi, 0, xi, 0, zi, 0, 1, 0, 0],
    [0, 0, 2 * zi, 0, xi, yi, 0, 0, 1, 0]
])

# Acumular na matriz A e vetor b
A += W_Vi * (c_i.T @ c_i) + W_ni * (G_i.T @ G_i)
b += W_ni * (G_i.T @ Ni)

# print(f"\nA_{no}: {A}")
# print(f"\nb_{no}: {b}")

# Resolver o sistema linear Ax = solucao
try:
    solucao = np.linalg.solve(A, b)
except np.linalg.LinAlgError:
    # Se a matriz for singular, use a pseudo-inversa (mais robusto)
    solucao = np.linalg.pinv(A) @ b

# 'solucao' agora contem os 10 coeficientes da quadrica:
# [a11, a22, a33, a12, a13, a23, a14, a24, a34, a44]

# Funcoes numericas de q, gradiente e Hessiana usando os
# coeficientes 'solucao'
# q_func ja e a funcao q_func_numeric definida acima
# grad_num ja e a funcao grad_q_func_numeric definida acima
hess_num = hess_q_func_numeric(V[0], V[1], V[2], solucao) #
# Hessiana e constante para quadricas

# Restricao: q(x, y, z) = 0
# A funcao lambda agora passa os coeficientes 'a' para
# q_func_numeric

```

```

restricao = {'type': 'eq', 'fun': lambda xyz:
            q_func_numeric(xyz[0], xyz[1], xyz[2], solucao)}

# Otimizacao
# A funcao objetivo agora passa os coeficientes 'a' para a funcao
    q_func_numeric
resultado = minimize(lambda xyz: objetivo(xyz, V, solucao), x0=V,
                    constraints=restricao, method='SLSQP',
                    options={'maxiter': 500})

# Projecao p(V)
if not resultado.success:
    pointsZq.append([np.nan] * 3)
    DE_tds.append(np.nan)
    Hs.append(np.nan)
    print(f"Otimizacao falhou para o no {no}: {resultado.message}")
    continue
PV = resultado.x

distancia = np.sqrt(resultado.fun)
pointsZq.append(PV)
DE_tds.append(distancia)

# Calculo da curvatura media H
grad_at_PV = grad_q_func_numeric(PV[0], PV[1], PV[2], solucao)

norm_grad = np.linalg.norm(grad_at_PV)

s = np.linalg.norm(N) * grad_at_PV / norm_grad

norm_s = np.linalg.norm(s)

# numerador:  $s^T H_q s - ||s||^2 * \text{tr}(H_q)$ 
# Note que  $||s||^2 = 1$ , pois s e um vetor unitario.

```

```

term1 = s.T @ hess_num @ s

term2 = norm_s ** 2 * np.trace(hess_num) # ||s||^2 e 1

numerator = term1 - term2

# denominador: 2 * ||s||^3
# Note que ||s||^3 = 1, pois s e um vetor unitario.
denominator = 2 * norm_s ** 3 # 2 * ||s||^3 e 2

# Curvatura
H = -1.0 * numerator / denominator

Hs.append(H)
Hs = np.array(Hs).flatten()

return pointsZq, DE_tds, Hs

```

A.3 Curvaturas Analíticas

CURVATURAS ANALITICAS

```

def curvatura_esfera(vertices):
    """
    Calcula a curvatura media analitica para pontos em uma esfera.

    A curvatura media (H) de uma esfera de raio R e constante e dada por:
    H = 1 / R

    """

    R = 0.5

```

```

# A curvatura media e constante para todos os pontos da esfera
curvatura = 1.0 / R

# Retorna uma lista com o mesmo valor de curvatura para cada ponto
return [curvatura] * len(vertices)

def curvatura_elipsoide(a, b, c, vertices):
    """
    Calcula a curvatura media analitica para pontos em um elipsoide.

