

Universidade Federal do Rio Grande do Sul
Escola de Engenharia
Programa de Pós-Graduação em Engenharia Civil

Otimização de forma de cascas via deformação livre de
forma baseado em NURBS

Luis Felipe da Rosa Espath

Porto Alegre

2009

Luis Felipe da Rosa Espath

Otimização de forma de cascas via deformação livre de forma baseado em NURBS

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Civil da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para obtenção do título de Mestre em Engenharia. Orientação: Prof. Dr. Armando Miguel Awruch. Co-orientação: Prof. Dr. Inácio Benvegnu Morsch.

Porto Alegre

2009

E77o Espath, Luis Felipe da Rosa
Otimização de forma de cascas via deformação livre de forma baseado em
NURBS / Luis Felipe da Rosa Espath. – 2009.

Dissertação (Mestrado) - Universidade Federal do Rio Grande do Sul.
Escola de Engenharia. Programa de Pós-Graduação em Engenharia Civil.
Porto Alegre, BR-RS, 2009

Orientador: Prof. Dr. Armando Miguel Awruch.

Co-orientador: Prof. Dr. Inácio Benvegnu Morsch.

1. Elementos finitos. 2. Otimização matemática. 3. Estruturas
(Engenharia). I. Awruch, Armando Miguel, orient. II. Morsch, Inácio
Benvegnu, coorient. III. Título.

CDU-624.04(043)

Luis Felipe da Rosa Espath

Otimização de forma de cascas via deformação livre de forma
baseado em NURBS

Esta dissertação de mestrado foi julgada adequada para a obtenção do título de MESTRE EM ENGENHARIA, na Área de Concentração de ESTRUTURAS, e aprovada em sua forma final pelo professor orientador e pelo Programa de Pós-Graduação em Engenharia Civil da Universidade Federal do Rio Grande do Sul.

Porto Alegre, 10 de Agosto de 2009

Prof. Armando Miguel Awruch
D.Sc. COPPE/UFRJ
Orientador

Prof. Inácio Benvegnu Morsch
Dr. PPGEC/UFRGS
Co-orientador

Ph.D. Luis Carlos Pinto da Silva Filho
Coordenador do PPGEC/UFRGS

Comissão Julgadora:

Dr. Maria Angela Vaz dos Santos
Dr. PPGEC/UFRGS

Dr. Samir Maghous
Dr. Ecole Nationale Des Ponts et Chaussees

Dr. Herbert Martins Gomes
Dr. PPGEC/UFRGS

Dedico a Luca

Agradecimentos

Ao Prof. Armando M. Awruch, pela orientação, dedicação e incentivo durante a realização deste trabalho.

À minha GRANDE família, Luca y Gise. Aos meus Pais e meu Irmão.

À agradável amizade que formei com os colegas do PPGEC. Em particular, ao amigo Sergio Cardozo, pelo companheirismo, por ser um grande colega de estudos de *FEM*, ForTran, \LaTeX entre outros. E ao amigo Felipe Almeida, um professor.

Aos professores do Programa de Pós-Graduação em Engenharia Civil – PPGEC/UFRGS, pelos ensinamentos transmitidos.

À Coordenação de Aperfeiçoamento de Pessoal de Ensino Superior – CAPES, pela bolsa de estudos.

Epígrafe

O que é bonito?
É o que persegue o infinito;
Mas eu não sou
Eu não sou, não...
Eu gosto é do inacabado,
O imperfeito, o estragado, o que dançou
O que dançou...
Eu quero mais erosão
Menos granito.
Namorar o zero e o não,
Escrever tudo o que desprezo
E desprezar tudo o que acredito.
Eu não quero a gravação, não,
Eu quero o grito.
Que a gente vai, a gente vai
E fica a obra,
Mas eu persigo o que falta
Não o que sobra.
Eu quero tudo que dá e passa.
Quero tudo que se despe,
Se despede, e despedaça.
O que é bonito...

Resumo

Espath, L.F.R. **Otimização de forma de cascas via deformação livre de forma baseado em NURBS**. Dissertação (Mestrado em Engenharia Civil) – Programa de Pós-Graduação em Engenharia Civil, UFRGS, Porto Alegre, 2009.

Neste trabalho buscou-se consolidar a união entre três áreas do conhecimento: a parametrização de curvas e superfícies do tipo *B-spline* racionais não-uniformes (NURBS), a otimização matemática e a análise estrutural por elementos finitos. A união destas três áreas é realizada neste trabalho através da otimização de formas de cascas, devido ao fato de que as características mecânicas dos materiais devem refletir-se na forma da estrutura e sua distribuição de espessura expressando um máximo desempenho. Estas variáveis, forma e distribuição de espessura, possuem um rol dominante nos projetos de engenharia, já que mínimas quantidades de materiais, uma frequência específica, um estado puro de tensões de membrana são típicos objetivos de projeto. Neste contexto, obter a forma e a distribuição de espessura adequadas são conceitos intrínsecos à otimização estrutural. Portanto, implementaram-se técnicas para modificar a geometria de cascas, sem perder a parametrização, sem a necessidade de gerar uma nova malha de elementos finitos ao se modificar a forma e ainda ter controle sobre a distorção da malha para evitar erros numéricos inaceitáveis. A modificação de forma é fomentada pelo código de otimização, programação quadrática sequencial (SQP), motivado pelas análises da casca por elementos finitos. A modificação de forma é realizada pela técnica de deformação livre de forma (*free-form deformation*) com a parametrização NURBS. Nos resultados da otimização de formas de cascas obtiveram-se cascas com alto desempenho estrutural e esteticamente agradáveis.

Palavras-chave: NURBS, Deformação Livre de Forma, Otimização Matemática, Método dos Elementos Finitos.

Abstract

Espath, L.F.R. **Shape optimization of shell via free-form deformation NURBS-based**. Dissertação (Mestrado em Engenharia Civil) – Programa de Pós-Graduação em Engenharia Civil, UFRGS, Porto Alegre, 2009.

Consolidation of the link among three fields, curves and surfaces described by non-uniform rational B-spline (NURBS), mathematical optimization and finite element structural analysis, applied to shape optimization of shells, is the main objective of this work. Shape optimization of shells are performed taking into account the fact that the material mechanical characteristics influence the structural shape and the thickness variation in order to obtain the best performance. These two variables, shape and thickness variation, have an essential role considering that the minimum material quantities, a specific frequency and a pure membrane stress state are typical design objectives. Suitable shapes and thickness variation are intrinsic concepts of structural optimization. Therefore, some techniques were implemented to modify the shell geometry conserving the same parameterization without a new finite element mesh generation and controlling mesh distortion in order to avoid relevant numerical errors. The shape modification is conducted by the optimization code and it is based in the data obtained by finite element analysis. In this work the optimization procedure is performed using a Sequential Quadratic Programming (SQP) algorithm, while the shape modification is carried out by the free-form deformation technique, based on NURBS parameterization. As a consequence of the shape optimization, shells with high structural performance and esthetically beautiful were obtained.

Keywords: NURBS, Free-Form Deformation, Mathematical Optimization, Finite Element Method.

Sumário

Lista de Figuras	10
Lista de Tabelas	12
1 Introdução	13
1.1 Motivação	13
1.2 História	14
1.3 Objetivo e justificativa	18
1.4 Bibliografia pesquisada	19
1.5 Descrição da dissertação	20
2 Geometrias parametrizadas via NURBS	22
2.1 Introdução	22
2.2 Funções paramétricas	22
2.3 Bézier	23
2.4 Coordenadas homogêneas	29
2.5 Bézier racional	31
2.6 Superfície: produto tensorial	34
2.7 Funções de base <i>B-spline</i>	37
2.7.1 Definição e propriedades das funções de base <i>B-spline</i>	40
2.7.2 Derivadas das funções de base <i>B-spline</i>	45
2.7.3 Mais propriedades das funções de base	46
2.8 Curvas e superfícies <i>B-spline</i>	48
2.8.1 Definição e propriedades das curvas <i>B-spline</i>	49
2.8.2 Derivadas das curvas <i>B-spline</i>	51
2.8.3 Definição e propriedades das superfícies <i>B-spline</i>	52
2.8.4 Derivadas das superfícies <i>B-spline</i>	54
2.9 <i>B-spline</i> racionais não-uniformes – NURBS –	55
2.9.1 Definição e propriedades das curvas NURBS	56
2.9.2 Derivadas das curvas NURBS	60
2.9.3 Definição e propriedades das superfícies NURBS	61
2.9.4 Derivadas das superfícies NURBS	67
2.10 Curvatura Gaussiana	68
2.10.1 Primeira forma fundamental	68
2.10.2 Segunda forma fundamental	68
2.10.3 Computo da curvatura Gaussiana	69
2.11 Modificação de forma	69
2.11.1 Algoritmo de modificação de forma – deformação livre de forma –	72
2.12 Exemplos de deformação livre de forma	74

3	Otimização	81
3.1	Introdução	81
3.2	Teoria de otimização sem restrições	81
3.2.1	Mínimo local	82
3.2.2	Condições do ponto de ótimo sem restrições	82
3.2.3	Problemas não suaves	84
3.2.4	Estratégias: busca linear e região confiável	85
3.2.5	Fator de escala	91
3.2.6	Razão de convergência	91
3.2.7	Razão de convergência R	92
3.3	Teoria de otimização com restrições	93
3.3.1	Condições do ponto de ótimo de primeira ordem	94
3.3.2	Derivação das condições de primeira ordem	95
3.3.3	Caracterizando as direções limitantes: qualificação das restrições	97
3.3.4	Multiplicadores de Lagrange	101
3.3.5	Demonstração do Teorema 3.6	103
3.3.6	Condições do ponto de ótimo de segunda ordem	104
3.3.7	Problemas não suaves	108
3.3.8	Sensibilidade	108
3.4	Programação quadrática	110
3.5	Programação quadrática sequencial – SQP –	111
3.6	Algoritmo de otimização – SQP –	112
4	Otimização de forma de cascas via deformação livre de forma baseado em NURBS	114
4.1	Introdução	114
4.2	Definição do problema de otimização estrutural	116
4.3	Algoritmo de otimização de forma de cascas via deformação livre de forma baseado em NURBS	118
4.4	Exemplos de otimização de forma de cascas via deformação livre de forma baseado em NURBS	120
4.4.1	Exemplo 1	121
4.4.2	Exemplo 2	122
4.4.3	Exemplo 3	123
4.4.4	Exemplo 4	124
4.4.5	Exemplo 5 – MIT –	124
4.4.6	Últimos exemplos	129
5	Conclusões Finais	138
	Referências Bibliográficas	141
A	Análise de cascas por elementos finitos	143
A.1	Formulação do Elemento Finito	143
A.1.1	Matriz de Rigidez	144
A.1.2	Análise geométrica não-linear	148
A.1.3	Matriz de massa consistente	150

Lista de Figuras

2.1	Curva Bézier de grau dois	24
2.2	Curvas Bézier de grau três	24
2.3	Curvas Bézier de grau quatro	25
2.4	Curva Bézier de grau cinco	25
2.5	Polinômios de Bernstein de grau dois	26
2.6	Polinômios de Bernstein de grau nove	26
2.7	Ponto sobre curva quadrática por repetidas interpolações lineares em u_0	28
2.8	Ponto sobre curva cúbica por repetidas interpolações lineares em u_0	29
2.9	Transformação de um ponto do espaço homogêneo ao espaço Euclidiano	30
2.10	Transformação de uma curva do espaço homogêneo ao espaço Euclidiano	31
2.11	Construção de um arco de circunferência	33
2.12	Mapeamento do arco do espaço homogêneo ao espaço Euclidiano	34
2.13	Três polinômios por partes de grau três	37
2.14	Curva da figura 2.13 na forma Bézier	38
2.15	Funções de base não-nulas de zero grau	42
2.16	Funções de base não-nulas de primeiro grau	42
2.17	Funções de base não-nulas de segundo grau	42
2.18	Funções de base B- <i>spline</i> de grau três	46
2.19	Derivadas das funções de base B- <i>spline</i> da figura 2.18	46
2.20	Funções de base NURBS, $p = 3$, para os $w_{(a)i}$	57
2.21	Funções de base NURBS, $p = 3$, para os $w_{(b)i}$	57
2.22	Funções de base NURBS, $p = 3$, para os $w_{(c)i}$	58
2.23	Curvas NURBS com diferentes pesos	59
2.24	Curvas NURBS sobrepostas com diferentes pesos	59
2.25	Derivadas das funções de base NURBSvda figura 2.20	61
2.26	Derivadas das funções de base NURBSvda figura 2.21	62
2.27	Derivadas das funções de base NURBSvda figura 2.22	62
2.28	Superfícies NURBS com a rede de controle (1)	63
2.29	Superfícies NURBS com a rede de controle (2)	63
2.30	Superfícies NURBS com a rede de controle (3)	64
2.31	Superfícies NURBS com a rede de controle (4)	64
2.32	Superfícies NURBS com a rede de controle (5)	64
2.33	Relação entre os conjuntos	66
2.34	Subdomínios à serem reavaliados	73
2.35	Superfície NURBS (1)	75
2.36	Superfície NURBS deformada (1)	75
2.37	Superfície NURBS (2)	75
2.38	Superfície NURBS deformada (2)	76

2.39	Superfície NURBS (3)	76
2.40	Superfície NURBS deformada (3)	76
2.41	Superfície NURBS (4)	77
2.42	Superfície NURBS deformada (4)	77
2.43	Superfície NURBS deformada 2 (4)	77
2.44	Superfície NURBS (5)	78
2.45	Superfície NURBS deformada (5)	78
2.46	Superfície NURBS (6)	79
2.47	Superfície NURBS deformada (6)	80
3.1	Fator de escala	91
4.1	Esquema do algoritmo	119
4.2	Exemplo 1	121
4.3	Geometria final 1	122
4.4	Geometria final 2	122
4.5	Exemplo 3	123
4.6	Geometria final 3	124
4.7	Geometria final 4	124
4.8	Auditório Kresge MIT	125
4.9	Malhas iniciais	126
4.10	Geometria e pontos de controle – MIT –	127
4.11	Sequência de iterações e geometria final (sem viga)	127
4.12	Sequência de iterações e geometria final (com viga)	128
4.13	Geometria otimizada obtida por Ramm et al. (1993)	129
4.14	Geometria e pontos de controle – Placa –	129
4.15	Deslocamentos da geometria inicial (6)	130
4.16	Sequência de iterações e geometria final (6)	131
4.17	Deslocamentos da geometria final 1 (6)	131
4.18	Deslocamentos da geometria final 2 (6)	131
4.19	Deslocamentos da geometria inicial (7)	132
4.20	Sequência de iterações e geometria final (7)	133
4.21	Deslocamentos da geometria final 1 (7)	133
4.22	Deslocamentos da geometria final 2 (7)	133
4.23	Deslocamentos da geometria inicial (8)	134
4.24	Sequência de iterações e geometria final (8)	135
4.25	Deslocamentos da geometria final 1 (8)	135
4.26	Deslocamentos da geometria final 2 (8)	135
4.27	Deslocamentos da geometria inicial (9)	136
4.28	Geometria final, agradável resultado estético (9)	136
4.29	Sequência de iterações e geometria final (9)	137
4.30	Deslocamentos da geometria final 1 (9)	137
4.31	Deslocamentos da geometria final 2 (9)	137
A.1	Graus de liberdade do elemento triangular plano para placa e casca (Almeida, 2006)	144
A.2	Lâminas formando o laminado e ângulo θ entre eixo principal do material 1-2 e sistema x-y do laminado (Almeida, 2006)	146
A.3	Eixos locais do elemento (López, 2009)	150

Lista de Tabelas

2.1	Geração do ponto $\mathbf{C}^w \left(\frac{1}{2}\right)$ em um arco de circunferência	34
2.2	Tabela triangular truncada das $N_{i,p}(u)$	41
2.3	Suporte local	43
2.4	Funções não-nulas sobre $u \in [u_3, u_4)$	43
4.1	Deslocamentos da geometria original	126
4.2	Deslocamentos da geometria otimizada	128
4.3	Espessuras da geometria otimizada	128

Capítulo 1

Introdução

1.1 Motivação

Esta dissertação foi motivada exclusivamente pela necessidade de implementar computacionalmente um método de modificação de forma, sem que este estivesse atrelado ou acoplado a um programa de modelagem computacional e muito menos a um gerador de malha e a um elemento finito em particular. O motivo pelo qual esta implementação deveria independer do programa de modelagem era para que o usuário faça uso do programa de modelagem de sua preferência, assim como esta deveria independer do gerador de malha, para que não se perca a eficiência tendo que remalhar a geometria a cada modificação de forma.

Ainda, por motivos de versatilidade, que ao longo desta monografia serão explanados, foi escolhido *B-spline* racionais não-uniformes para a parametrização da geometria. Este tipo de parametrização, junto com *T-spline*, talvez sejam os mais versáteis, no que concerne ao poder de representar geometrias altamente complexas.

Para apresentar o potencial da implementação do método de modificação de forma, o mesmo foi acoplado ao problema de otimização com elementos finitos de casca, resultando então na otimização de formas de cascas. Um tema que ao longo do trabalho, se mostrou bastante atual, já que na pesquisa bibliográfica encontraram-se trabalhos bem sucedidos de otimização de forma; este fato deve-se em grande parte à parametrização ser baseada em *B-spline* racionais não-uniformes.

1.2 História

História da spline

Há um tempo, antes dos computadores surgirem, arquitetos, engenheiros, e artistas desenhavam seus projetos para edifícios, estradas, partes de máquinas e semelhantes usando lápis, papel e várias ferramentas de desenho. Estas ferramentas incluíam régua e T-quadrado para as retas de desenho reto, compassos para desenho de círculos, arcos circulares, triângulos e transferidores para fazer ângulos precisos. Obviamente, muitos objetos de formato interessante não podiam ser desenhados somente com estas ferramentas simples, pois eles tinham partes curvas que não eram círculos ou elipses. Frequentemente, uma curva precisava que fosse suave em vários pontos pré-determinados. Este problema era particularmente crítico na construção naval: embora um artista ou desenhista qualificado pudesse confiantemente desenhar a mão tais curvas em tabela de desenhos. Construções de navios frequentemente precisavam ser desenhadas de tamanho natural (ou quase do tamanho natural), no qual o tamanho empinado das curvas exigido para ser desenhado a mão era impossível. Por causa de seu grande tamanho, tais desenhos eram frequentemente feitos na área de sótão de um grande edifício, por um especialista conhecido como “homem do sótão”. Para ajudar na tarefa, o “homem do sótão” empregava tiras longas, magras, flexíveis de madeira, plástico, ou metal, chamadas spline. A spline era segura no lugar com pesos principais. As curvas resultantes eram lisas, e variadas em curvatura que dependiam da posição dos pesos. Como computadores foram introduzidos no processo de projeto, as propriedades físicas de tal spline foram investigadas de forma que elas pudessem ser matematicamente modeladas no computador (*deBoor, 1978*).

História da otimização

Está na Eneida de Virgílio. Dido, uma fenícia, persuadiu um chefe africano a lhe dar tanta terra quanto ela pudesse cercar com a tripa de um touro. Assim foi. Primeiro, ela cortou as tripas em centenas de tiras bem fininhas. Depois, espertamente, as uniu para traçar um semicírculo no chão, a beira do mar Mediterrâneo. Era a máxima área costeira que ela poderia envolver. Neste lugar ela construiu uma cidade. A famosa Cartago.

Antes de se tornar rainha de Cartago, Dido teria resolvido o primeiro problema de

otimização da história. Mesmo sendo literário, o relato demonstraria que os povos da antiguidade possuíam conhecimento a respeito de áreas e comprimentos. Sabiam que, dentre as figuras de igual perímetro, o círculo é aquela com maior área.

É fato. Sabiam, com toda certeza. Contudo, a atitude da rainha Dido não demonstra exatamente isso. Trata-se de um mito. Tem mais a ver com a simbologia feminina associada à cidade de Cartago. A figura que Dido traça no chão não seria, necessariamente, um sinal de otimização da área. Um semicírculo é capaz de representar um útero engravidado. Esboçando então um modesto entender: a hipótese a ser explorada poderia ser outra. A de que a forma do terreno, escolhida a partir da tripa de um touro, a beiramar, por uma mulher, para gerar uma cidade, simboliza um máximo. Porém, no caso, um máximo de adequação ao arquétipo da mãe-terra. Muito difundido, por sinal, naquelas paragens.

Não obstante, a ilustração vale e nos leva a ponderar sobre a dinâmica do conhecimento humano. Compartilhamos com as civilizações da antiguidade vários itens do conhecimento científico. Em particular, há verdades matemáticas que já eram sabidas antigamente. Mas, não do mesmo modo que as sabemos hoje. Diferentes culturas embrulham um mesmo conhecimento de formas diferentes.

Costuma-se admitir que a matemática com o aspecto que ela apresenta hoje, surgiu lá pelo século XVII. Em se tratando de datas é uma boa escolha. Coincide com o nascimento da ciência moderna. O século XVII é um século pródigo, onde nasceram e viveram várias figurinhas carimbadas. Leibniz, Newton, Huygens, os dois primeiros Bernoulli, Descartes, Mersennes, Pascal, Torricelli, Cavalieri, Desargues, Halley e muitos outros.

Copérnico nasceu no final do XV viveu até meados do seguinte. Giordano Bruno e Ticho Brahe viveram na segunda metade do XVI. Galileu e Kepler nasceram em meados do XVI e adentraram o XVII. Os cinco representam os prelúdios da modernidade.

Do outro lado, já no século XVIII, podemos escalar Euler, Lagrange e, para terminar, Cauchy e Gauss. Há vários outros personagens igualmente significativos, como MacLaurin, que tanto impressionou Lagrange.

O que interessa aqui é debulhar um pouco a história do cálculo matemático de máximos

e mínimos. Certo, tem a ver com as noções de derivada e integral. Mas não como se conhecem hoje. No século XVII estes conceitos estavam atrelados ao traçado de tangentes e à avaliação de áreas. Como era de se esperar, o cálculo, criado por Leibniz e Newton, não se afastou da tradição geométrica consagrada, por exemplo, em Arquimedes.

Daí a importância de Euler e Lagrange. Ambos trabalharam no sentido de “desgeometrizá-la” a matemática criada pelos seus antecessores. Entre inúmeros outros feitos, Euler é o responsável pelo conceito de função matemática. Lagrange, com o seu “*Mécanique Analytique*”, reconstruiu a mecânica newtoniana em bases analíticas, desvincilhada de figuras geométricas. Sobretudo, um e depois o outro plantaram e fortaleceram as raízes do cálculo variacional.

Finalmente, Cauchy. Hoje em dia, estamos acostumados a definir integrais e diferenciais em termos de “limites”. Mas não era bem assim antes de Cauchy. Foi ele quem botou ordem. Livrou o cálculo dos “infinitésimos” introduzindo os conceitos de limite e de continuidade. Deu fundamentos às idéias de derivada e integral. Inaugurou a análise matemática. Com a forma, a autonomia e o rigor atuais.

Para entender melhor isto, volta-se ao século XVII. Um personagem emblemático desta época era Fermat. Pierre de Fermat não era um matemático profissional. De temperamento tímido, publicou apenas um artigo em vida. Dos seus feitos, há apenas anotações esparsas, cartas e comentários de admiradores. Consta que era basco, tinha uma inclinação por línguas e literatura, estudou leis e se tornou uma espécie de procurador da justiça. Por tédio, vocação ou apenas para se entreter, resolveu recriar, seguindo a moda da época, uma obra perdida de Apolônio. Era o “*Plane Loci*”, sobre lugares geométricos. Talvez por conta disso, se sentiu estimulado a representar curvas por meio de relações algébricas entre as coordenadas. Inventou assim a geometria cartesiana, lá por 1630.

Esta geometria é “cartesiana” em honra a Descartes que a divulgou em 1637. O que é justo pois, ao que tudo indica, não se sabia do trabalho de Fermat até a sua publicação póstuma. Além disso, a notação matemática de Fermat era terrível. Acostumava-se a usar o método de Viète (criador da primeira sistematização de símbolos algébricos, no final do século XVI) mais perto da linguagem corrente do que do simbolismo que temos hoje. Contudo, a geometria “cartesiana” de Fermat é considerada mais genérica e mais

acurada do que a de Descartes.

De posse de representações algébricas para curvas e superfícies, Fermat resolveu se aventurar na resolução de uma outra classe de problemas clássicos. O traçado de tangentes. Em decorrência, se debruçou na investigação de pontos extremos de curvas.

Nesta área, praticamente inexplorada até então, criou um método próprio. Um método de obtenção de máximos e mínimos que funcionava mais ou menos assim:

- (i) de posse da expressão algébrica da curva (digamos, $f(x) = ax^2 + bx + c$) se avalia a mesma levemente deslocada (ou seja, $f(x + \delta)$);
- (ii) se expande a expressão, separando-se os termos no deslocamento (ou seja, $a\delta^2 + 2ax\delta + b\delta$);
- (iii) os deslocamentos são muito pequenos, tomá-se apenas os termos mais relevantes (ou seja, $(2ax + b)\delta$);
- (iv) se investiga quando a expressão resultante é nula para qualquer valor do deslocamento (ou seja, em $x = -\frac{b}{2a}$);
- (v) neste ponto a curva é “quase constante” e a tangente a ela deve ser horizontal. Assim sendo, pode se tratar de um ponto extremo.

Obviamente, desta maneira apenas se conseguem condições necessárias.

Em suma: Fermat usava um procedimento de cálculo algébrico idêntico ao de hoje. Mas o utilizava como uma espécie de algoritmo, sem o amparo dos conceitos de função, derivada ou diferencial. Neste sentido, foi um precursor. Mais de cem anos depois, Laplace reconheceu isto. Creditou a Fermat a invenção do cálculo diferencial.

No século XVII havia uma polêmica a respeito da natureza da luz. Huygens defendia a hipótese ondulatória, Newton acabou se sobrepondo com a corpuscular. Na verdade, a discussão era mais ideológica, pois ambas se equivaliam ao explicarem a reflexão e a refração. No meio destas discussões de fundamentos, Fermat propõe um ponto de partida matemático. A luz teria velocidade finita, dependente do meio circulante. Quais as

trajetórias dos raios luminosos? Aquelas de tempo mínimo. As que a luz levasse o menor tempo de percurso.

Era um “princípio”, e acabou ficando conhecido como princípio de Fermat. Independente da natureza da luz este princípio, sozinho, é capaz de resolver todos os problemas da ótica geométrica. Por outro lado, por ser teológico, criou uma polêmica filosófica. Como que a luz saberia ser o caminho de tempo mínimo antes de tê-lo percorrido?

Leibniz e depois Euler se tornaram ardentes admiradores do princípio de Fermat. Tentaram generalizá-lo, aplicar a mesma teologia a outras situações. Com o desenvolvimento do cálculo variacional isto se tornou uma realidade. Outros princípios, chamados variacionais, surgiram. Estão hoje por toda a física. Condensam fatos, verdades ou leis que surgem quando a integral no tempo de determinadas entidades matemáticas é otimizada.

Circunscreve-se ao que foi exposto a contribuição de Fermat ao cálculo de máximos e mínimos. Mas suas atividades não se esgotam aí. Vão mais longe ainda. Fermat é indiscutivelmente o patrono da teoria dos números. Talvez devido às suas leituras de Diofanto, dedicou grande parte da vida a estabelecer teoremas sobre os inteiros; em particular, os números primos (*Jocelyn Freitas Bennaton – 04.2001*).

1.3 Objetivo e justificativa

O objetivo desta dissertação é essencialmente a implementação de um código computacional que manipula curvas e superfícies, assim como suas derivadas, paramétricas do tipo *B-spline* racionais não-uniformes. Entretanto, buscou-se uma aplicação de engenharia para apresentar o potencial do código desenvolvido. Esta aplicação resultou ser a otimização de formas de cascas, onde, obviamente, a modificação da forma é dada pelo código computacional que manipula tais entidades como curvas, superfícies.

Este se justifica devido ao fato de que a forma, distribuição de espessura e características dos materiais possuem um rol dominante nos projetos de engenharia. Mínimas quantidades de materiais, uma frequência específica, um estado puro de tensões de membrana são típicos objetivos de projeto. Neste contexto, obter a forma, a distribuição

de espessura e o material adequado são conceitos intrínsecos à otimização estrutural. E uma justificativa indireta é que os códigos neste trabalho desenvolvidos podem ser utilizados para a otimização aerodinâmica; um problema mais atual na engenharia.

1.4 Bibliografia pesquisada

Visto que o objetido desta dissertação é a implementação do método de modificação de forma denominado por: deformação livre de forma (*free-form deformation*) baseado em parametrizações do tipo *B-spline* racionais não-uniformes (NURBS), no que concerne à bibliografia, foram pesquisados livros e artigos da computação gráfica de grande relevância científica.

Na década de 60 o Dr. Pierre Bézier, um engenheiro da corporação Renault, criou uma formulação paramétrica para o uso em modelagem geométrica, publicado em 1972, Bézier (1972) baseado no trabalho de Bernstein (1912). deCasteljau, engenheiro contemporâneo a Bézier, da corporação Citroën, desenvolveu algoritmos altamente eficientes para o computo de pontos sobre uma curva Bézier. As *spline* foram estudadas por N. Lobachevsky, cuja importante contribuição para a matemática é talvez a denominada geometria hiperbólica (não-Euclidiana) no final século XVIII. A partir da proposta destes engenheiros outros pesquisadores uniram os conceitos de funções *spline* com a formulação paramétrica de Bézier e os algoritmos de deCasteljau. Este fato deu-se quase que instantaneamente por deBoor (1972) e Cox (1972), desenvolvendo a parametrização do tipo *B-spline*. Esta linha de pesquisa evoluiu rapidamente, Riesenfeld (1981) e Patterson (1985) baseados na excelente capacidade de representação das funções racionais, encontraram em Roberts (1965) uma maneira eficiente de implementar a parametrização do tipo Bézier racional, i.e., por meio de coordenadas homogêneas. A teoria à cerca das parametrizações do tipo *B-spline* racionais não-uniformes foi condensada por Piegl e Tiller (1997), no livro *The NURBS Book*. A pesquisa sobre o tema não parou por aqui, até os dias de hoje continuam ocorrendo inovações. Recentemente foram desenvolvidas as parametrizações do tipo *T-spline* por Sederberg et al. (2003). Estas parametrizações surgem a partir da parametrização NURBS juntamente com os algoritmos de remoção e inserção de nós.

No que concerne à modificação de forma deve-se observar que todas as técnicas

desenvolvidas foram elaboradas para a computação gráfica, devido a este fato, as técnicas de modificação podem não ser adequadas para outros fins. Os tipos de modificação de forma, podem ser discriminadas em dois grandes grupos: baseadas na geometria e baseadas na física. Os métodos baseados na física são os mais recentes e são chamados de D-NURBS (*Dynamic NURBS*); foram propostos por Terzopoulos e Qin (1994), são os mais intuitivos, já que a deformação é imposta por meio de forças aplicadas às superfícies. Os métodos baseados na geometria são os apresentados por Piegl (1989a), Piegl (1989b) e Sederberg e Parry (1986); onde (ambos métodos são matematicamente equivalentes). Sederberg e Parry (1986) apresentaram o método de deformação livre de forma (*Free-form deformation*), provavelmente o mais fácil de implementar e o mais adequado para otimização de forma.

1.5 Descrição da dissertação

No capítulo 2 é apresentado em ordem cronológica, os tipos de parametrizações que surgiram nos últimos quarenta anos. É apresentada a teoria da parametrização que Bézier propôs como ferramenta de projeto. Esta abre uma gama imensa de novas pesquisas, como *B-spline*, uma extensão da proposta de Bézier; apresentada na sequência. Ainda, apresenta-se a teoria elementar da parametrização *B-spline* racional não-uniforme. Na última seção é apresentada a técnica utilizada para modificar superfícies, assim como o algoritmo e alguns resultados preliminares de modificação de forma.

No capítulo 3 é apresentada a teoria de otimização numérica. A teoria de otimização sem restrições e alguns métodos consagrados são apresentados na primeira seção. A teoria de otimização com restrições é analisada detalhadamente e é apresentado o problema de otimização sob o enfoque da programação quadrática sequencial, assim como seu algoritmo em linhas gerais.

No capítulo 4 é apresentado o problema de otimização estrutural e como é realizada a transformação da otimização matemática para a otimização em termos mecânicos. O acoplamento do código computacional desenvolvido com a otimização e o código computacional de elementos finitos de casca é apresentado com a descrição do algoritmo final. E concluindo, neste capítulo são apresentados nove exemplos simples com a intenção

de validar o código e consolidar a idéia de que a otimização de forma não é um problema trivial.

No capítulo 5 constam as conclusões e sugestões para trabalhos futuros.

No apêndice A é apresentada a formulação do elemento de casca utilizado.

Capítulo 2

Geometrias parametrizadas via NURBS

2.1 Introdução

As duas maneiras, de se representar analiticamente curvas, superfícies e sólidos (largamente usadas na modelagem geométrica computacional ou CAGD, do acrônimo em inglês, *Computer Aided Geometry Design*) são funções implícitas ou funções paramétricas. A função implícita de uma curva contida no plano xy toma a forma $f(x,y) = 0$. Esta equação descreve uma relação implícita entre as quantidades x e y . Na forma paramétrica, cada quantidade é representada de forma separada na forma de função explícita em função de um, dois ou três parâmetros independentes, para curvas, superfícies e sólidos, respectivamente.

2.2 Funções paramétricas

A função paramétrica de uma curva no espaço Euclideano toma a seguinte forma

$$\mathbf{C}(u) = (x(u), y(u), z(u)) \quad a \leq u \leq b \quad (2.1)$$

logo, $\mathbf{C}(u)$ é uma função vetorial de uma variável independente, u . Para superfícies são duas variáveis independentes e para sólidos, três variáveis independentes, resultando em

$$\mathbf{S}(u,v) = (x(u,v), y(u,v), z(u,v)) \quad (u,v) \in [a,b] \times [c,d] \quad (2.2)$$

$$\mathbf{G}(u,v,w) = (x(u,v,w), y(u,v,w), z(u,v,w)) \quad (u,v,w) \in [a,b] \times [c,d] \times [e,f] \quad (2.3)$$

Em relação ao campo de variação dos parâmetros, os intervalos $[a,b]$, $[c,d]$ e $[e,f]$ são arbitrários, mas comumente se usa o intervalo normalizado.

Talvez a diferença mais relevante entre as formas implícita e paramétrica é que na implícita existe uma única função que descreve analiticamente certa geometria (curva, superfície ou sólido) já na forma paramétrica existem infinitas parametrizações que descrevem analiticamente a mesma geometria. No restante desta dissertação apenas serão tratadas geometrias na forma paramétrica de curvas e superfícies.

2.3 Bézier

Curvas e superfícies Bézier foram nomeadas após seu inventor, Dr. Pierre Bézier, um engenheiro da corporação Renault na década de 60, desenvolver uma formulação paramétrica para o uso em modelagem geométrica, publicado em 1972, Bézier (1972). A formulação é bastante intuitiva para que projetistas e artistas possam usar sem requerer o conhecimento matemático de geometria analítica e diferencial.

Geometrias do tipo Bézier usam funções polinomiais explícitas para descrever suas coordenadas. Uma curva Bézier de n -ésimo grau toma a seguinte forma

$$\mathbf{C}(u) = \sum_{i=0}^n B_{i,n}(u) \mathbf{P}_i \quad u \in [0,1] \quad (2.4)$$

onde as funções de base (*blending*), $\{B_{i,n}\}$, são os clássicos polinômios de *Bernstein* de n -ésimo grau, divulgados por Bernstein (1912), que vem dados por

$$B_{i,n}(u) = \binom{n}{i} u^i (1-u)^{n-i} \quad (2.5)$$

onde $\binom{n}{i}$ é o número ou coeficiente binomial dado por

$$\binom{n}{i} = C_i^n = \frac{n!}{i!(n-i)!} \quad (2.6)$$

e os coeficientes geométricos, $\{\mathbf{P}_i\}$, são chamados de *pontos de controle*, já que estes definem ou controlam a geometria parametrizada. O polígono formado pelos pontos de controles é chamado de *polígono de controle*. Algumas propriedades importantes são listadas a seguir.

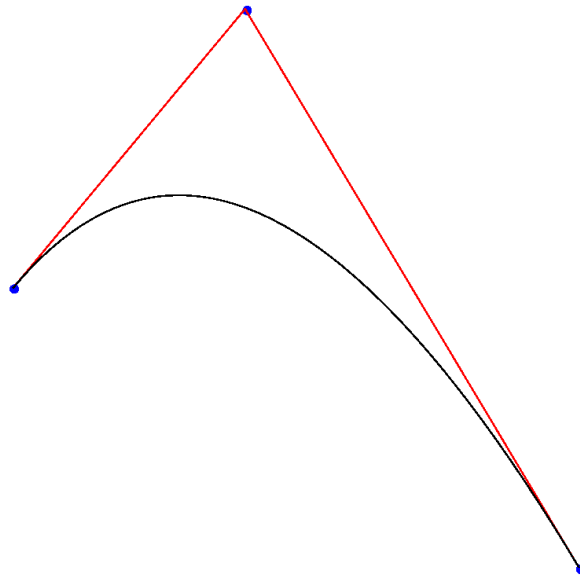
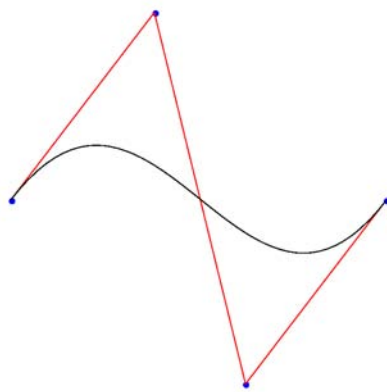
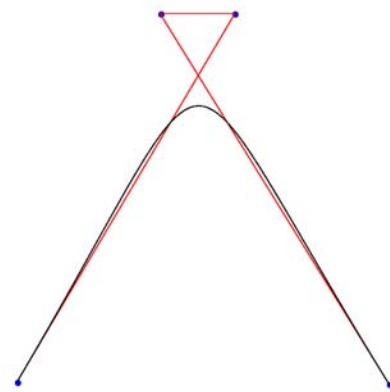


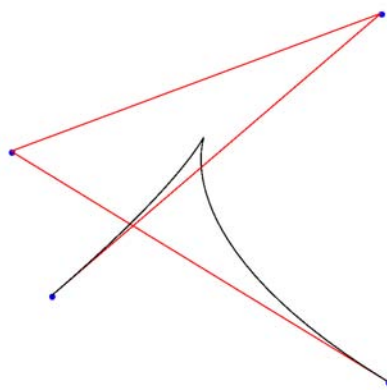
Figura 2.1: Curva Bézier de grau dois.



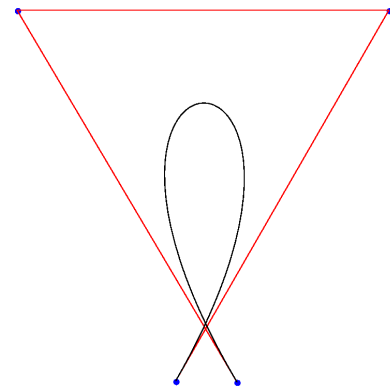
(a) ponto de inflexão



(b) polígono cruzado



(c) descontinuidade visual



(d) curva com uma volta

Figura 2.2: Curvas Bézier de grau três.

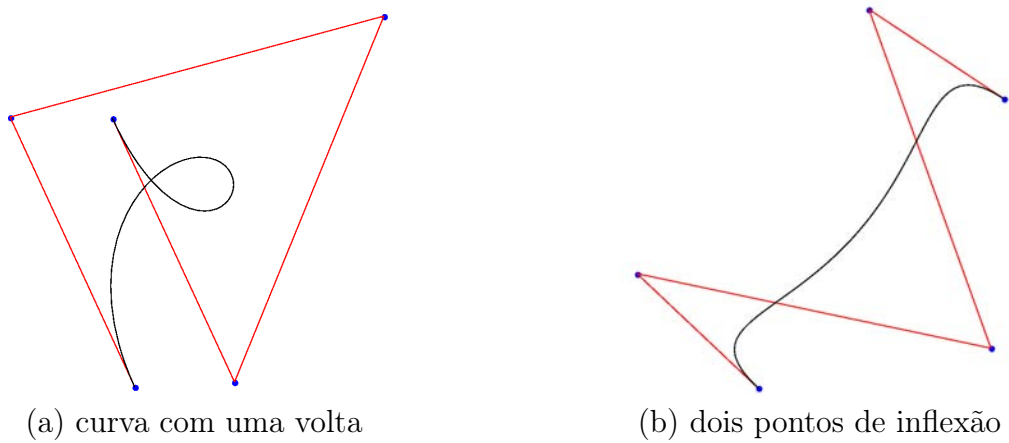


Figura 2.3: Curvas Bézier de grau quatro.