    A curvatura media (H) de um elipsoide  $(x/a)^2 + (y/b)^2 + (z/c)^2 = 1$ 
    em um ponto (x, y, z) e dada por:
    
$$p = \sqrt{x^2/a^4 + y^2/b^4 + z^2/c^4}$$


$$H = (1 / (2*p)) * (1/a^2 + 1/b^2 + 1/c^2) - (1 / (2*p^3)) * (x^2/a^6 + y^2/b^6 + z^2/c^6)$$

    """

    Parametros:
    -----
        Assume-se que os pontos estao aproximadamente na superficie do
        elipsoide e centrados na origem.
    a : float
        Semi-eixo do elipsoide ao longo do eixo x.
    b : float
        Semi-eixo do elipsoide ao longo do eixo y.
    c : float
        Semi-eixo do elipsoide ao longo do eixo z.
    """

    curvaturas = []
    x, y, z = vertices.T

```

```

# Pre-calcular potencias dos semi-eixos para eficiencia
a2, b2, c2 = a * a, b * b, c * c
a4, b4, c4 = a2 * a2, b2 * b2, c2 * c2
a6, b6, c6 = a4 * a2, b4 * b2, c4 * c2

# Calcular o termo p = sqrt(x^2/a^4 + y^2/b^4 + z^2/c^4)
p_squared = (x * x / a4) + (y * y / b4) + (z * z / c4)
# Adicionar pequena tolerancia para evitar divisao por zero se p for
# muito proximo de zero
p = np.sqrt(p_squared + 1e-15)
p3 = p * p_squared + 1e-15 # p^3

# Calcular termos da formula da curvatura media
termo1 = (1.0 / (2.0 * p)) * (1.0 / a2 + 1.0 / b2 + 1.0 / c2)
termo2 = (1.0 / (2.0 * p3)) * (x * x / a6 + y * y / b6 + z * z / c6)

# Curvatura media H
H = termo1 - termo2

return H.tolist()

def curvatura_toro(vertices):
    """
    Calcula a curvatura media analitica para pontos em um toro.

    A curvatura media (H) de um toro com raio maior R e raio menor r em
    um ponto (x, y, z) e dada por:
    
$$\rho = \sqrt{x^2 + y^2}$$


$$H = (2 * \rho - R) / (2 * r * \rho)$$


    Parametros:
    -----
    """

```

```

R : float
    Raio maior do toro (distancia do centro do toro ao centro do tubo).
r : float
    Raio menor do toro (raio do tubo).

"""

R = 0.5
r = 0.4

curvaturas = []
x, y, z = vertices.T
# z nao e usado diretamente na formula final, mas esta implicito na
    localizacao do ponto

# Calcular rho = sqrt(x^2 + y^2)
rho_squared = x * x + y * y
# Adicionar pequena tolerancia para evitar divisao por zero no centro
    do toro
rho = np.sqrt(rho_squared + 1e-15)

# Calcular a curvatura media H
numerador = (2.0 * rho - R)
denominador = (2.0 * r * rho)
H = numerador / denominador

return H.tolist()

def curvatura_cilindro(vertices, R=0.5, H=2.0, tol=1e-6,
    singular_value=np.nan):
    """
    Retorna as curvaturas analiticas de um cilindro (e os indices dos
        pontos da juncao).

```


Parametros

vertices : array-like (N,3)

Lista/array de pontos (x,y,z).

R : float

Raio do cilindro (padrao 0.5).

H : float

Comprimento total no eixo x (padrao 2.0).

tol : float

Tolerancia para decidir se um ponto esta na lateral/tampa.

singular_value : float

Valor atribuido aos pontos de juncao (por padrao np.nan).

Retorna

curv : ndarray (N,)

Curvaturas medias por ponto (float). Juncoes recebem

'singular_value'.

junc_indices : ndarray (k,)

indices inteiros dos pontos que sao da juncao lateral_tampa.

"""

```
v = np.asarray(vertices)
```

```
if v.ndim != 2 or v.shape[1] != 3:
```

```
    raise ValueError("vertices deve ser shape (N,3)")
```

```
x = v[:, 0]
```

```
y = v[:, 1]
```

```
z = v[:, 2]
```

```
r_yz = np.sqrt(y**2 + z**2)
```

```
is_lateral = np.isclose(r_yz, R, atol=tol)
```

```
is_tampa = np.isclose(np.abs(x), H/2.0, atol=tol)
```

```

is_junc = is_lateral & is_tampa

N = v.shape[0]
curv = np.empty(N, dtype=float)