Nas figuras 2.1, 2.2, 2.3 e 2.4 são apresentadas algumas curvas Bézier com seus respectivos polígonos de controle.

Nestas figuras pode-se observar que os polígonos de controle definem a geometria de uma maneira bastante intuitiva. Por esta razão o método ainda hoje é muito usado, e a maior parte dos programas comerciais tem implementado curvas Bézier.

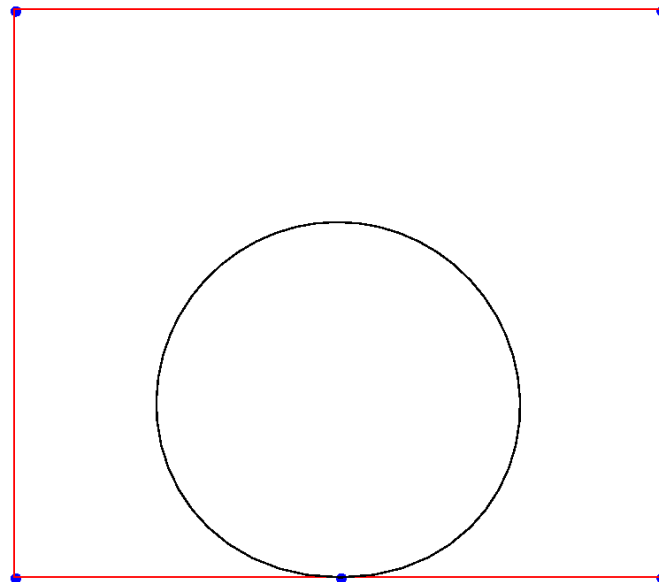


Figura 2.4: Curva Bézier de grau cinco.

Nas figuras 2.5 e 2.6 são apresentados alguns polinômios de Bernstein.

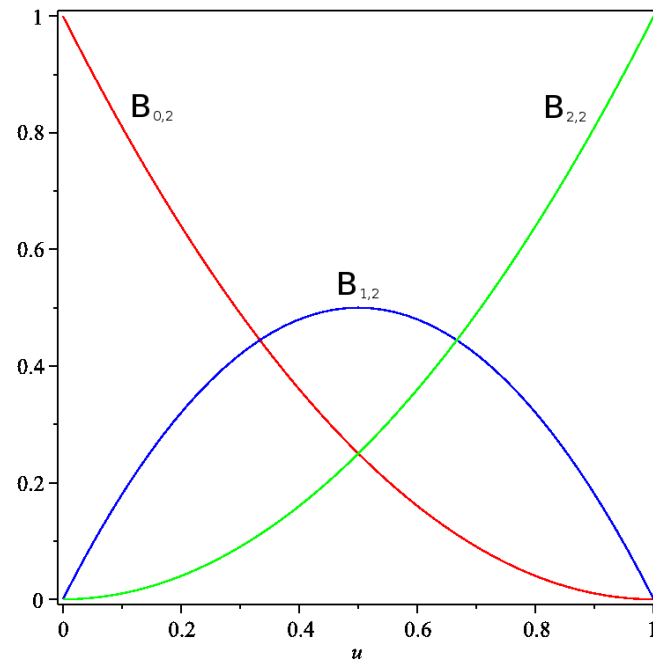


Figura 2.5: Polinômios de Bernstein de grau dois.

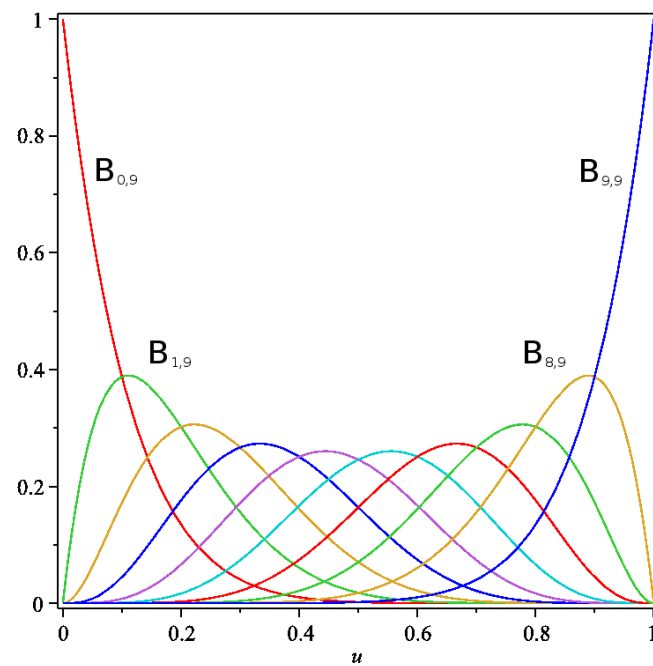


Figura 2.6: Polinômios de Bernstein de grau nove.

Propriedades (Piegl e Tiller, 1997):

P.Bézier. 1 *Função positiva semi-definida;*

P.Bézier. 2 *Partição da unidade:* $\sum_{i=0}^n B_{i,n} = 1 \quad \forall u \in [0,1];$

P.Bézier. 3 $B_{0,n}(0) = B_{n,n}(1) = 1;$

P.Bézier. 4 $B_{i,n}(u)$ possui um e somente um máximo no intervalo $[0,1]$ e este ocorre em $u = \frac{i}{n};$

P.Bézier. 5 *Simetria:* $\forall n$ o conjunto $\{B_{i,n}\}$ é simétrico em relação à $u = \frac{1}{2};$

P.Bézier. 6 *Definição recursiva:* $B_{i,n}(u) = (1-u)B_{i,n-1}(u) + uB_{i-1,n-1}(u)$, tendo definido $B_{i,n}(u) \equiv 0 \Leftrightarrow i < 0 \vee i > n;$

P.Bézier. 7 *Derivadas:* $B'_{i,n}(u) = \frac{dB_{i,n}(u)}{du} = n(B_{i-1,n-1}(u) - B_{i,n-1}(u))$ com $B_{-1,n-1}(u) \equiv B_{n,n-1}(u) \equiv 0.$

A propriedade P.Bézier. 6 produz um simples algoritmo para avaliar os polinômios de Bernstein para um dado valor do parâmetro u , já que desta mesma propriedade pode-se concluir que um polinômio de grau n , é a combinação linear de dois polinômios de grau $n - 1$, para um dado valor do parâmetro.

Da propriedade P.Bézier. 7 tem-se que a derivada de um polinômio Bézier de grau n é um outro polinômio Bézier de grau $n - 1$. Derivando a curva temos

$$\begin{aligned} C'(u) &= \frac{d(\sum_{i=0}^n B_{i,n}(u) \mathbf{P}_i)}{du} = \sum_{i=0}^n B'_{i,n}(u) \mathbf{P}_i \\ &= \sum_{i=0}^n n(B_{i-1,n-1}(u) - B_{i,n-1}(u)) \mathbf{P}_i \\ &= n \sum_{i=0}^n B_{i,n-1}(u) (\mathbf{P}_{i+1} - \mathbf{P}_i) \end{aligned} \quad (2.7)$$

Deixando $n = 2$ e $\mathbf{C}(u) = \sum_{i=0}^2 B_{i,2}(u) \mathbf{P}_i$, tem-se

$$\begin{aligned} \mathbf{C}(u) &= (1-u)^2 \mathbf{P}_0 + 2u(1-u) \mathbf{P}_1 + u^2 \mathbf{P}_2 \\ &= (1-u) \underbrace{[(1-u) \mathbf{P}_0 + u \mathbf{P}_1]}_{\text{linear}} + u \underbrace{[(1-u) \mathbf{P}_1 + u \mathbf{P}_2]}_{\text{linear}} \end{aligned}$$

Logo, $\mathbf{C}(u)$ é obtido a partir da interpolação linear de duas curvas Bézier de primeiro grau; em particular, um ponto sobre $\mathbf{C}(u)$ é obtido por três interpolações lineares.

Denotando, para uma curva Bézier geral de n -ésimo grau, $\mathbf{C}_n(\mathbf{P}_0, \dots, \mathbf{P}_n)$ tem-se

$$\mathbf{C}_n(\mathbf{P}_0, \dots, \mathbf{P}_n) = (1-u) \mathbf{C}_{n-1}(\mathbf{P}_0, \dots, \mathbf{P}_{n-1}) + u \mathbf{C}_{n-1}(\mathbf{P}_1, \dots, \mathbf{P}_n) \quad (2.8)$$

deCasteljau, engenheiro contemporâneo a Bézier da corporação Citroën, a partir da expressão (2.8) desenvolveu o algoritmo para avaliar um ponto sobre uma curva Bézier para um dado valor do parâmetro. Fixando $u = u_0$ e denotando \mathbf{P}_i como $\mathbf{P}_{0,i}$, a expressão (2.8) produz o algoritmo de deCasteljau, $\mathbf{C}(u_0) = \mathbf{P}_{n,0}(u_0)$

$$\mathbf{P}_{k,i}(u_0) = (1-u_0) \mathbf{P}_{k-1,i}(u_0) + u_0 \mathbf{P}_{k-1,i+1}(u_0) \quad \text{para} \quad \begin{cases} k = 1, \dots, n \\ i = 0, \dots, n-k \end{cases} \quad (2.9)$$

As figuras 2.7 e 2.8, apresentam a proposta de deCasteljau.

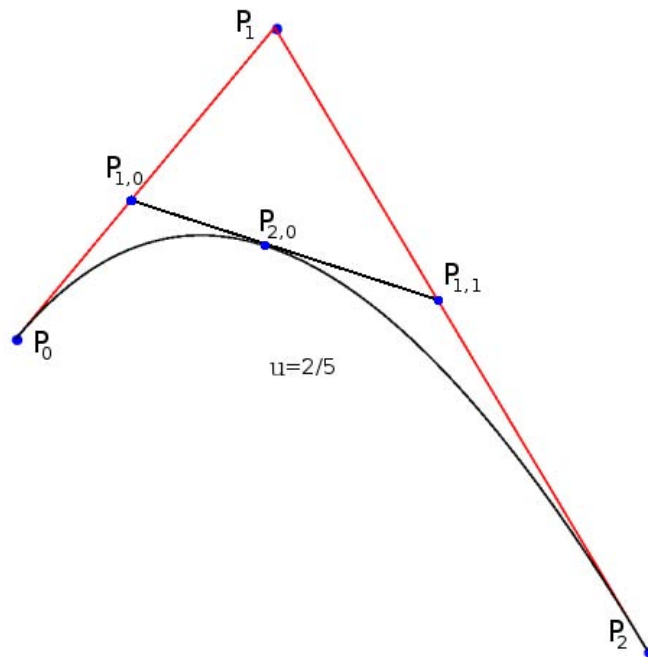


Figura 2.7: Ponto sobre curva quadrática por repetidas interpolações lineares em u_0 .

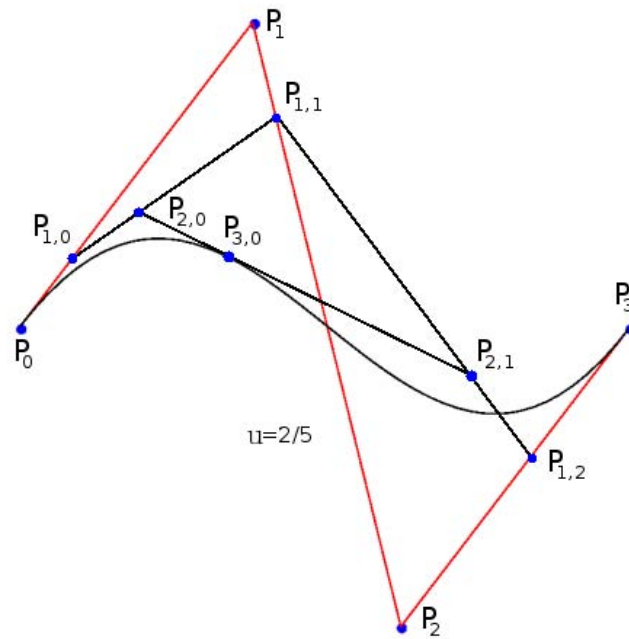


Figura 2.8: Ponto sobre curva cúbica por repetidas interpolações lineares em u_0 .

2.4 Coordenadas homogêneas

O conceito de infinito na computação somente pode ser definido com a introdução do conceito de coordenadas homogêneas. Considere a quantidade a e um número real w , onde a quantidade a pode ser um ponto no \mathbb{R}^1 , \mathbb{R}^2 ou \mathbb{R}^3 . Fixando a e variando w , a relação $v = \frac{a}{w}$ pode tomar uma gama de variação de menos a mais infinito. Computacionalmente pode-se definir v como o par ordenado (a,w) , ou seja, para todo w diferente de zero, $v = (a,w)$ em coordenadas homogêneas ou $v = \frac{a}{w}$ em coordenadas cartesianas; se $w = 0$ então o ponto $(a,0)$ representa um ponto no infinito.

Tomando o conceito de infinito para o sistema de coordenadas cartesianas, pode-se substituir a tríade ordenada (x,y,z) por $(\frac{x}{w}, \frac{y}{w}, \frac{z}{w})$; neste caso a função $f(x,y,z)$ torná-se $f(\frac{x}{w}, \frac{y}{w}, \frac{z}{w})$. Sendo f uma função de n -ésimo grau, multiplicando f por w^n se eliminará todos os denominadores e ainda todas as condições (termos dos polinômios) serão de grau n . Como consequência estes polinômios são chamados de polinômios homogêneos e as coordenadas (xw,yw,zw,w) são chamadas de coordenadas homogêneas.

Sabe-se da matemática clássica que todas as cônicas, curvas e superfícies, podem ser representadas através de polinômios racionais, apresentado por Roberts (1965), Riesenfeld

(1981) e Patterson (1985). A idéia das coordenadas homogêneas é representar uma curva polinômial racional no espaço n -dimensional como uma curva polinômial no espaço $(n + 1)$ -dimensional. Os pontos $\mathbf{P} = (x,y,z)$ no espaço homogêneo resultam ser $\mathbf{P}^w = (xw,yw,zw,w) = (X,Y,Z,W)$ no espaço de quarta dimensão. \mathbf{P} é obtido de \mathbf{P}^w dividindo todas as coordenadas pela quarta coordenada, W , i.e., quando se faz o mapeamento de \mathbf{P}^w desde a origem ao hiperplano $W = 1$. Este mapeamento, denotado por \mathcal{H} , é uma perspectiva com ponto de fuga na origem sobre o hiperplano $W = 1$

$$\mathbf{P} = \mathcal{H}\{\mathbf{P}^w\} = \mathcal{H}\{(X,Y,Z,W)\} = \begin{cases} \left(\frac{X}{W}, \frac{Y}{W}, \frac{Z}{W}\right) & \text{se } W \neq 0 \\ \text{direção } (X,Y,Z) & \text{se } W = 0 \end{cases} \quad (2.10)$$

A figura 2.9 exemplifica para duas dimensões este conceito.

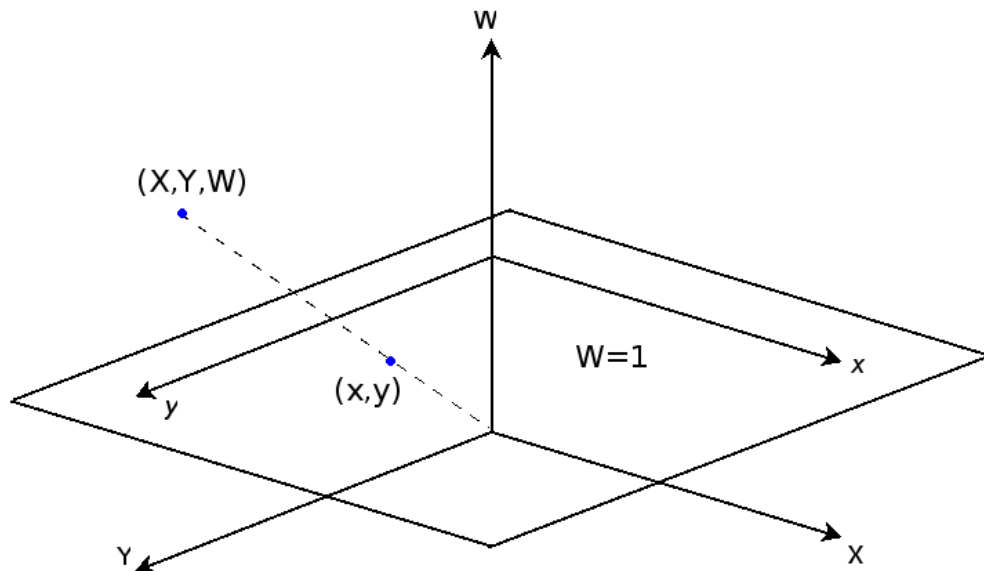


Figura 2.9: Transformação de um ponto do espaço homogêneo ao espaço Euclidiano.

Obviamente, para mapear um ponto do espaço paramétrico ao espaço Euclidiano, para representações paramétricas, se faz o mapeamento do espaço paramétrico para o espaço homogêneo e do espaço homogêneo ao espaço Euclidiano.

2.5 Bézier racional

Unindo os conceitos apresentados nas seções 2.3 e 2.4, têm-se funções racionais que podem ser escritas na forma

$$x(u) = \frac{X(u)}{W(u)} \quad y(u) = \frac{Y(u)}{W(u)} \quad z(u) = \frac{Z(u)}{W(u)} \quad (2.11)$$

onde $X(u)$, $Y(u)$, $Z(u)$ e $W(u)$ são polinômios, e ainda, cada coordenada tem o denominador comum. Definindo uma curva Bézier racional de n -ésimo grau por

$$\mathbf{C}(u) = \frac{\sum_{i=0}^n B_{i,n}(u) w_i \mathbf{P}_i}{\sum_{i=0}^n B_{i,n}(u) w_i} \quad u \in [0,1] \quad (2.12)$$

\mathbf{P}_i e $B_{i,n}$ foram definidos anteriormente na seção 2.3, os w_i são escalares, chamados de *pesos*. Então, $W(u) = \sum_{i=0}^n B_{i,n}(u) w_i$ é o denominador comum da função. Exceto quando explícito, $w_i > 0$, assim $W(u) > 0$ para todo $u \in [0,1]$. Pode-se escrever (2.12) da seguinte maneira

$$\mathbf{C}(u) = \sum_{i=0}^n R_{i,n}(u) \mathbf{P}_i \quad u \in [0,1] \quad (2.13)$$

onde

$$R_{i,n}(u) = \frac{B_{i,n}(u) w_i}{\sum_{j=0}^n B_{j,n}(u) w_j} \quad (2.14)$$

As $R_{i,n}(u)$ são as funções de base racionais. A figura 2.10 mostra a transformação de uma curva do espaço homogêneo ao espaço Euclideano, tornando esta uma curva racional.

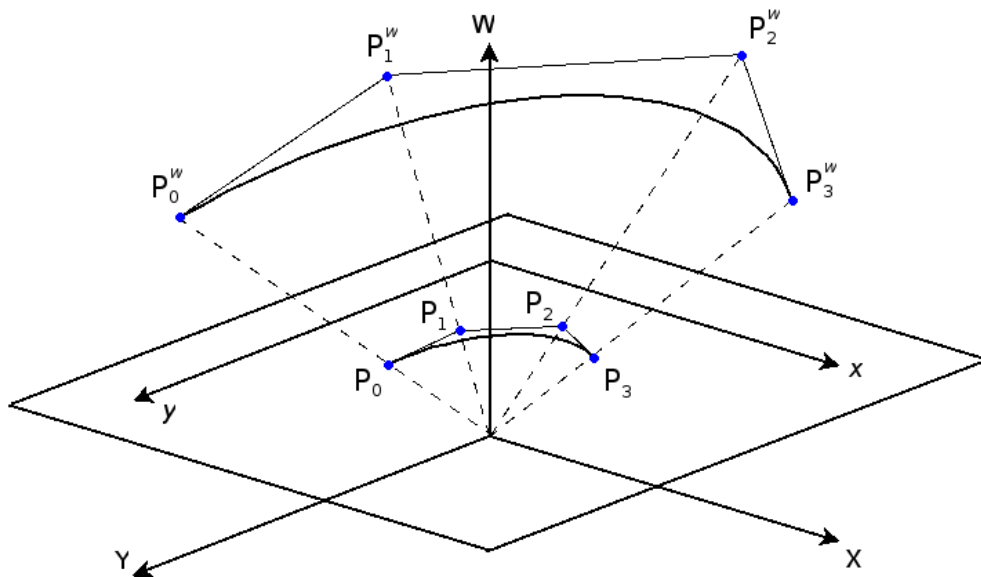


Figura 2.10: Transformação de uma curva do espaço homogêneo ao espaço Euclideano.