# juncao
curv[is_junc] = singular_value

# lateral sem ser juncao
lateral_only = is_lateral & ~is_junc
curv[lateral_only] = 1.0 / (2.0 * R)

# tampa sem ser juncao
tampa_only = is_tampa & ~is_junc
curv[tampa_only] = 0.0

return curv

```

Apêndice B

Dados Coletados

B.1 ESFERA

B.1.1 Malhas - Info

Malha	Nos	Elementos	h	Tempo_LB(s)	Tempo_AQFC1(s)	Tempo_AQFC2(s)	Tempo_AQFC3(s)
esfera_007	25597	51190	0,07	24,1	17,0	25,5	38,5
esfera_012	8815	17626	0,12	8,5	6,0	8,8	13,7
esfera_025	2116	4228	0,25	2,0	1,4	2,1	3,1
esfera_035	1058	2112	0,35	1,0	0,7	1,1	1,6
esfera_050	562	1120	0,5	0,5	0,4	0,6	0,8

Tabela B.1: Dados das Malhas da Esfera

B.1.2 Erro Médio - Norma L2

Malha	LB	AQFC1	AQFC2	AQFC3
esfera_007	0,6402%	0,9867%	0,2272%	0,0794%
esfera_012	0,4430%	1,1585%	0,2020%	0,0544%
esfera_025	0,0595%	1,2375%	0,0987%	0,0205%
esfera_035	0,0724%	1,4237%	0,0632%	0,0110%
esfera_050	2,3417%	1,6261%	0,0599%	0,0106%

Tabela B.2: Erro Médio - Norma L2, Esfera

B.1.3 Erro Relativo Médio

Malha	LB	AQFC1	AQFC2	AQFC3
esfera_007	0,006%	0,566%	0,155%	0,058%
esfera_012	0,008%	0,718%	0,141%	0,040%
esfera_025	0,006%	0,798%	0,073%	0,016%
esfera_035	0,008%	0,944%	0,048%	0,009%
esfera_050	0,154%	0,992%	0,034%	0,006%

Tabela B.3: Erro Relativo Médio, Esfera

B.1.4 Erro Relativo Máximo

Malha	LB	AQFC1	AQFC2	AQFC3
esfera_007	101,55%	8,85%	1,52%	0,54%
esfera_012	34,50%	10,58%	2,67%	0,47%
esfera_025	1,30%	9,33%	0,70%	0,12%
esfera_035	1,28%	7,54%	0,29%	0,05%
esfera_050	42,16%	10,79%	0,85%	0,07%

Tabela B.4: Máximo Erro Relativo, Esfera

B.2 ELIPSOIDE 1

B.2.1 Malhas - Info

Malha	Nos	Elementos	h	Tempo_LB(s)	Tempo_AQFC1(s)	Tempo_AQFC2(s)	Tempo_AQFC3(s)
elipsoide_004	27159	54314	0,04	25,9	18,6	30,9	46,2
elipsoide_007	8898	17792	0,07	9,3	6,6	9,9	14,5
elipsoide_012	3119	6234	0,12	3,3	2,3	3,5	5,3
elipsoide_020	1130	2256	0,2	1,1	1,0	1,2	1,8
elipsoide_030	594	1184	0,3	0,6	0,6	0,7	1,3

Tabela B.5: Dados das Malhas do Elipsoide 1

B.2.2 Erro Médio - Norma L2

Malha	LB	AQFC1	AQFC2	AQFC3
elipsoide_004	3,6215%	1,1490%	0,2160%	0,3614%
elipsoide_007	10,2457%	1,5597%	0,5521%	1,1631%
elipsoide_012	3,5517%	3,4642%	1,7114%	3,1975%
elipsoide_020	13,9666%	5,6722%	4,2899%	6,6499%
elipsoide_030	2,9109%	6,7249%	6,6566%	10,0665%

Tabela B.6: Erro Médio - Norma L2, Elipsoide 1

B.2.3 Erro Relativo Médio

Malha	LB	AQFC1	AQFC2	AQFC3
elipsoide_004	0,704%	0,626%	0,064%	0,082%
elipsoide_007	1,114%	0,885%	0,119%	0,187%
elipsoide_012	1,645%	1,934%	0,284%	0,529%
elipsoide_020	3,284%	3,551%	0,787%	1,345%
elipsoide_030	2,343%	3,874%	1,565%	2,400%