As propriedades de $R_{i,n}$ são derivadas da expressão (2.13) e das propriedades de $B_{i,n}$.

Propriedades (Piegl e Tiller, 1997):

P.Bézier. 8 *Função positiva semi-definida;*

P.Bézier. 9 *Partição da unidade:* $\sum_{i=0}^n R_{i,n} = 1 \quad \forall u \in [0,1];$

P.Bézier. 10 $R_{0,n}(0) = R_{n,n}(1) = 1;$

P.Bézier. 11 $R_{i,n}(u)$ possui um e somente um máximo no intervalo $[0,1];$

P.Bézier. 12 se $w_i = 1 \quad \forall i;$ então $R_{i,n}(u) = B_{i,n}(u) \quad \forall i;$ i.e., $B_{i,n}(u)$ é um caso particular de $R_{i,n}(u);$

Estas produzem propriedades geométricas das curvas racionais Bézier;

P.Bézier. 13 *Região convexa:* a curva está contida na região convexa formada pelo polígono de controle;

P.Bézier. 14 *Transformações lineares:* transformações tais como rotação, translação, escala e cisalhamento aplicados à curva são realizadas aplicando tal transformação ao polígono de controle;

P.Bézier. 15 *Nenhuma reta ou plano contém mais interseções com uma curva Bézier do que esta reta ou plano contém com o polígono de controle da curva;*

P.Bézier. 16 *Interpolação dos pontos finais:* $\mathbf{C}(0) = \mathbf{P}_0$ e $\mathbf{C}(1) = \mathbf{P}_n;$

P.Bézier. 17 *A k -ésima derivada em $u = 0$ e $u = 1$ depende dos primeiros e últimos $k + 1$ pontos de controle e pesos; em particular $\mathbf{C}'(0)$ e $\mathbf{C}'(1)$, são paralelos à $\mathbf{P}_1 - \mathbf{P}_0$ e $\mathbf{P}_n - \mathbf{P}_{n-1}$, respectivamente;*

P.Bézier. 18 *Curvas Bézier são um caso particular de curvas racionais Bézier.*

A figura 2.11 apresenta a construção de um arco de circunferência e a figura 2.12 apresenta o mapeamento do arco do espaço homogêneo ao espaço Euclidiano. A descrição paramétrica desta curva Bézier racional é

$$\mathbf{C}(u) = (x(u), y(u)) = \left(\frac{1-u^2}{1+u^2}, \frac{2u}{1+u^2} \right) \quad u \in [0,1] \quad (2.15)$$

Para um dado conjunto de pontos de controle, \mathbf{P}_i , e pesos, w_i , os pontos de controle em coordenadas homogêneas resultam ser $\mathbf{P}_i^w = (w_i x_i, w_i y_i, w_i z_i, w_i)$, e assim se pode definir a curva Bézier *não-racional* no espaço de quarta dimensão (em coordenadas homogêneas) como

$$\mathbf{C}^w(u) = \sum_{i=0}^n B_{i,n}(u) \mathbf{P}_i^w \quad (2.16)$$

O algoritmo de deCasteljau (2.9), pode ser utilizado para Bézier em coordenadas homogêneas

$$\mathbf{P}_{k,i}^w(u_0) = (1-u_0) \mathbf{P}_{k-1,i}^w(u_0) + u_0 \mathbf{P}_{k-1,i+1}^w(u_0) \quad \text{para } \begin{cases} k = 1, \dots, n \\ i = 0, \dots, n-k \end{cases} \quad (2.17)$$

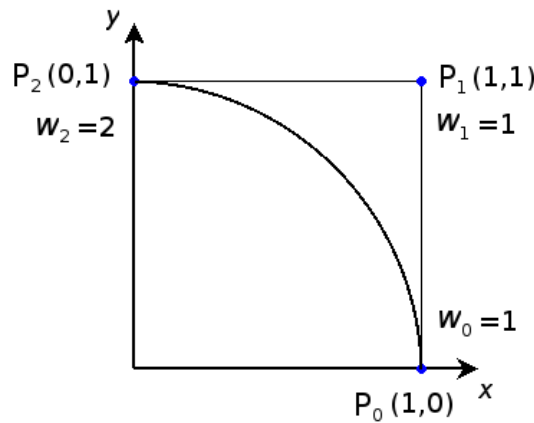


Figura 2.11: Construção de um arco de circunferência.

Na tabela 2.1 a expressão (2.17) é exemplificada para um ponto em um arco de circunferência, com o valor do parâmetro $u = \frac{1}{2}$.

Mapeando este ponto para o espaço Euclidiano, tem-se $\mathbf{C}\left(\frac{1}{2}\right) = \left(\frac{3}{5}, \frac{4}{5}\right)$. Observar que o ponto médio no espaço paramétrico não resultou ser o ponto médio no espaço Euclidiano;

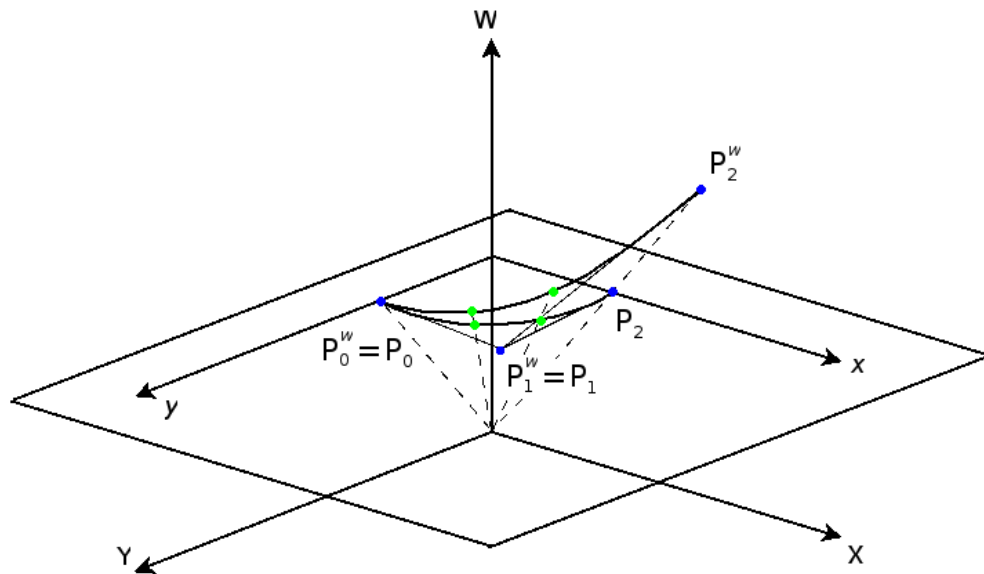


Figura 2.12: Mapeamento do arco do espaço homogêneo ao espaço Euclidiano.

Tabela 2.1: Geração do ponto $C^w(\frac{1}{2})$ em um arco de circunferência.

$P_{0,i=0,1,2}^w$	$P_{1,i=0,1}^w$	$P_{2,0}^w$	
(1,0,1)			
	$\left(1, \frac{1}{2}, 1\right)$		
(1,1,1)		$\left(\frac{3}{4}, 1, \frac{5}{4}\right)$	$= C^w\left(\frac{1}{2}\right)$
	$\left(\frac{1}{2}, \frac{3}{2}, \frac{3}{2}\right)$		
(0,2,2)			

isto ocorre pelo fato da parametrização não ser uniforme, i.e., a velocidade do parâmetro não é constante. No ponto inicial a magnitude da derivada vale o dobro do que no ponto final (diferenciando a expressão (2.15) se percebe este fato).

2.6 Superfície: produto tensorial

As curvas são funções vetoriais que relacionam \mathbb{R}^1 com \mathbb{R}^2 ou \mathbb{R}^3 , i.e., são vetores que dependem de um parâmetro, onde se faz o mapeamento do espaço paramétrico para o espaço homogêneo e do espaço homogêneo ao espaço Euclidiano. As superfícies são funções vetoriais que relacionam \mathbb{R}^2 com \mathbb{R}^3 , i.e., são vetores que dependem de dois parâmetros, onde se faz o mapeamento do espaço paramétrico para o espaço homogêneo

e do espaço homogêneo ao espaço Euclidiano.

As superfícies sempre resultam ser da forma (2.2), entretanto existem diversos esquemas para representar superfícies, sendo que alguns destes podem ser vistos em Piegl e Tiller (1989). Provavelmente o método mais simples e o mais usado para modelagem geométrica é o esquema de *produto tensorial*, que será o esquema empregado neste trabalho.

Basicamente, uma superfície gerada por produto tensorial, constitui um esquema de curvas bidirecionais. Este esquema usa funções de base e coeficientes geométricos. As funções de base são bivariantes de u e v , as quais são construídas pelo produto de funções de base univariantes. Os coeficientes geométricos são arranjados (topologicamente) em uma rede bidirecional $n \times m$. Logo, uma superfície de produto tensorial possui a forma

$$\mathbf{S}(u,v) = (x(u,v), y(u,v), z(u,v)) = \sum_{i=0}^n \sum_{j=0}^m f_i(u) g_j(v) \mathbf{b}_{i,j} \quad (2.18)$$

onde

$$\begin{cases} \mathbf{b}_{i,j} = (x_{i,j}, y_{i,j}, z_{i,j}) \\ (u,v) \in [0,1] \times [0,1] \end{cases}$$

Note que o domínio paramétrico (u,v) é um quadrado; mais adiante são usados domínios retangulares. A superfície (2.18) na forma matricial toma a seguinte forma

$$\mathbf{S}(u,v) = [f_i(u)]^T [\mathbf{b}_{i,j}] [g_j(v)] \quad (2.19)$$

onde $[f_i(u)]^T$ é um vetor em linha de dimensão $(n+1)$, $[g_j(v)]$ é um vetor em coluna de dimensão $(m+1)$ e $[\mathbf{b}_{i,j}]$ é uma matriz de dimensão $(n+1) \times (m+1)$ dos pontos de controle em três dimensões.

Superfícies Bézier não-rationais, são obtidas tomando uma rede bidirecional dos pontos de controle e o produto dos polinômios univariantes de Bernstein.

$$\mathbf{S}(u,v) = \sum_{i=0}^n \sum_{j=0}^m B_{i,n}(u) B_{j,m}(v) \mathbf{P}_{i,j} \quad (u,v) \in [0,1] \times [0,1] \quad (2.20)$$

Para um $u = u_0$

$$\begin{aligned} \mathbf{C}_{u_0}(v) = \mathbf{S}(u_0,v) &= \sum_{i=0}^n \sum_{j=0}^m B_{i,n}(u_0) B_{j,m}(v) \mathbf{P}_{i,j} \\ &= \sum_{j=0}^m B_{j,m}(v) \left(\sum_{i=0}^n B_{i,n}(u_0) \mathbf{P}_{i,j} \right) \\ &= \sum_{j=0}^m B_{j,m}(v) \mathbf{Q}_j(u_0) \end{aligned} \quad (2.21)$$

onde $\mathbf{Q}_j(u_0) = \sum_{i=0}^n B_{i,n}(u_0) \mathbf{P}_{i,j}$ para $j = 0, \dots, m$ é uma curva Bézier que está contida na superfície. De forma análoga tem-se, $\mathbf{C}_{v_0}(u) = \sum_{i=0}^n B_{i,n}(u) \mathbf{Q}_i(v_0)$.

As propriedades das superfícies Bézier são análogas às propriedades já explanadas para as curvas, com exceção da propriedade P.Bézier. 15; esta não possui uma analogia.

O algoritmo de deCasteljau facilmente é estendido para avaliar um ponto sobre uma superfície Bézier. Da expressão (2.21), com (u_0, v_0) e $j = j_0$ tem-se $\mathbf{Q}_{j_0}(u_0) = \sum_{i=0}^n B_{i,n}(u_0) \mathbf{P}_{i,j_0}$ pela aplicação do algoritmo de deCasteljau à linha j_0 dos pontos de controle, i.e, aplicado à \mathbf{P}_{i,j_0} , $i = 0, \dots, n$. Assim tem-se um novo ponto de controle sobre a linha j_0 . Logo, aplicando o mesmo $(m+1)$ vezes se produz $\mathbf{C}_{u_0}(v)$, curva Bézier sobre o parâmetro u_0 . Aplicando o algoritmo de deCasteljau uma vez mais para o parâmetro v_0 sobre a curva Bézier $\mathbf{C}_{u_0}(v)$ tem-se $\mathbf{C}_{u_0}(v_0) = \mathbf{S}(u_0, v_0)$. Este processo requer

$$\frac{n(n+1)(m+1)}{2} + \frac{m(m+1)}{2} \quad (2.22)$$

interpolações lineares.

Se $n > m$ então se computa $\mathbf{C}_{u_0}(v)$ e logo $\mathbf{C}_{u_0}(v_0)$

Definindo uma *superfície Bézier racional* pela projeção perspectiva de uma superfície Bézier polinomial de quatro dimensões, tem-se

$$\mathbf{S}^w(u, v) = \sum_{i=0}^n \sum_{j=0}^m B_{i,n}(u) B_{j,m}(v) \mathbf{P}_{i,j}^w \quad (2.23)$$

e

$$\begin{aligned} \mathbf{S}(u, v) = \mathcal{H}\{\mathbf{S}^w(u, v)\} &= \frac{\sum_{i=0}^n \sum_{j=0}^m B_{i,n}(u) B_{j,m}(v) w_{i,j} \mathbf{P}_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m B_{i,n}(u) B_{j,m}(v) w_{i,j}} \\ &= \sum_{i=0}^n \sum_{j=0}^m R_{i,n}(u, v) \mathbf{P}_{i,j} \end{aligned} \quad (2.24)$$

onde

$$R_{i,n}(u, v) = \frac{B_{i,n}(u) B_{j,m}(v) w_{i,j}}{\sum_{r=0}^n \sum_{s=0}^m B_{r,n}(u) B_{s,m}(v) w_{r,s}} \quad (2.25)$$

Observar que $R_{i,n}(u, v)$ são funções racionais, mas não são o produto das outras funções de base, da expressão (2.14). Logo $\mathbf{S}(u, v)$ não é uma superfície por produto tensorial, mas $\mathbf{S}^w(u, v)$ sim. Assim como para as curvas, os algoritmos desenvolvidos foram elaborados em coordenadas homogêneas, e o resultado foi mapeado para o espaço Euclidiano.

2.7 Funções de base B-spline

Curvas e superfícies que consistem de apenas um polinômio ou um polinômio racional são inadequadas em geral para representar geometrias complexas, já que

- um alto grau é necessário para satisfazer um grande número de restrições; dados n pontos se precisa de um polinômio de grau $(n - 1)$. Ainda, polinômios de alto grau são ineficientes para processar e são numericamente instáveis;
- curvas e superfícies de alto grau são necessárias para ajustar formas complexas;
- curvas e superfícies que consistem de apenas um segmento são difíceis de manipular, i.e., o controle não é suficientemente local.

A solução é usar curvas e superfícies que são descritas por polinômios por partes, ou polinômios racionais por partes. A figura 2.13 mostra uma curva, $\mathbf{C}(u)$, que consiste de

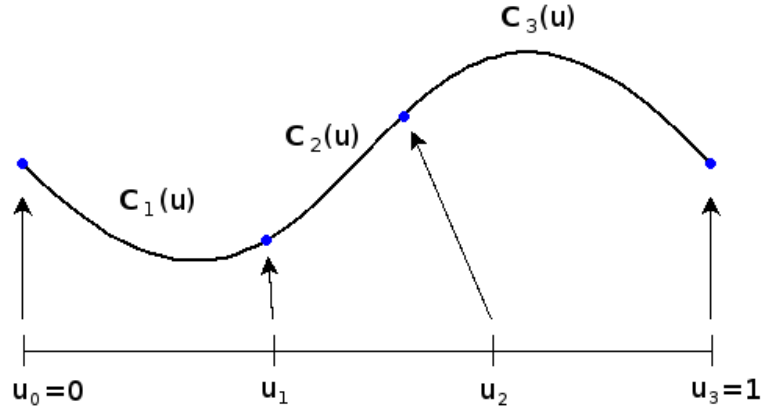


Figura 2.13: Três polinômios por partes de grau três.

três ($m = 3$) polinômios de grau três. $\mathbf{C}(u)$ está definida em $u \in [0, 1]$. Os valores dos parâmetros $u_0 = 0 < u_1 < u_2 < u_3 = 1$ são chamados de *pontos de parada* (*breakpoints*). Eles são os pontos dos extremos de cada segmento de polinômio. Denota-se por $\mathbf{C}_i(u)$ com $i \in [1, m]$ os segmentos da curva. Os segmentos são construídos de maneira que na união se tenha algum nível de continuidade, não necessariamente o mesmo para cada ponto de parada. Sendo $\mathbf{C}_i^{(j)}(u)$ a j -ésima derivada de $\mathbf{C}_i(u)$, $\mathbf{C}(u)$ possui continuidade paramétrica C^k no ponto de parada u_i se $\mathbf{C}_i^{(j)}(u_i) = \mathbf{C}_{i+1}^{(j)}(u_i)$ para todo $j \in [0, k]$.

Obviamente, os polinômios aqui usados serão os polinômios na forma Bézier. A figura 2.14 apresenta três segmentos na forma cúbica de Bézier. \mathbf{P}_i^j denota o i -ésimo ponto de

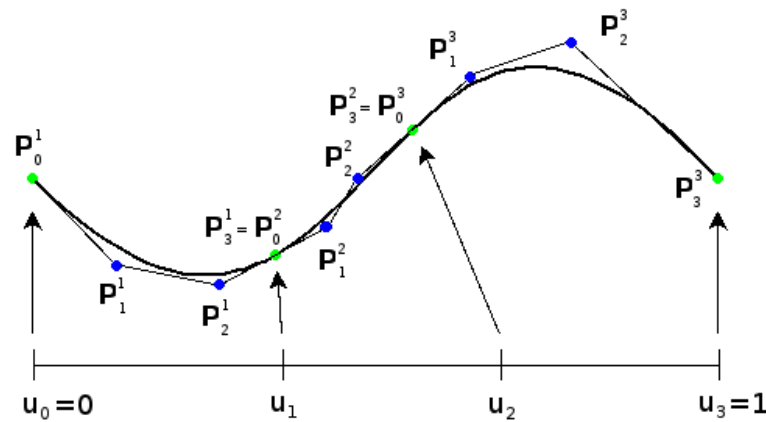


Figura 2.14: Curva da figura 2.13 na forma Bézier.

controle do j -ésimo segmento.

Se o grau três, for mantido, e os pontos de parada $\mathcal{U} = \{u_0, u_1, u_2, u_3\}$ forem fixados, permitindo os doze pontos de controle, \mathbf{P}_i^j , variarem arbitrariamente, se obtém um espaço vetorial, \mathcal{S} , contendo todas as curvas polinomiais por partes de grau três sobre \mathcal{U} .

\mathcal{S} tem dimensão doze, e uma curva em \mathcal{S} pode ser descontínua em u_1 ou u_2 . Supondo que se especifica (como na figura 2.14) $\mathbf{P}_3^1 = \mathbf{P}_0^2$ e $\mathbf{P}_3^2 = \mathbf{P}_0^3$, isto produz um espaço vetorial \mathcal{S}^0 , i.e., todas as curvas polinomiais por partes de grau três sobre \mathcal{U} são, pelo menos, de continuidade C^0 em qualquer ponto. \mathcal{S}^0 tem dimensão dez e $\mathcal{S}^0 \subset \mathcal{S}$.