Tabela B.7: Erro Relativo Médio, Elipsoide 1

B.2.4 Erro Relativo Máximo

Malha	LB	AQFC1	AQFC2	AQFC3
elipsoide_004	513,43%	12,40%	1,27%	1,55%
elipsoide_007	1314,94%	13,56%	2,25%	4,20%
elipsoide_012	148,06%	11,13%	5,74%	9,62%
elipsoide_020	621,91%	16,13%	12,70%	18,06%
elipsoide_030	19,12%	17,58%	18,73%	26,23%

Tabela B.8: Máximo Erro Relativo, Elipsoide 1

B.3 ELIPSOIDE 2

B.3.1 Malhas - Info

Malha	Nos	Elementos	h	Tempo_LB(s)	Tempo_AQFC1(s)	Tempo_AQFC2(s)	Tempo_AQFC3(s)
newelip_017	21990	43976	0,17	22,9	15,8	25,1	38,2
newelip_025	10174	20344	0,25	9,5	7,0	10,5	15,6
newelip_045	3214	6424	0,45	3,0	2,3	3,4	5,1
newelip_065	1550	3096	0,65	1,5	1,3	1,6	2,4
newelip_090	866	1728	0,9	0,8	0,8	0,9	1,3

Tabela B.9: Dados das Malhas do Elipsoide 2

B.3.2 Erro Médio - Norma L2

Malha	LB	AQFC1	AQFC2	AQFC3
newelip_017	0,8359%	1,4910%	0,2945%	0,4564%
newelip_025	0,9017%	1,6020%	0,4812%	0,9836%
newelip_045	1,3814%	3,4503%	1,6660%	3,1515%
newelip_065	1,2430%	5,7473%	3,2498%	5,4128%
newelip_090	1,5892%	6,9166%	5,0736%	7,6261%

Tabela B.10: Erro Médio - Norma L2, Elipsoide 2

B.3.3 Erro Relativo Médio

Malha	LB	AQFC1	AQFC2	AQFC3
newelip_017	0,499%	0,770%	0,080%	0,095%
newelip_025	0,564%	0,746%	0,099%	0,152%
newelip_045	0,958%	1,605%	0,274%	0,507%
newelip_065	0,914%	3,332%	0,565%	0,984%
newelip_090	1,140%	4,256%	0,956%	1,684%

Tabela B.11: Erro Relativo Médio, Elipsoide 2

B.3.4 Erro Relativo Máximo

Malha	LB	AQFC1	AQFC2	AQFC3
newelip_017	7,75%	11,23%	1,72%	1,95%
newelip_025	7,25%	6,50%	2,07%	3,79%
newelip_045	5,96%	10,96%	5,40%	8,96%
newelip_065	4,92%	15,45%	9,80%	14,63%
newelip_090	5,05%	17,51%	15,88%	21,17%

Tabela B.12: Máximo Erro Relativo, Elipsoide 2

B.4 TORO

B.4.1 Malhas - Info

Malha	Nos	Elementos	h	Tempo_LB(s)	Tempo_AQFC1(s)	Tempo_AQFC2(s)	Tempo_AQFC3(s)	Perc_Zero
toro_007	23457	46914	0,07	22,8	16,0	24,2	37,5	0,17%
toro_012	8004	16008	0,12	7,8	5,4	8,0	12,2	0,12%
toro_020	2925	5850	0,2	2,8	2,1	3,2	4,7	0,07%
toro_028	1532	3064	0,28	1,4	1,1	1,7	2,5	0,20%
toro_045	605	1210	0,45	0,6	0,5	0,7	1,1	0,50%

Tabela B.13: Dados das Malhas do Toro

B.4.2 Erro Médio - Norma L2

Malha	LB	AQFC1	AQFC2	AQFC3
toro_007	3,2820%	1,5804%	0,7274%	0,9441%
toro_012	3,1645%	1,8530%	1,4899%	2,4900%
toro_020	3,7473%	3,2948%	4,0198%	8,8953%
toro_028	3,4329%	5,4652%	8,2697%	18,5824%
toro_045	5,1764%	10,6823%	23,7144%	42,6929%

Tabela B.14: Erro Médio - Norma L2, Toro

B.4.3 Erro Relativo Médio

Malha	LB	AQFC1	AQFC2	AQFC3
toro_007	1,898%	1,367%	0,768%	0,926%
toro_012	2,504%	1,986%	1,836%	2,154%
toro_020	3,468%	3,961%	3,521%	5,356%