Impondo continuidade C^1 (já considerando $\mathbf{P}_3^1 = \mathbf{P}_0^2$), usando a expressão (2.7) e avaliando em zero e em um, tem-se

$$\mathbf{C}'(0) = n(\mathbf{P}_1 - \mathbf{P}_0) \quad \mathbf{C}'(1) = n(\mathbf{P}_n - \mathbf{P}_{n-1}) \quad (2.26)$$

Se faz uma mudança de variável do tipo

$$v = \frac{u - u_0}{u_1 - u_0} \quad w = \frac{u - u_1}{u_2 - u_1}$$

sendo v e w parâmetros locais nos intervalos $[u_0, u_1]$ e $[u_1, u_2]$, respectivamente. Então, $(v, w) \in [0, 1] \times [0, 1]$. Continuidade C^1 em u_1 implica

$$\frac{1}{u_1 - u_0} \mathbf{C}_1^{(1)}(v = 1) = \mathbf{C}_1^{(1)}(u_1) = \mathbf{C}_2^{(1)}(u_1) = \frac{1}{u_2 - u_1} \mathbf{C}_2^{(1)}(w = 0)$$

e a partir das expressões de (2.26), tem-se que

$$\frac{3}{u_1 - u_0} (\mathbf{P}_3^1 - \mathbf{P}_2^1) = \frac{3}{u_2 - u_1} (\mathbf{P}_1^2 - \mathbf{P}_0^2)$$

logo

$$\mathbf{P}_3^1 = \frac{(u_2 - u_1) \mathbf{P}_2^1 + (u_1 - u_0) \mathbf{P}_1^2}{u_2 - u_0} \quad (2.27)$$

A expressão (2.27) mostra a dependência de \mathbf{P}_3^1 e \mathbf{P}_3^2 , já que podem ser escritos em função de \mathbf{P}_2^1 , \mathbf{P}_1^2 e \mathbf{P}_2^2 , \mathbf{P}_1^3 , respectivamente.

Logo, isto produz um espaço vetorial \mathcal{S}^1 , i.e., todas as curvas polinomiais por partes de grau três sobre \mathcal{U} são, pelo menos, de continuidade C^1 em qualquer ponto. \mathcal{S}^1 tem dimensão oito e $\mathcal{S}^1 \subset \mathcal{S}^0 \subset \mathcal{S}$. A continuidade C^2 é obtida com o espaço vetorial \mathcal{S}^2 de dimensão seis.

Observar que o que se procura é uma representação para uma curva do tipo

$$\mathbf{C}(u) = \sum_{i=0}^n f_i(u) \mathbf{P}_i \quad (2.28)$$

onde \mathbf{P}_i são os *pontos de controle*, e as $\{f_i(u), i = 0, \dots, n\}$ são *funções polinomiais por partes* formando uma base do espaço vetorial de todas as funções polinomiais por partes que satisfaçam as expectativas prescritas, tais como grau e continuidade, (para uma sequência de pontos de parada fixa $\mathcal{U} = \{u_i\}, i \in [0, m]$). A continuidade vem dada pelas funções de base, já que os pontos de controle podem ser manipulados sem alterar a continuidade da curva. Ainda, $\{f_i\}$ podem ser as apresentadas na seção 2.3; isto assegura que as curvas definidas por (2.28) possuem propriedades similares às propriedades das curvas Bézier. Outra propriedade importante destas novas funções de base é o *suporte local*; isto implica que cada $f_i(u)$ não é zero somente em um número limitado de subintervalos, não no domínio inteiro, $[u_0, u_m]$. Já que \mathbf{P}_i é multiplicado por $f_i(u)$, movendo \mathbf{P}_i somente é afetada a forma da curva no subintervalo onde $f_i(u)$ não for zero.

Finalmente, dado um polinômio por partes adequado, podem-se construir curvas por partes racionais.

$$\mathbf{C}^w(u) = \sum_{i=0}^n f_i(u) \mathbf{P}_i^w \quad (2.29)$$

Uma superfície não-racional e racional por produto tensorial, são dadas respectivamente

$$\mathbf{S}(u, v) = \sum_{i=0}^n \sum_{j=0}^m f_i g_j(u) \mathbf{P}_{i,j} \quad (2.30)$$

$$\mathbf{S}^w(u, v) = \sum_{i=0}^n \sum_{j=0}^m f_i g_j(u) \mathbf{P}_{i,j}^w \quad (2.31)$$

2.7.1 Definição e propriedades das funções de base B-spline

Existem inúmeras maneiras para definir as funções de base B-spline, a mais aceita e conhecida foi dada por deBoor (1972) e por Cox (1972), a definição recursiva.

Sendo $\mathcal{U} = \{u_0, \dots, u_m\}$ uma sequência de números reais crescentes (não necessariamente monótona crescente), i.e., $u_i \leq u_{i+1}$, $i = 0, \dots, m-1$. Os u_i são os nós (*knots*), e \mathcal{U} é o vetor de nós (*knots vector*). A i -ésima função de base B-spline de grau p (ordem $p+1$), denotada por $N_{i,p}(u)$, é definida como

$$\begin{aligned} N_{i,0}(u) &= \begin{cases} 1 & \text{se } u_i \leq u \leq u_{i+1} \\ 0 & \text{caso contrário} \end{cases} \\ N_{i,p}(u) &= \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \end{aligned} \quad (2.32)$$

Observar que

- $N_{i,0}(u)$ é a função de *Heaviside* (de grau unitário), igual à zero em todo o domínio exceto no intervalo semi-aberto $u \in [u_i, u_{i+1})$;
- para $p > 0$, $N_{i,p}(u)$ é uma combinação linear de duas funções de base de grau $(p-1)$;
- o computo de um conjunto de funções de base requer a especificação do vetor de nós, \mathcal{U} , e do grau, p ;
- a expressão (2.32) pode produzir o quociente zero sobre zero; este é definido igual à zero;
- as $N_{i,p}(u)$ funções polinomiais por partes estão definidas no conjunto dos reais; em geral somente interessam no intervalo semi-aberto $[u_0, u_m)$;
- o intervalo semi-aberto $[u_i, u_{i+1})$, é chamado de período do i -ésimo nó; este pode ter comprimento zero já que os nós podem não ser distintos;
- o computo das funções de base geram uma tabela triangular truncada, tabela 2.2.

Tabela 2.2: Tabela triangular truncada das $N_{i,p}(u)$.

$N_{0,0}$			
$N_{1,0}$	$N_{0,1}$		
$N_{2,0}$	$N_{1,1}$	$N_{0,2}$	
$N_{3,0}$	$N_{2,1}$	$N_{1,2}$	$N_{0,3}$
$N_{4,0}$	$N_{3,1}$	$N_{2,2}$	$N_{1,3}$
\vdots	\vdots	\vdots	\vdots
\vdots	\vdots	\vdots	\vdots

Por simplificação $N_{i,p}(u)$ pode ser escrito como $N_{i,p}$.

Em relação à nomenclatura, o termo pontos de parada (*breakpoints*) é usado sempre que $u_i < u_{i+1}$, i.e., sequência monótona crescente; o termo vetor de nós (*knots vector*) é usado sempre que $u_i \leq u_{i+1}$, i.e., sequência crescente. O termo B-*spline* significa que se está fazendo uso de funções de Bernstein (B) por partes (*spline*).

Logo, um caso particular das curvas e superfícies B-*spline* devem ser as curvas e superfícies Bézier. Esta particularização vem dada sempre que o vetor de nós vier dado na forma

$$\mathcal{U} = \{\underbrace{0, \dots, 0}_{p+1}, \underbrace{1, \dots, 1}_{p+1}\}$$

Sendo

$\mathcal{U} = \{u_0 = 0, u_1 = 0, u_2 = 0, u_3 = 1, u_4 = 2, u_5 = 3, u_6 = 4, u_7 = 4, u_8 = 5, u_9 = 5, u_{10} = 5\}$ e $p = 2$. As funções de base, não nulas, de grau zero, um e dois são apresentadas nas figuras 2.15, 2.16 e 2.17, respectivamente.

Algumas propriedades importantes das funções de base B-*spline* são listadas, as quais fornecem características geométricas desejáveis na modelagem e na manipulação de curvas e superfícies.

Propriedades (Piegl e Tiller, 1997):

P.B-*spline*. 1 Função positiva semi-definida (demonstrado por indução sobre p);

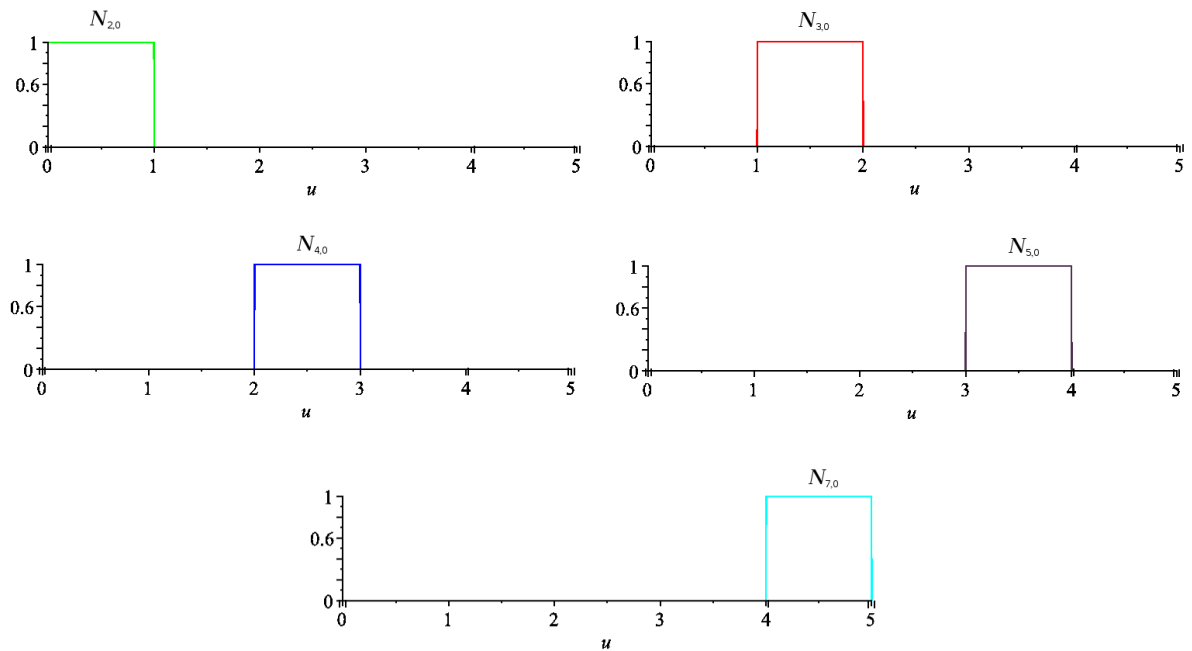


Figura 2.15: Funções de base não-nulas de zero grau.

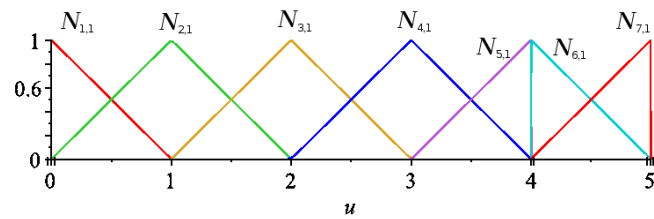


Figura 2.16: Funções de base não-nulas de primeiro grau.

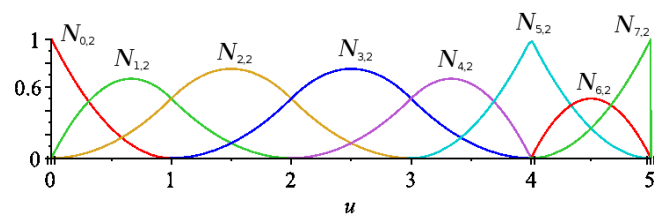


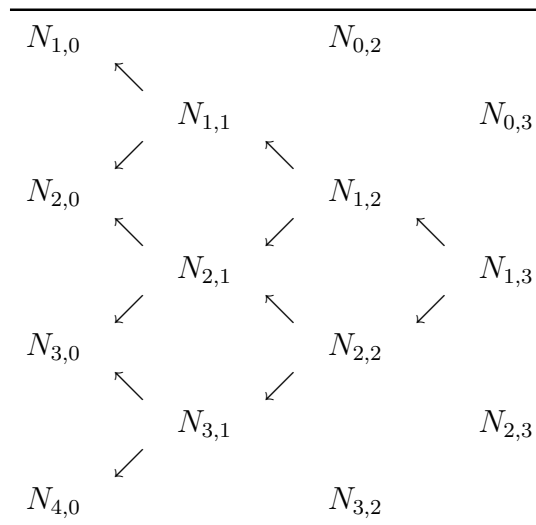
Figura 2.17: Funções de base não-nulas de segundo grau.

P.B-spline. 2 *Suporte local:* $N_{i,p}(u) = 0$ se u estiver fora do intervalo $[u_i, u_{i+p+1})$ (local support property);

Este esquema é ilustrado na tabela triangular 2.3. Observar que $N_{1,3}$ é uma combinação de $N_{1,0}$, $N_{2,0}$, $N_{3,0}$ e $N_{4,0}$. Logo, $N_{1,3}$ é não-nula somente no intervalo para $u \in [u_1, u_5)$

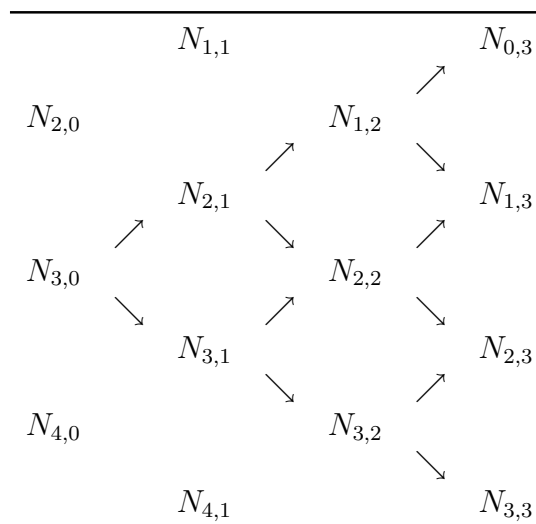
P.B-spline. 3 *Em um dado período* (knot span) $[u_j, u_{j+1})$, no máximo $p + 1$ das $N_{i,p}(u)$

Tabela 2.3: Suporte local.



são não-nulas, e são chamadas de $N_{j-p,p}, \dots, N_{j,p}$;

Sobre $u \in [u_3, u_4)$ a única função não-nula de grau zero é $N_{3,0}$, e as únicas funções não-nulas de grau três são $N_{0,3}, \dots, N_{3,3}$. Esta propriedade é ilustrada na tabela triangular 2.4.

Tabela 2.4: Funções não-nulas sobre $u \in [u_3, u_4)$.

P.B-spline. 4 Partição da unidade: dado um período arbitrário, $[u_i, u_{i+1})$,

$$\sum_{j=i-p}^i N_{j,p}(u) = 1 \quad \forall u \in [u_i, u_{i+1});$$

P.B-spline. 5 Todas as derivadas de $N_{i,p}(u)$ existem no interior do período. Sobre o nó, $N_{i,p}(u)$ é C^{p-k} contínuo, onde k é a multiplicidade do nó;

P.B-spline. 6 Exceto para $p = 0$, $N_{i,p}(u)$ possui um e somente um máximo.

O efeito da multiplicidade dos nós deve ser entendido em dois diferentes aspectos. Supondo um vetor de nós dados por $\mathcal{U} = \{0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5\}$ e o grau $p = 2$, pela propriedade P.B-spline. 2 tem-se que as funções base são computadas nos seguintes períodos, pois valem zero nos demais

$$N_{0,2} : \{0, 0, 0, 1\}$$

$$N_{1,2} : \{0, 0, 1, 2\}$$

$$N_{2,2} : \{0, 1, 2, 3\}$$

$$N_{3,2} : \{1, 2, 3, 4\}$$

$$N_{4,2} : \{2, 3, 4, 4\}$$

$$N_{5,2} : \{3, 4, 4, 5\}$$

$$N_{6,2} : \{4, 4, 5, 5\}$$

$$N_{7,2} : \{4, 5, 5, 5\}$$

Os dois diferentes aspectos da multiplicidade dos nós são:

- a multiplicidade do nó em relação ao vetor de nós;
- a multiplicidade do nó em relação às funções de base.

Para este dado vetor de nós, em $u = 0$ tem-se multiplicidade três, mas em relação às funções de base $N_{0,2}$, $N_{1,2}$ e $N_{2,2}$, em $u = 0$ tem-se multiplicidade três, dois e um, respectivamente. Logo, da propriedade P.B-spline. 5, tem-se que a continuidade das funções de base em $u = 0$ é:

- $N_{0,2}$ é descontínua;
- $N_{1,2}$ é C^0 contínua;

- $N_{2,2}$ é C^1 contínua;
- $N_{3,2}$, $N_{4,2}$, $N_{5,2}$, $N_{6,2}$ e $N_{7,2}$ não são afetadas.

Este fato mostra como as funções de base “percebem” os nós, i.e., $N_{0,2}$ percebe uma multiplicidade três enquanto $N_{0,2}$ percebe uma multiplicidade dois em $u = 0$. Assim como $N_{6,2}$ é C^0 contínua em $u = 4$ e $u = 5$.

2.7.2 Derivadas das funções de base B-spline

As derivadas das funções de base, vem dadas por

$$N'_{i,p}(u) = \frac{p}{u_{i+p} - u_i} N_{i,p-1}(u) - \frac{p}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (2.33)$$

A expressão (2.33) é demonstrada por indução sobre p (Piegl e Tiller, 1997). Agora, denotando por $N_{i,p}^{(k)}(u)$ a k -ésima derivada de $N_{i,p}(u)$, diferenciar repetidamente produz

$$N_{i,p}^{(k)}(u) = p \left(\frac{N_{i,p-1}^{(k-1)}}{u_{i+p} - u_i} - \frac{N_{i+1,p-1}^{(k-1)}}{u_{i+p+1} - u_{i+1}} \right) \quad (2.34)$$

A expressão (2.35), dada a seguir, é uma outra generalização de (2.34). Esta computa a derivada $N_{i,p}^{(k)}(u)$ em termos das funções $N_{i,p-k}, \dots, N_{i+k,p-k}$

$$N_{i,p}^{(k)}(u) = \frac{p!}{(p-k)!} \sum_{j=0}^k a_{k,j} N_{i+j,p-k} \quad (2.35)$$

$$a_{0,0} = 1$$

$$a_{k,0} = \frac{a_{k-1,0}}{u_{i+p-k+1} - u_i}$$

$$a_{k,j} = \frac{a_{k-1,j} - a_{k-1,j-1}}{u_{i+p+j-k+1} - u_{i+j}} \quad j = 1, \dots, k-1$$

$$a_{k,k} = \frac{-a_{k-1,k-1}}{u_{i+p+1} - u_{i+k}}$$

Ainda, Butterfield (1976) fornece uma equação para o computo das derivadas das funções de base B-spline, dada por

$$N_{i,p}^{(k)}(u) = \frac{p}{p-k} \left(\frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}^{(k)} + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}^{(k)} \right) \quad k = 0, \dots, p-1 \quad (2.36)$$

Sobre um vetor de nós $\mathcal{U} = \{0, 0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5, 5\}$ são definidas as nove funções de base B-spline de grau três, na figura 2.18, e suas respectivas derivadas na figura 2.19.

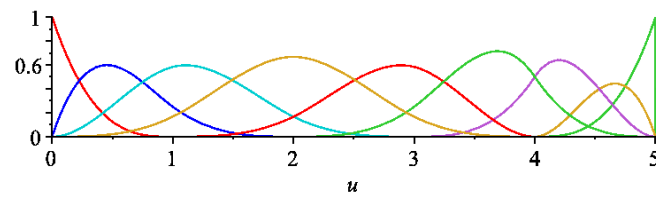


Figura 2.18: Funções de base B-*spline* de grau três.

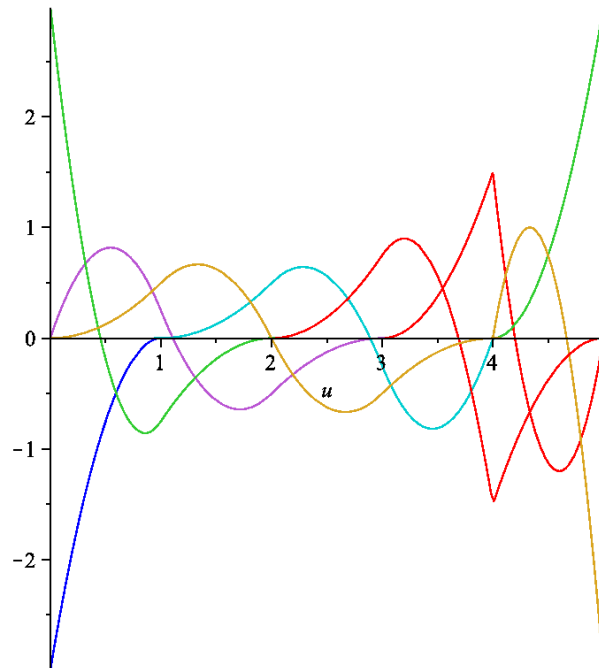


Figura 2.19: Derivadas das funções de base B-*spline* da figura 2.18.

2.7.3 Mais propriedades das funções de base

Seja $\{u_j\}$, $0 \leq j \leq k$, um conjunto de pontos de parada. O conjunto de todas as funções de base polinomiais por partes de grau p sobre $\{u_j\}$, as quais são C^{r_j} contínuas em $u = u_j$, formam um espaço vetorial, \mathcal{S} ($-1 \leq r_j \leq p$). Se nenhuma restrição de continuidade for imposta ($r_j = -1 \forall j$), a dimensão de \mathcal{S} (denotada por $\dim(\mathcal{S})$) é igual à $k(p+1)$. Cada restrição de continuidade que for imposta, reduz a dimensão em uma unidade, logo

$$\dim(\mathcal{S}) = k(p+1) - \sum_{j=0}^k (r_j + 1) \quad (2.37)$$

Pela propriedade P.B-*spline*. 5, conseguem-se as funções de base B-*spline* de grau p sobre $\{u_j\}$, com a continuidade desejada, pela apropriada escolha da multiplicidade dos nós s_j ,

onde $s_j = p - r_j$. Logo, o vetor de nós toma a seguinte forma

$$\mathcal{U} = \left\{ \underbrace{u_0, \dots, u_0}_{s_0}, \underbrace{u_1, \dots, u_1}_{s_1}, \dots, \underbrace{u_k, \dots, u_k}_{s_k} \right\}$$

Logo

$$m = \left(\sum_{j=0}^k s_j \right) - 1$$

Observar que existem m funções de grau zero, $N_{i,0}$, $m - 1$ funções de primeiro grau, $N_{i,1}$, e em geral, $m - p$ funções de grau p , $N_{i,p}$, as quais devem ter a continuidade desejada, $r_j = p - s_j$. Logo, $N_{i,p} \subset \mathcal{S}$. Substituindo $s_j = p - r_j$ em (2.37) produz

$$\begin{aligned} \dim(\mathcal{S}) &= k(p+1) - \sum_{j=0}^k (p - s_j + 1) \\ &= k(p+1) - (k+1)p + \sum_{j=0}^k s_j - (k+1) \\ &= -p - 1 + \sum_{j=0}^k s_j \\ &= m - p \end{aligned}$$

Então, o número de funções de base de grau p sobre \mathcal{U} é igual a $\dim(\mathcal{S})$. Para justificar o termo funções de ‘base’ se demonstra que $N_{i,p}(u)$ são linearmente independentes (indução sobre p), i.e., estas formam uma base do espaço vetorial, \mathcal{S} . Claramente, as funções de grau zero, são linearmente independentes.

Assume-se que para o grau $(p - 1)$, com $p > 0$, as funções são linearmente independentes. Sendo $n = m - p - 1$, assume-se que

$$\sum_{i=0}^n \alpha_i N_{i,p}(u) = 0$$

Pela expressão (2.33) tem-se que

$$\begin{aligned} 0 &= \left(\sum_{i=0}^n \alpha_i N_{i,p} \right)' = \sum_{i=0}^n \alpha_i N'_{i,p} \\ &= p \sum_{i=0}^n \alpha_i \left(\frac{N_{i,p-1}}{u_{i+p} - u_i} - \frac{N_{i+1,p-1}}{u_{i+p+1} - u_{i+1}} \right) \end{aligned}$$

implicando em

$$0 = \sum_{i=0}^n \alpha_i \frac{N_{i,p-1}}{u_{i+p} - u_i} - \sum_{i=0}^n \alpha_i \frac{N_{i+1,p-1}}{u_{i+p+1} - u_{i+1}}$$

$N_{0,p-1} = N_{n+1,p-1} = 0$; logo, a expressão (2.38) resulta

$$0 = \sum_{i=1}^n \frac{\alpha_i - \alpha_{i-1}}{u_{i+p} - u_i} N_{i,p-1}$$

isto implica que $\alpha_i - \alpha_{i-1} = 0$, e como foi assumido que funções de grau $(p-1)$ são linearmente independentes para $p > 0$, tem-se que $\alpha_i = 0 \forall i$. Isto completa a demonstração.

Obviamente, uma vez fixado o grau, o vetor de nós determina completamente as funções $N_{i,p}(u)$. Neste trabalho apenas são considerados os vetores de nós *não-periódicos* (ou abertos), que tem a forma

$$\mathcal{U} = \{\underbrace{a, \dots, a}_{p+1}, u_{p+1}, \dots, u_{m-p-1}, \underbrace{b, \dots, b}_{p+1}\} \quad (2.38)$$

O primeiro e último nó possuem multiplicidade $p+1$. Para vetores de nós não-periódicos, existem propriedades adicionais.

Propriedades (Piegl e Tiller, 1997):

P.B-spline. 7 Um vetor de nós que possuem a forma

$$\mathcal{U} = \{\underbrace{0, \dots, 0}_{p+1}, \underbrace{1, \dots, 1}_{p+1}\}$$

produzem os polinômios de Bernstein de grau p ;

P.B-spline. 8 Se o número de nós $m+1$, o grau das funções base p , e o número da funções base de grau p é $n+1$, então $m = n + p + 1$.

Seja $N_{n,p}(u)$ a última função de base de grau p , e não-nula em $[u_n, u_{n+p+1})$. Então o último nó é $u_{n+p+1} = u_m$, propriedade P.B-spline. 8. Ainda, existe a classificação do vetor de nós quanto à uniformidade, i.e., um vetor de nós uniforme é aquele que possui a mesma distância entre os nós. Caso contrário este vetor de nós é não-uniforme.

2.8 Curvas e superfícies B-spline

O uso de B-spline para modelagem geométrica computacional ou CAGD foi proposta por Riesenfeld (1973). Hoje, diversos programas computacionais de engenharia fazem o uso

de B-*spline*, e talvez sejam a descrição mais usada.

2.8.1 Definição e propriedades das curvas B-*spline*

Uma curva B-*spline* de grau p é definida por

$$\mathbf{C}(u) = \sum_{i=0}^n N_{i,p}(u) \mathbf{P}_i \quad u \in [a,b] \quad (2.39)$$

onde $\{\mathbf{P}_i\}$ são os *pontos de controle*, e $\{N_{i,p}(u)\}$ são as funções base B-*spline* de grau p definidas sobre um vetor de nós não-periódicos e não-uniformes

$$\mathcal{U} = \underbrace{\{a, \dots, a\}}_{p+1}, u_{p+1}, \dots, u_{m-p-1}, \underbrace{\{b, \dots, b\}}_{p+1} \quad (2.40)$$

de $(m + 1)$ nós.

Propriedades (Piegl e Tiller, 1997):

P.B-*spline*. 9 Se $n = p$ e $\mathcal{U} = \{0, \dots, 0, 1, \dots, 1\}$, então $\mathbf{C}(u)$ é uma curva Bézier;

P.B-*spline*. 10 $\mathbf{C}(u)$ é uma curva polinomial por partes (desde que as $N_{i,p}(u)$ sejam polinomiais por partes); o grau, p , o número de pontos de controle, $n + 1$, e o número de nós, $m + 1$, estão relacionados por

$$m = n + p + 1 \quad (2.41)$$

P.B-*spline*. 11 Interpolação dos pontos de controle extremos: $\mathbf{C}(a) = \mathbf{P}_0$ e $\mathbf{C}(b) = \mathbf{P}_n$

P.B-*spline*. 12 Transformações lineares: transformações tais como rotação, translação, escala e cisalhamento aplicados à curva são realizadas aplicando tal transformação ao polígono de controle; Considerando a transformação

$$\Phi(\mathbf{r}) = A\mathbf{r} + \mathbf{v} \quad (2.42)$$

onde A é uma matriz 3×3 , \mathbf{r} um ponto no espaço Euclidiano, e \mathbf{v} um vetor, tem-se

baseado na propriedade P.B-spline. 4 a partição da unidade, i.e.,

$$\begin{aligned}\Phi(\mathbf{r}) &= \Phi\left(\sum_{i=0}^n N_{i,p}(u) \mathbf{P}_i\right) = A\left(\sum_{i=0}^n N_{i,p}(u) \mathbf{P}_i\right) + \mathbf{v} \\ &= \sum_{i=0}^n AN_{i,p}(u) \mathbf{P}_i + \sum_{i=0}^n N_{i,p}(u) \mathbf{v} \\ &= \sum_{i=0}^n N_{i,p}(u) (A\mathbf{P}_i + \mathbf{v}) = \sum_{i=0}^n N_{i,p}(u) \Phi(\mathbf{P}_i)\end{aligned}$$

P.B-spline. 13 Região convexa: a curva está contida na região convexa formada pelo polígono de controle; De fato, se $u \in [u_i, u_{i+1})$, $p \leq i \leq m-p-1$, então $\mathbf{C}(u)$ estará contida no polígono de controle $\mathbf{P}_{i-p}, \dots, \mathbf{P}_i$. Esta vem dada pelas propriedades P.B-spline. 1, P.B-spline. 2 e P.B-spline. 4;

P.B-spline. 14 Esquema de modificação local: movendo \mathbf{P}_i a curva $\mathbf{C}(u)$ modifica sua forma somente no intervalo $[u_i, u_{i+p+1})$, esta resulta da propriedade P.B-spline. 2;

P.B-spline. 15 O polígono de controle representa a aproximação linear da curva;

P.B-spline. 16 Movendo o parâmetro u , ao longo da curva, i.e., de $u = a$ até $u = b$, as funções $N_{i,p}(u)$ vão sendo “desligadas” e “ligadas”, resulta da propriedade P.B-spline. 3; ou seja, quando u passa por um nó j , é “desligada” a função $N_{j-p-1,p}$ e “ligada” a função $N_{j,p}$;

P.B-spline. 17 Nenhuma reta ou plano contém mais interseções com uma curva B-spline do que esta reta ou plano contém com o polígono de controle da curva;

P.B-spline. 18 A continuidade e a diferenciabilidade da curva $\mathbf{C}(u)$ seguem das funções de base $N_{i,p}(u)$, já que $\mathbf{C}(u)$ é uma combinação linear das $N_{i,p}(u)$. Logo, $\mathbf{C}(u)$ é infinitamente diferenciável no interior do intervalo de nós, e é C^{p-k} contínua sobre o nó de multiplicidade k ;

P.B-spline. 19 É possível e às vezes vantajoso usar múltiplos (coincidentes) pontos de controle; pela propriedade P.B-spline. 13, a curva $\mathbf{C}(u)$ inevitavelmente irá passar pelos pontos de controle coincidentes, assim gerando uma descontinuidade visual, já que as funções de base permanecem com a mesma continuidade.

2.8.2 Derivadas das curvas B-spline

Denotando por $\mathbf{C}^{(k)}(u)$ a k -ésima derivada da curva $\mathbf{C}(u)$, fixando u , tem-se $\mathbf{C}^{(k)}(u)$ pelo computo das k -ésimas derivadas das funções de base.

$$\mathbf{C}^{(k)}(u) = \sum_{i=0}^n N_{i,p}^{(k)}(u) \mathbf{P}_i \quad (2.43)$$

Observar que para $k > p$ as derivadas valem zero; o mesmo não acontece para curvas e superfícies racionais. Diferenciando a curva B-spline de grau p ,

$$\mathbf{C}(u) = \sum_{i=0}^n N_{i,p}(u) \mathbf{P}_i$$

definida sobre um vetor de nós

$$\mathcal{U} = \{\underbrace{0, \dots, 0}_{p+1}, u_{p+1}, \dots, u_{m-p-1}, \underbrace{1, \dots, 1}_{p+1}\}$$

das expressões (2.43) e (2.33) tem-se que

$$\begin{aligned} \mathbf{C}'(u) &= \sum_{i=0}^n N'_{i,p}(u) \mathbf{P}_i \\ &= \sum_{i=0}^n \left(\frac{p}{u_{i+p} - u_i} N_{i,p-1}(u) - \frac{p}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \right) \mathbf{P}_i \\ &= \left(p \sum_{i=-1}^{n-1} N_{i+1,p-1}(u) \frac{\mathbf{P}_{i+1}}{u_{i+p+1} - u_{i+1}} \right) - \left(p \sum_{i=0}^n N_{i+1,p-1}(u) \frac{\mathbf{P}_i}{u_{i+p+1} - u_{i+1}} \right) \\ &= p \frac{N_{0,p-1}(u) \mathbf{P}_0}{u_p - u_0} + p \sum_{i=0}^{n-1} N_{i+1,p-1}(u) \frac{\mathbf{P}_{i+1} - \mathbf{P}_i}{u_{i+p+1} - u_{i+1}} - p \frac{N_{n+1,p-1}(u) \mathbf{P}_n}{u_{n+p+1} - u_{n+1}} \end{aligned}$$

O primeiro e último termo possuem denominador zero, por definição este quociente resulta zero. Logo

$$\mathbf{C}'(u) = p \sum_{i=0}^{n-1} N_{i+1,p-1}(u) \frac{\mathbf{P}_{i+1} - \mathbf{P}_i}{u_{i+p+1} - u_{i+1}} = \sum_{i=0}^{n-1} N_{i+1,p-1}(u) \mathbf{Q}_i$$

onde

$$\mathbf{Q}_i = p \frac{\mathbf{P}_{i+1} - \mathbf{P}_i}{u_{i+p+1} - u_{i+1}} \quad (2.44)$$

Agora, sendo \mathcal{U}' o vetor de nós obtido pela subtração do nó inicial e final de \mathcal{U} , tem-se

$$\mathcal{U}' = \{\underbrace{0, \dots, 0}_p, u_{p+1}, \dots, u_{m-p-1}, \underbrace{1, \dots, 1}_p\} \quad (2.45)$$

\mathcal{U}' possui $m - 1$ nós. Observar que as funções $N_{i+1,p-1}(u)$ computadas sobre \mathcal{U} , produzem o mesmo resultado que as funções $N_{i,p-1}(u)$ computadas sobre \mathcal{U}' . Então

$$\mathbf{C}'(u) = \sum_{i=0}^{n-1} N_{i,p-1}(u) \mathbf{Q}_i \quad (2.46)$$

onde os \mathbf{Q}_i são definidos pela expressão (2.44), e as $N_{i,p-1}(u)$ são computadas sobre \mathcal{U}' . Logo, $\mathbf{C}'(u)$ é uma curva B-spline de grau $p - 1$. Já que $\mathbf{C}'(u)$ é uma curva B-spline, aplicando as expressões (2.46) e (2.44) recursivamente se obtém as derivadas de mais alta ordem. Deixando $\mathbf{P}_i^{(0)} = \mathbf{P}_i$, tem-se

$$\mathbf{C}(u) = \mathbf{C}^{(0)}(u) = \sum_{i=0}^n N_{i,p}(u) \mathbf{P}_i^{(0)}$$

Logo

$$\mathbf{C}^{(k)}(u) = \sum_{i=0}^{n-k} N_{i,p-k}(u) \mathbf{P}_i^{(k)} \quad (2.47)$$

onde

$$\mathbf{P}_i^{(k)} = \begin{cases} \mathbf{P}_i & k = 0 \\ \frac{p - k + 1}{u_{i+p+1} - u_{i+1}} \left(\mathbf{P}_{i+1}^{(k-1)} - \mathbf{P}_i^{(k-1)} \right) & k > 0 \end{cases}$$

2.8.3 Definição e propriedades das superfícies B-spline

Uma superfícies B-spline é obtida a partir de uma rede bidirecional de pontos de controle, dois vetores de nós, e o produto das funções de base B-spline univariantes. Assim sendo, tem-se

$$\mathbf{S}(u,v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) \mathbf{P}_{i,j} \quad (u,v) \in [a,b] \times [c,d] \quad (2.48)$$

com

$$\begin{aligned} \mathcal{U} &= \left\{ \underbrace{a, \dots, a}_{p+1}, u_{p+1}, \dots, u_{r-p-1}, \underbrace{b, \dots, b}_{p+1} \right\} \\ \mathcal{V} &= \left\{ \underbrace{c, \dots, c}_{q+1}, v_{q+1}, \dots, v_{s-q-1}, \underbrace{d, \dots, d}_{q+1} \right\} \end{aligned}$$

\mathcal{U} possui $r + 1$ nós, \mathcal{V} possui $s + 1$ nós. A expressão (2.41), toma a seguinte forma

$$r = n + p + 1 \quad \wedge \quad s = m + q + 1 \quad (2.49)$$

Ainda, a expressão (2.48) pode ser escrita na forma matricial

$$\mathbf{S}(u,v) = [N_{k,p}(u)]^T [\mathbf{P}_{k,l}] [N_{l,q}(v)] \quad k \in [i - p, i], l \in [j - q, j] \quad (2.50)$$

Observar que $[N_{k,p}(u)]^T$ é um vetor de dimensão $(p+1)$, $[\mathbf{P}_{k,l}]$ é uma matriz de dimensão $(p+1) \times (q+1)$, e $[N_{l,q}(v)]$ é um vetor de dimensão $(q+1)$. As propriedades do produto tensorial das funções de base seguem das correspondentes propriedades univariantes das funções de base.

Propriedades (Piegl e Tiller, 1997):

P.B-spline. 20 *Função positiva semi-definida:* $N_{i,p}(u) N_{j,q}(v) \geq 0, \forall i, j, p, q, u, v;$

P.B-spline. 21 *Partição da unidade:* $\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) = 1, \forall (u,v) \in [0,1] \times [0,1];$

P.B-spline. 22 *Se $n = p$ e $m = q$, $\mathcal{U} = \{0, \dots, 0, 1, \dots, 1\}$ e $\mathcal{V} = \{0, \dots, 0, 1, \dots, 1\}$, logo $N_{i,p}(u) N_{j,q}(v) = B_{i,n}(u) B_{j,m}(v), \forall i, j;$ i.e., este produto de B-spline se degenera no produto dos polinômios de Bernstein;*

P.B-spline. 23 $N_{i,p}(u) N_{j,q}(v) = 0$ se (u,v) não pertence ao subdomínio $[u_i, u_{i+p+1}] \times [v_j, v_{j+q+1});$

P.B-spline. 24 *Em um dado período $[u_{i_0}, u_{i_0+1}] \times [v_{j_0}, v_{j_0+1})$, no máximo $(p+1)(q+1)$ funções de base são não-nulas, em particular $N_{i,p}(u) N_{j,q}(v), \forall i \in [i_0 - p, i_0] \wedge j \in [j_0 - q, j_0];$*

P.B-spline. 25 *Se $p > 0$ e $q > 0$, logo $N_{i,p}(u) N_{j,q}(v)$ possui um e somente um máximo;*

P.B-spline. 26 *No domínio retangular formado pela linha de nós u e v , onde a função é um polinômio bivalente, todas as derivadas parciais de $N_{i,p}(u)$ e $N_{j,q}(v)$ existem; sobre o nó u e v a função é C^{p-k} e C^{q-k} contínua na direção u e v , respectivamente; onde k é a multiplicidade do nó;*

Superfícies B-spline possuem as seguintes propriedades:

P.B-spline. 27 Se $n = p$ e $m = q$, $\mathcal{U} = \{0, \dots, 0, 1, \dots, 1\}$ e $\mathcal{V} = \{0, \dots, 0, 1, \dots, 1\}$, logo $\mathbf{S}(u, v)$ é uma superfície Bézier;

P.B-spline. 28 Interpolação dos pontos de controle extremos: $\mathbf{S}(0, 0) = \mathbf{P}_{0,0}$, $\mathbf{S}(1, 0) = \mathbf{P}_{1,0}$, $\mathbf{S}(0, 1) = \mathbf{P}_{0,1}$ e $\mathbf{S}(1, 1) = \mathbf{P}_{1,1}$;

P.B-spline. 29 Transformações lineares: transformações tais como rotação, translação, escala e cisalhamento aplicados à superfície são realizadas aplicando tal transformação ao polígono de controle;

P.B-spline. 30 Região convexa: a superfície está contida na região convexa formada pelo polígono de controle; De fato, se $(u, v) \in [u_{i_0}, u_{i_0+1}) \times [v_{j_0}, v_{j_0+1})$, então $\mathbf{S}(u, v)$ estará contida na região formada pelos pontos de controle $\mathbf{P}_{i,j}, \forall i \in [i_0 - p, i_0] \wedge j \in [j_0 - q, j_0]$; Esta vem dada pelas propriedades P.B-spline. 20, P.B-spline. 21 e P.B-spline. 24;

P.B-spline. 31 Se triangularizado, a rede de controle forma uma aproximação plana da superfície;

P.B-spline. 32 Esquema de modificação local: movendo $\mathbf{P}_{i,j}$ a superfície $\mathbf{S}(u, v)$ modifica sua forma somente no intervalo $[u_i, u_{i+p+1}) \times [v_j, v_{j+q+1})$, esta resulta da propriedade P.B-spline. 2;

P.B-spline. 33 A continuidade e a diferenciabilidade da superfície $\mathbf{S}(u, v)$ seguem das funções de base; em particular $\mathbf{S}(u, v)$ é infinitamente diferenciável no interior do intervalo de nós, e é C^{p-k} e C^{q-k} contínua na direção u e v , respectivamente; onde k é a multiplicidade do nó;

2.8.4 Derivadas das superfícies B-spline

Fixando (u, v) , interessa computar todas as derivadas parciais de $\mathbf{S}(u, v)$ até uma ordem d (incluindo d), dadas por

$$\frac{\partial^{k+l}}{\partial^k u \partial^l v} \mathbf{S}(u, v), \quad \forall (k+l) \in [0, d] \quad (2.51)$$

Assim como para as curvas, estas derivadas são obtidas a partir das derivadas das funções de base. Logo

$$\frac{\partial^{k+l}}{\partial^k u \partial^l v} \mathbf{S}(u,v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}^{(k)}(u) N_{j,q}^{(l)}(v) \mathbf{P}_{i,j} \quad (2.52)$$

$$\frac{\partial^{k+l}}{\partial^k u \partial^l v} \mathbf{S}(u,v) = [N_{r,p}^{(k)}(u)]^T [\mathbf{P}_{r,s}] [N_{s,q}^{(l)}(v)] \quad r \in [i-p, i], s \in [j-q, j] \\ (k+l) \in [0, d] \quad (2.53)$$

Assim como para curvas, se definiram recursivamente todas as derivadas em (2.47), de forma análoga tem-se para as superfícies

$$\frac{\partial^{k+l}}{\partial^k u \partial^l v} \mathbf{S}(u,v) = \sum_{i=0}^{n-k} \sum_{j=0}^{m-l} N_{i,p-k}(u) N_{j,q-l}(v) \mathbf{P}_{i,j}^{(k,l)} \quad (2.54) \\ \mathbf{P}_{i,j}^{(k,l)} = (q-l+1) \frac{\mathbf{P}_{i,j+1}^{(k,l-1)} - \mathbf{P}_{i,j}^{(k,l-1)}}{v_{j+q+1} - v_{j+l}}$$

2.9 B-spline racionais não-uniformes – NURBS –

Nesta seção são combinados os conceitos das seções 2.4 e 2.8, i.e., criar curvas e superfícies B-spline no espaço homogêneo, para que estas duas entidades geométricas sejam mapeadas para o espaço Euclidiano e assim torná-las racionais. Deste modo, são criadas B-spline Não-Uniformes Racionais (e ainda não-periódicas), i.e., NURBS do acrônimo em inglês, *NonUniform Rational B-Spline*.