toro_028	4,973%	9,292%	8,046%	14,074%
toro_045	4,855%	12,403%	15,554%	34,342%

Tabela B.15: Erro Relativo Médio, Toro

B.4.4 Erro Relativo Máximo

Malha	LB	AQFC1	AQFC2	AQFC3
toro_007	493,00%	399,51%	282,35%	334,39%
toro_012	581,80%	499,53%	630,16%	637,98%
toro_020	844,49%	424,89%	379,15%	568,72%
toro_028	841,62%	1656,98%	863,34%	1011,34%
toro_045	408,14%	617,50%	633,10%	216,07%

Tabela B.16: Máximo Erro Relativo, Toro

B.5 CILINDRO

B.5.1 Malhas - Info

Malha	Nos	Elementos	h	Tempo_LB(s)	Tempo_AQFC1(s)	Tempo_AQFC2(s)	Tempo_AQFC3(s)
cilindro_008	24062	48120	0,08	21,4	17,1	24,9	37,2
cilindro_012	10677	21350	0,12	9,6	7,6	11,2	16,6
cilindro_020	3923	7842	0,2	3,6	2,8	4,0	6,2
cilindro_030	1768	3532	0,3	1,6	1,3	1,9	2,9
cilindro_050	665	1326	0,5	0,6	0,6	0,8	1,1

Tabela B.17: Dados das Malhas do Cilindro

Malha	Perc_Zero		PercNaN		
	TODOS	LB	LB1/AQFC1	LB2/AQFC2	LB3/AQFC3
cilindro_008	19,41%	1,34%	4,06%	6,79%	9,46%
cilindro_012	19,20%	2,00%	6,11%	10,17%	14,11%
cilindro_020	18,81%	3,31%	9,99%	16,87%	23,48%
cilindro_030	18,44%	4,86%	14,71%	23,81%	32,18%
cilindro_050	18,05%	7,82%	23,76%	38,20%	50,83%

Tabela B.18: Percentual de Zeros e Pontos Ignorados (NaNs) por Método nas Métricas Globais

B.5.2 Erro Médio - Norma L2

Malha	LB	AQFC1	AQFC2	AQFC3	LB1	LB2	LB3
cilindro_008	0,6071%	0,4172%	0,0711%	0,0206%	0,5922%	0,5714%	0,5619%
cilindro_012	0,7712%	0,5716%	0,0652%	0,0159%	0,7633%	0,7247%	0,7122%
cilindro_020	1,2074%	0,5009%	0,0347%	0,0074%	1,1484%	0,8593%	0,4457%
cilindro_030	1,1293%	0,7806%	0,0262%	0,0045%	1,0923%	1,0388%	1,0094%
cilindro_050	1,9155%	0,9748%	0,0155%	0,0044%	1,8487%	1,5278%	1,2122%

Tabela B.19: Erro Médio - Norma L2, Cilindro

B.5.3 Erro Relativo Médio

Malha	LB	AQFC1	AQFC2	AQFC3	LB1	LB2	LB3
cilindro_008	0,144%	0,116%	0,024%	0,008%	0,136%	0,125%	0,118%
cilindro_012	0,227%	0,188%	0,026%	0,007%	0,220%	0,205%	0,195%
cilindro_020	0,455%	0,207%	0,015%	0,003%	0,405%	0,268%	0,137%
cilindro_030	0,468%	0,335%	0,012%	0,002%	0,448%	0,416%	0,387%
cilindro_050	1,047%	0,512%	0,008%	0,002%	0,988%	0,719%	0,446%

Tabela B.20: Erro Relativo Médio, Cilindro

B.5.4 Erro Relativo Máximo

Malha	LB	AQFC1	AQFC2	AQFC3	LB1	LB2	LB3
cilindro_008	8,5082%	6,1312%	1,0221%	0,2562%	8,5082%	8,5082%	8,5082%
cilindro_012	10,3204%	10,6839%	0,7270%	0,1971%	9,2656%	8,5653%	8,5653%
cilindro_020	9,2488%	7,3126%	0,2883%	0,0615%	9,2488%	9,2488%	9,2488%
cilindro_030	7,6011%	6,2748%	0,1893%	0,0347%	7,6011%	6,6183%	6,6183%
cilindro_050	8,5166%	5,9820%	0,1207%	0,0259%	8,5166%	8,5166%	8,5166%

Tabela B.21: Máximo Erro Relativo, Cilindro