O fato de se optar por trabalhar com vetores de nós não-periódicos, é devido a que esta característica torna as entidades geométricas interpoladas nos pontos de controle extremos, i.e., tal entidade é “obrigada” a passar pelos pontos de controle extremos, P.B-spline. 28 e P.B-spline. 11.

Para o vetor de nós periódicos, os pontos de controle não representam de maneira intuitiva a entidade. A vantagem de trabalhar com nós não-periódicos é que tal torna as entidades mais “maleáveis”, já que pela posição dos pontos de controle, de maneira intuitiva, se sabe como tal entidade irá aparentar. Todos os algoritmos criados neste trabalho não discriminam o vetor de nós, entre periódicos ou não, para que não se perda a generalidade, entretanto tais algoritmos só foram utilizados para vetor de nós não-periódicos.

2.9.1 Definição e propriedades das curvas NURBS

Uma curva NURBS de grau p é definida por

$$\mathbf{C}(u) = \frac{\sum_{i=0}^n N_{i,p}(u) w_i \mathbf{P}_i}{\sum_{i=0}^n N_{i,p}(u) w_i} \quad u \in [a, b] \quad (2.55)$$

onde $\{\mathbf{P}_i\}$ são os *pontos de controle*, $\{w_i\}$ são os *pesos* e as $\{N_{i,p}(u)\}$ são as funções base B-spline de grau p definidas sobre um vetor de nós não-periódicos e não-uniformes

$$\mathcal{U} = \{\underbrace{a, \dots, a}_{p+1}, u_{p+1}, \dots, u_{m-p-1}, \underbrace{b, \dots, b}_{p+1}\} \quad (2.56)$$

de $(m + 1)$ nós.

Assume-se, sem perder generalidade, que $a = 0$, $b = 1$, e $w_i > 0 \forall i$. Logo, fazendo

$$R_{i,p}(u) = \frac{N_{i,p}(u) w_i}{\sum_{i=0}^n N_{i,p}(u) w_i} \quad (2.57)$$

permite reescrever (2.55) na forma

$$\mathbf{C}(u) = \sum_{i=0}^n R_{i,p}(u) \mathbf{P}_i \quad (2.58)$$

As $\{R_{i,p}(u)\}$ são as *funções base racionais*; elas são funções racionais por partes sobre $u \in [0,1]$. As propriedades das $R_{i,p}(u)$ são derivadas de (2.57) e das correspondentes propriedades das $N_{i,p}(u)$.

Propriedades (Piegl e Tiller, 1997):

P.NURBS. 1 *Função positiva semi-definida:* $R_{i,p}(u) \geq 0, \forall i, p$ e $u \in [0,1]$;

P.NURBS. 2 *Partição da unidade:* $\sum_{i=0}^n R_{i,p}(u) = 1, \forall u \in [0,1]$;

P.NURBS. 3 $R_{0,p}(0) = R_{n,p}(1) = 1$;

P.NURBS. 4 *Toda função possui no mínimo dois pontos extremos, i.e., ao menos possui um mínimo e um máximo (teorema de Weierstrass); Toda função NURBS $R_{i,p}(u)$ para $p > 0$ possui um mínimo e um máximo no intervalo $u \in [0,1]$;*

P.NURBS. 5 *Suporte local:* $R_{i,p}(u) = 0$ para $u \notin [u_i, u_{i+p+1})$. Ainda, em um dado período $[u_i, u_{i+1})$ no máximo $p+1$ das $R_{i,p}(u)$ são não-nulas, chamadas de $R_{i-p,p}, \dots, R_{i,p}$;

P.NURBS. 6 *Todas as derivadas de $R_{i,p}(u)$ existem no interior do período, estas são funções racionais de denominador não-nulo. Sobre o nó, $R_{i,p}(u)$ é C^{p-k} contínuo, onde k é a multiplicidade do nó;*

P.NURBS. 7 *Se $w_i = 1$ para todo i , então $R_{i,p}(u) = N_{i,p}(u)$ para todo i ; i.e., $N_{i,p}(u)$ é um caso particular de $R_{i,p}(u)$;*

Sobre um vetor de nós $\mathcal{U} = \{0, 0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5, 5\}$, definido na seção 2.7.2 e nas figuras 2.18 e 2.19, são definidas as funções de base NURBS de grau três, com o vetor de pesos definidos por $w_{(a)i} = \{1, 1, 1, 1, 10, 1, 1, 1, 1\}$ na figura 2.20, com vetor de pesos definidos por $w_{(b)i} = \{1, 1, 0.1, 0.1, 10, 1, 1, 1, 1\}$ na figura 2.21, e com vetor de pesos definidos por $w_{(c)i} = \{0.1, 0.4, 0.1, 0.7, 3, 0.5, 10, 4, 1\}$ na figura 2.22. Nestas figuras claramente percebe-se o rol dos pesos.

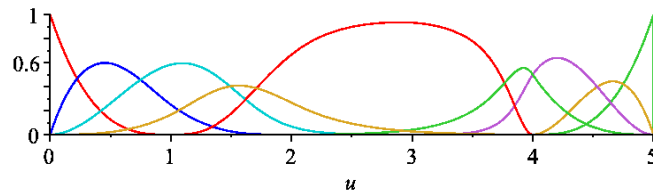


Figura 2.20: Funções de base NURBS, $p = 3$, para os $w_{(a)i}$.

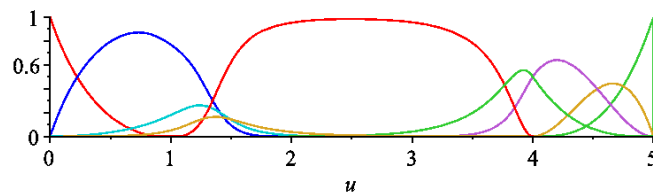


Figura 2.21: Funções de base NURBS, $p = 3$, para os $w_{(b)i}$.

As propriedades de P.NURBS. 1 a P.NURBS. 7, constituem importantes características geométricas de curvas NURBS.

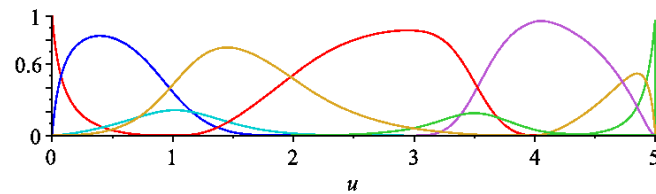


Figura 2.22: Funções de base NURBS, $p = 3$, para os $w_{(c)_i}$.

P.NURBS. 8 $\mathbf{C}(0) = \mathbf{P}_0$ e $\mathbf{C}(1) = \mathbf{P}_1$ de P.NURBS. 3;

P.NURBS. 9 *Transformações lineares: transformações tais como rotação, translação, escala e cisalhamento aplicados à curva são realizadas aplicando tal transformação ao polígono de controle;*

P.NURBS. 10 *Região convexa: a curva está contida na região convexa formada pelo polígono de controle; De fato, se $u \in [u_i, u_{i+1})$, $p \leq i \leq m - p - 1$, então $\mathbf{C}(u)$ estará contida no polígono de controle $\mathbf{P}_{i-p}, \dots, \mathbf{P}_i$;*

P.NURBS. 11 $\mathbf{C}(u)$ é infinitamente diferenciável no interior do intervalo de nós, e é C^{p-k} contínua sobre o nó de multiplicidade k ;

P.NURBS. 12 *Nenhuma reta ou plano contêm mais interseções com uma curva NURBS do que esta reta ou plano contêm com o polígono de controle da curva;*

P.NURBS. 13 *Uma curva NURBS sem nós internos é uma curva Bézier racional, já que as $N_{i,p}(u)$ se convertem nas $B_{i,n}(u)$;*

P.NURBS. 14 *Esquema de modificação local: modificando \mathbf{P}_i ou w_i a curva $\mathbf{C}(u)$ modifica sua forma somente no intervalo $[u_i, u_{i+p+1})$, esta resulta da propriedade P.NURBS. 5;*

Resumidamente a propriedade P.NURBS. 14, gera a motivação para todos os tipos de algoritmos de modificação de forma baseada em NURBS. Já que a partir desta se pode modificar o conjunto \mathbf{P}_i e w_i sem alterar toda a entidade, mas apenas um subdomínio.

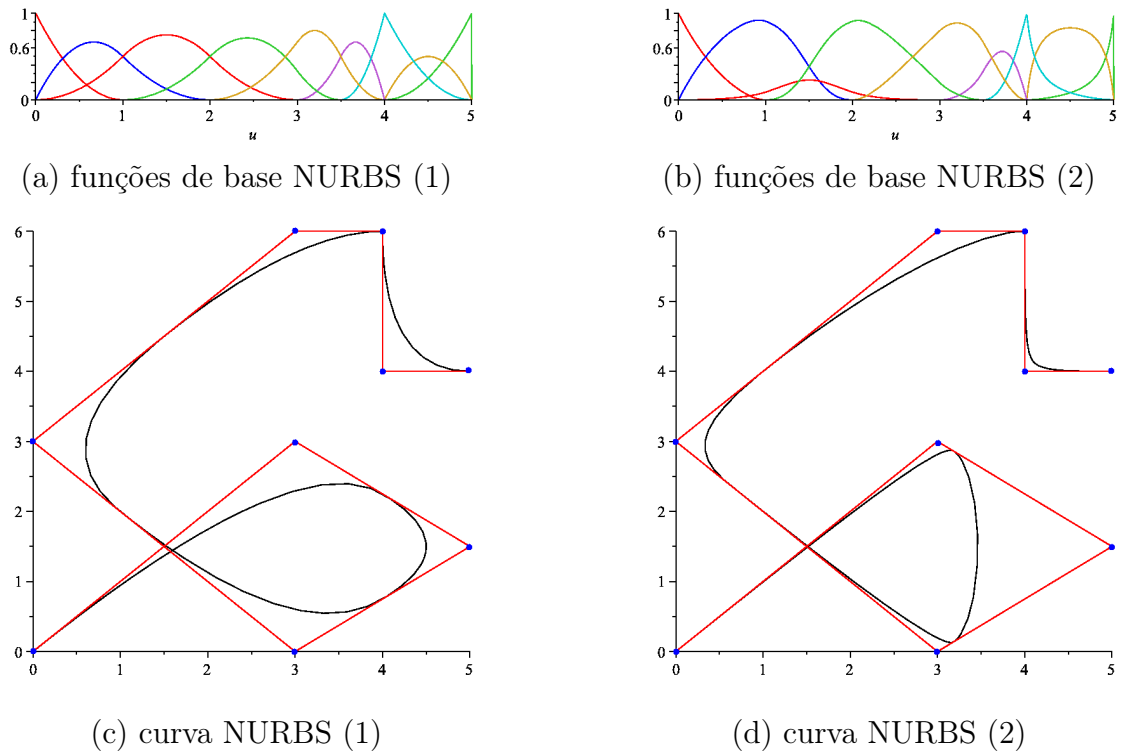


Figura 2.23: Curvas NURBS com diferentes pesos.

As curvas das figuras 2.23 estão baseada no vetor de nós $\mathcal{U} = \{0, 0, 0, 1, 2, 3, 3.5, 4, 4, 5, 5, 5\}$ e são de grau dois. Entretanto estas diferem por possuir pesos distintos. Para a curva da figura 2.23c, tem-se $w_i = \{1, 1, 1, 1, 1, 1, 1, 1, 1\}$, i.e., esta é uma *B-spline*. Já para a curva da figura 2.23d, tem-se $w_i = \{1, 1, 0.1, 1, 2, 1, 1, 5, 1\}$.

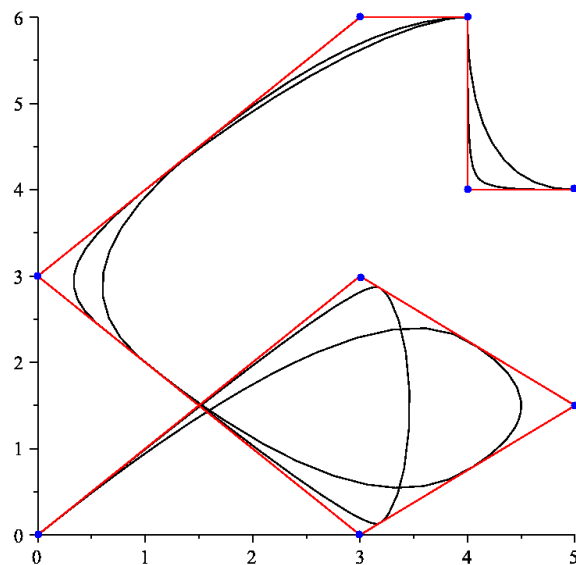


Figura 2.24: Curvas NURBS sobrepostas com diferentes pesos.

Assim como para o caso das curvas Bézier, as coordenadas homogêneas oferecem um método eficiente para representar curvas NURBS. Sendo \mathcal{H} o mapeamento dado por (2.10), se consegue construir, a partir do conjunto de pontos de controle, $\{\mathbf{P}_i\}$, e pesos, $\{w_i\}$, os pontos de controle no espaço homogêneo, $\mathbf{P}_i^w = (w_i x_i, w_i y_i, w_i z_i, w_i)$. Logo, definindo os polinômios por partes B-spline não-rationais no espaço homogêneo como

$$\mathbf{C}^w(u) = \sum_{i=0}^n N_{i,p}(u) \mathbf{P}_i^w \quad (2.59)$$

Aplicando o mapeamento, \mathcal{H} , à curva $\mathbf{C}^w(u)$ produz a correspondente curva B-spline racional no espaço Euclidiano.

$$\mathbf{C}(u) = \mathcal{H}\{\mathbf{C}^w(u)\} = \mathcal{H}\left\{\sum_{i=0}^n N_{i,p}(u) \mathbf{P}_i^w\right\} \quad (2.60)$$

2.9.2 Derivadas das curvas NURBS

Derivadas de funções racionais são mais trabalhosas de serem obtidas, envolvendo denominadores de altas potências, devido ao teorema da derivada do quociente. Os algoritmos desenvolvidos para as curvas B-spline não-rationais, podem ser aplicados, obviamente, às curvas $\mathbf{C}^w(u)$, já que estas são não-rationais no espaço homogêneo. Agora, a partir das derivadas de $\mathbf{C}^w(u)$ são obtidas as derivadas de $\mathbf{C}(u)$. Seja

$$\mathbf{C}(u) = \frac{w(u) \mathbf{C}^w(u)}{w(u)} = \frac{\mathbf{A}(u)}{w(u)}$$

onde $\mathbf{A}(u)$ é um vetor contendo as três primeiras coordenadas de $\mathbf{C}^w(u)$ ($\mathbf{A}(u)$ é o numerador de (2.55)). Logo

$$\begin{aligned} \mathbf{C}'(u) &= \frac{w(u) \mathbf{A}'(u) - w'(u) \mathbf{A}(u)}{w(u)^2} \\ &= \frac{w(u) \mathbf{A}'(u) - w'(u) w(u) \mathbf{C}(u)}{w(u)^2} = \frac{\mathbf{A}'(u) - w'(u) \mathbf{C}(u)}{w(u)} \end{aligned} \quad (2.61)$$

Já que $\mathbf{A}(u)$ e $w(u)$ representam as coordenadas de $\mathbf{C}^w(u)$, as derivadas de primeira ordem são obtidas usando (2.44) e (2.46). O computo de derivadas de alta ordem são obtidas diferenciando $\mathbf{A}(u)$ usando o teorema de Leibnitz

$$\begin{aligned} \mathbf{A}^{(k)}(u) &= (w(u) \mathbf{C}(u))^{(k)} = \sum_{i=0}^k \binom{k}{i} w^{(i)}(u) \mathbf{C}^{(k-i)}(u) \\ &= w(u) \mathbf{C}^{(k)}(u) + \sum_{i=1}^k \binom{k}{i} w^{(i)}(u) \mathbf{C}^{(k-i)}(u) \end{aligned} \quad (2.62)$$

desta tem-se

$$\mathbf{C}^{(k)}(u) = \frac{\mathbf{A}^{(k)}(u) - \sum_{i=1}^k \binom{k}{i} w^{(i)}(u) \mathbf{C}^{(k-i)}(u)}{w(u)} \quad (2.63)$$

Esta expressão (2.63) fornece a k -ésima derivada de $\mathbf{C}(u)$ em termos da k -ésima derivada de $\mathbf{A}(u)$ e das $(k-1)$ derivadas anteriores de $\mathbf{C}(u)$ e $w(u)$. As derivadas $\mathbf{A}^{(k)}(u)$ e $w^{(i)}(u)$ são obtidas usando (2.43) ou (2.47).

As derivadas das funções de base das figuras 2.20, 2.21 e 2.22 são apresentadas nas figuras 2.25, 2.26 e 2.27.

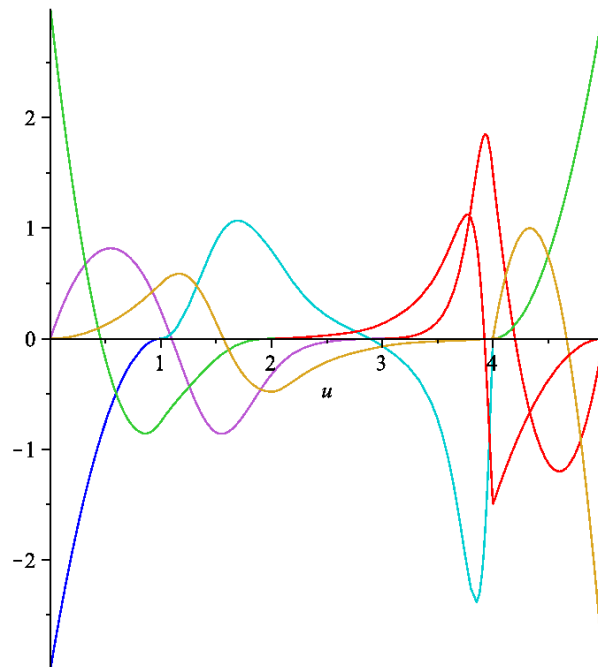


Figura 2.25: Derivadas das funções de base NURBS da figura 2.20.

2.9.3 Definição e propriedades das superfícies NURBS

Uma superfície NURBS de grau p na direção u e de grau q na direção v é uma função vetorial bivalente racional por partes da forma

$$\mathbf{S}(u,v) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{i,j} \mathbf{P}_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{i,j}} \quad (u,v) \in [0,1] \times [0,1] \quad (2.64)$$

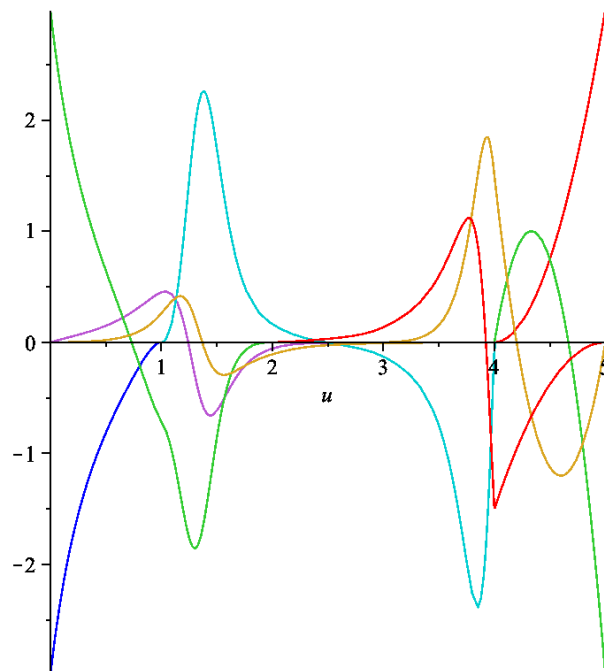


Figura 2.26: Derivadas das funções de base NURBSvda figura 2.21.

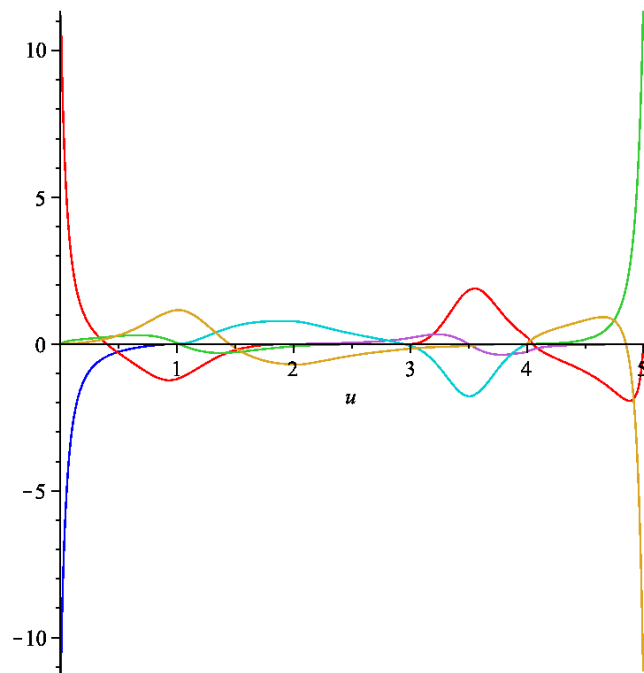


Figura 2.27: Derivadas das funções de base NURBSvda figura 2.22.

Os $\{\mathbf{P}_{i,j}\}$ formam uma rede de controle bidirecional, os $\{w_{i,j}\}$ são os pesos e as $\{N_{i,p}(u)\}$ e $\{N_{j,q}(v)\}$ são as funções de base B-spline não-rationais definidas sobre o vetor de nós

$$\mathcal{U} = \underbrace{\{0, \dots, 0\}}_{p+1}, u_{p+1}, \dots, u_{r-p-1}, \underbrace{\{1, \dots, 1\}}_{p+1}$$

$$\mathcal{V} = \underbrace{\{0, \dots, 0\}}_{q+1}, v_{q+1}, \dots, v_{s-q-1}, \underbrace{\{1, \dots, 1\}}_{q+1}$$

onde $r = n + p + 1$ e $s = m + q + 1$.

Introduzindo as funções de base racionais

$$R_{i,j}(u,v) = \frac{N_{i,p}(u) N_{j,q}(v) w_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) w_{i,j}} \quad (u,v) \in [0,1] \times [0,1] \quad (2.65)$$

a superfície (2.64) pode ser reescrita como

$$\mathbf{S}(u,v) = \sum_{i=0}^n \sum_{j=0}^m R_{i,j}(u,v) \mathbf{P}_{i,j} \quad (2.66)$$

Algumas superfícies NURBS com suas respectivas redes de controle são apresentadas nas figuras 2.28, 2.29, 2.30, 2.31 e 2.32.

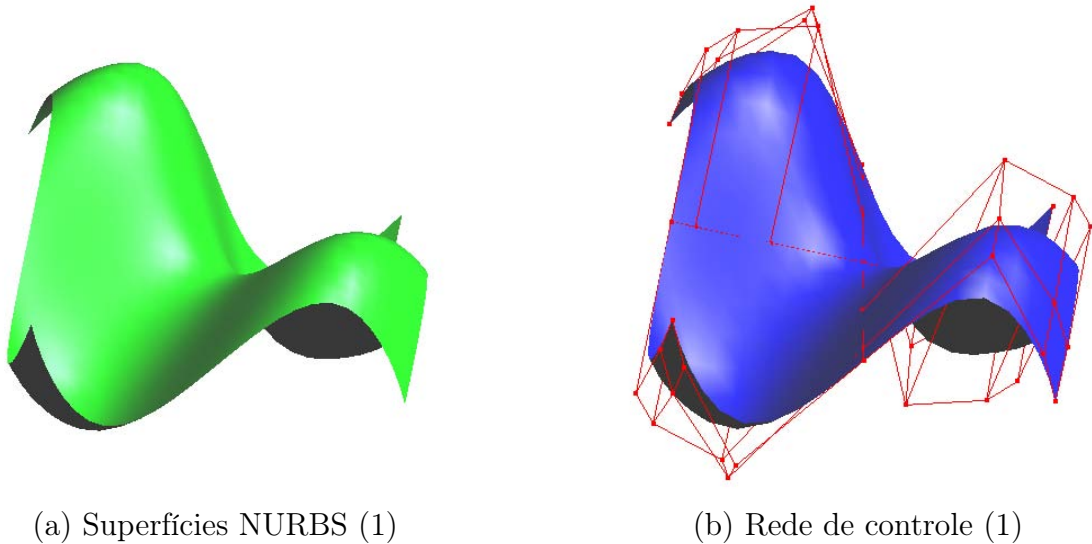


Figura 2.28: Superfícies NURBS com a rede de controle (1).

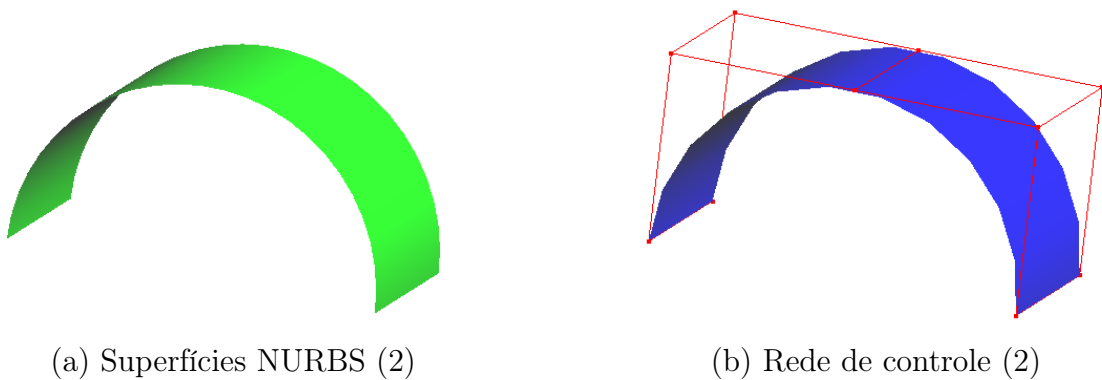


Figura 2.29: Superfícies NURBS com a rede de controle (2).

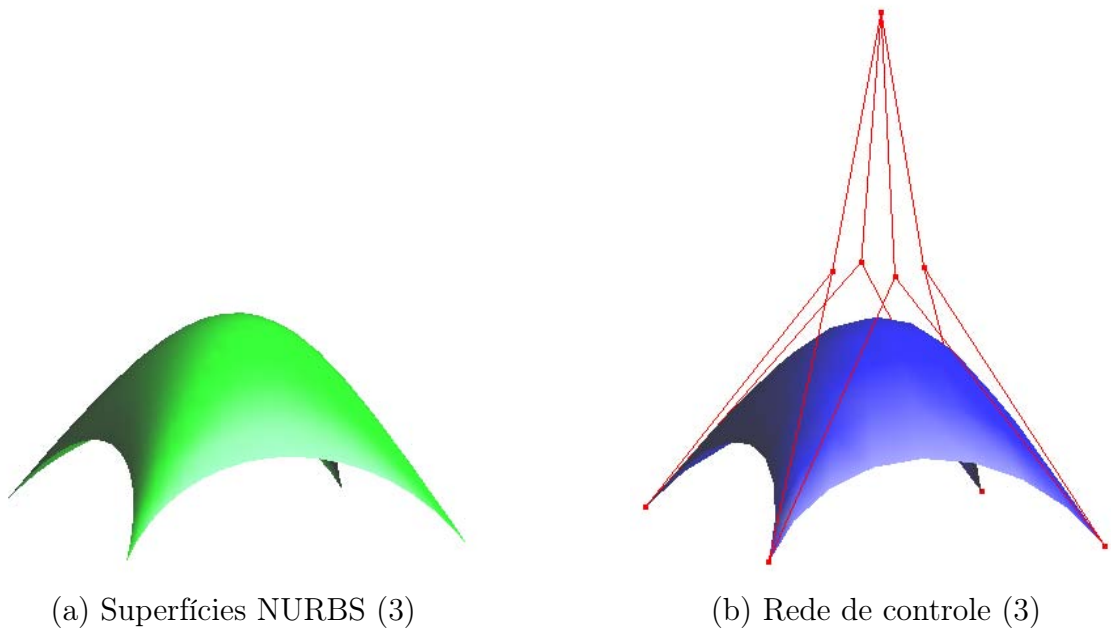


Figura 2.30: Superfícies NURBS com a rede de controle (3).

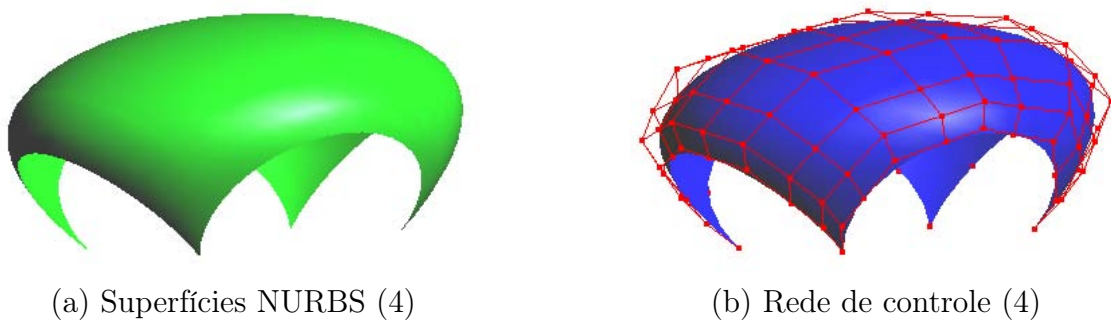


Figura 2.31: Superfícies NURBS com a rede de controle (4).

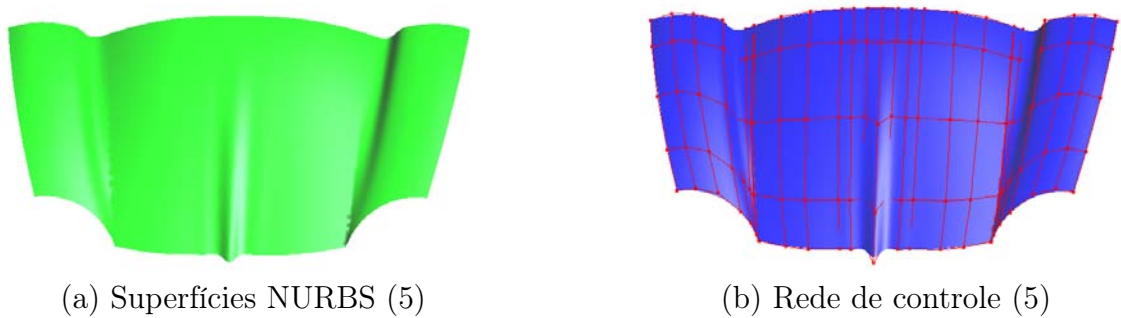


Figura 2.32: Superfícies NURBS com a rede de controle (5).

Propriedades (Piegl e Tiller, 1997):

P.NURBS. 15 *Função positiva semi-definida:* $R_{i,j}(u,v) \geq 0$, $\forall i, j, u, v$ e $\forall (u,v) \in [0,1] \times [0,1]$;

P.NURBS. 16 *Partição da unidade:* $\sum_{i=0}^n \sum_{j=0}^m R_{i,j}(u,v) = 1$, $\forall (u,v) \in [0,1] \times [0,1]$;

P.NURBS. 17 *Suporte local:* $R_{i,j}(u,v) = 0$ para $(u,v) \notin [u_i, u_{i+p+1}] \times [v_j, v_{j+q+1}]$;

P.NURBS. 18 *Em um dado período $[u_{i_0}, u_{i_0+p+1}] \times [v_{j_0}, v_{j_0+q+1}]$, no máximo $(p+1)(q+1)$ funções de base são não-nulas, em particular $R_{i,j}(u,v)$ para $(i,j) \in [i_0 - p, i_0] \times [j_0 - q, j_0]$;*

P.NURBS. 19 *Toda função possui no mínimo dois pontos extremos, i.e., ao menos possui um mínimo e um máximo (teorema de Weierstrass); Toda função NURBS $R_{i,j}(u,v)$ para $p > 0$ e $q > 0$ possui um mínimo e um máximo no intervalo $(u,v) \in [0,1] \times [0,1]$;*

P.NURBS. 20 $R_{0,0} = R_{n,0} = R_{0,m} = R_{n,m} = 1$;

P.NURBS. 21 *Todas as derivadas parciais de $R_{i,j}(u,v)$ existem no interior do período, estas são funções racionais de denominador não-nulo. Sobre o nó, $R_{i,j}(u,v)$ é C^{p-k} e C^{q-k} contínuo, na direção u e v , respectivamente, onde k é a multiplicidade do nó;*

P.NURBS. 22 *Se $w_{i,j} = a$ para todo (i,j) , então $R_{i,j}(u,v) = N_{i,p}(u) N_{j,q}(v)$ para todo (i,j) ;*

As propriedades de P.NURBS. 15 a P.NURBS. 22, constituem importantes características geométricas de superfícies NURBS.

P.NURBS. 23 *Interpolação dos pontos de controle extremos:* $\mathbf{S}(0,0) = \mathbf{P}_{0,0}$, $\mathbf{S}(1,0) = \mathbf{P}_{1,0}$, $\mathbf{S}(0,1) = \mathbf{P}_{0,1}$ e $\mathbf{S}(1,1) = \mathbf{P}_{1,1}$;

P.NURBS. 24 *Transformações lineares: transformações tais como rotação, translação, escala e cisalhamento aplicados à superfície são realizadas aplicando tal transformação ao polígono de controle;*

P.NURBS. 25 *Região convexa: se $w_{i,j} \geq 0$ para todo (i,j) , a superfície está contida na região convexa formada pelo polígono de controle; De fato, se $(u,v) \in [u_{i_0}, u_{i_0+1}) \times [v_{j_0}, v_{j_0+1})$, então $\mathbf{S}(u,v)$ estará contida na região formada pelos pontos de controle $\mathbf{P}_{i,j}, \forall i \in [i_0 - p, i_0] \wedge j \in [j_0 - q, j_0]$;*

P.NURBS. 26 *Esquema de modificação local: movendo $\mathbf{P}_{i,j}$ ou $w_{i,j}$ a superfície $\mathbf{S}(u,v)$ modifica sua forma somente no intervalo $[u_i, u_{i+p+1}) \times [v_j, v_{j+q+1})$*

P.NURBS. 27 *Bézier, B-spline não-rationais e Bézier racionais são um caso particular de superfícies NURBS;*

P.NURBS. 28 *A continuidade e a diferenciabilidade da superfície $\mathbf{S}(u,v)$ seguem das funções de base; em particular $\mathbf{S}(u,v)$ é infinitamente diferenciável no interior do intervalo de nós, e é C^{p-k} e C^{q-k} contínua na direção u e v , respectivamente; onde k é a multiplicidade do nó.*

A relação entre os conjuntos formados pelas funções de base de cada tipo de representação é apresentada na figura 2.33. Para as superfícies também resulta vantajoso fazer a

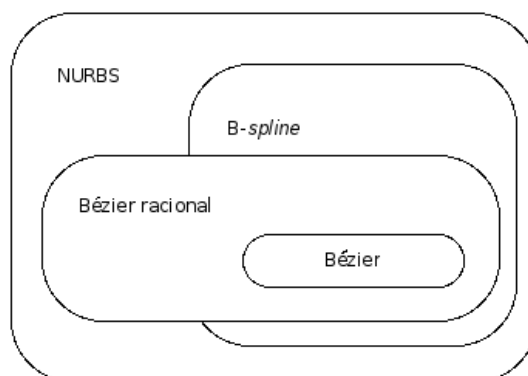


Figura 2.33: Relação entre os conjuntos.

representação no espaço homogêneo. Assim, a superfície toma a forma

$$\mathbf{S}^w(u,v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) \mathbf{P}_{i,j}^w \quad (2.67)$$

onde $\mathbf{P}_{i,j}^w = (w_{i,j}x_{i,j}, w_{i,j}y_{i,j}, w_{i,j}z_{i,j}, w_{i,j})$. Logo $\mathbf{S}(u,v) = \mathcal{H}\{\mathbf{S}^w(u,v)\}$. $\mathbf{S}^w(u,v)$ é formada por produto tensorial, sendo um polinômio por partes não-racional no espaço homogêneo. Enquanto, $\mathbf{S}(u,v)$ não é formada por produto tensorial, sendo um polinômio por partes racional no espaço Euclideano.

2.9.4 Derivadas das superfícies NURBS

As derivadas de $\mathbf{S}^w(u,v)$ são computadas usando as expressões (2.51) a (2.54) aplicadas no espaço homogêneo. A partir das derivadas parciais de $\mathbf{S}^w(u,v)$ são obtidas as derivadas parciais $\mathbf{S}(u,v)$. Sendo

$$\mathbf{S}(u,v) = \frac{w(u,v) \mathbf{S}^w(u,v)}{w(u,v)} = \frac{\mathbf{A}(u,v)}{w(u,v)} \quad (2.68)$$

tem-se

$$\mathbf{S}_\alpha(u,v) = \frac{\mathbf{A}_\alpha(u,v) - w_\alpha(u,v) \mathbf{S}(u,v)}{w(u,v)} \quad (2.69)$$

onde α denota a derivada em relação u ou v .

Em geral

$$\begin{aligned} \mathbf{A}^{(k,l)} &= \left[(w\mathbf{S})^k \right]^l = \left(\sum_{i=0}^k \binom{k}{i} w^{(i,0)} \mathbf{S}^{(k-i,0)} \right)^l \\ &= \sum_{i=0}^k \binom{k}{i} \sum_{j=0}^l \binom{l}{j} w^{(i,j)} \mathbf{S}^{(k-i,l-j)} \\ &= w^{(0,0)} \mathbf{S}^{(k,l)} + \sum_{i=1}^k \binom{k}{i} w^{(i,0)} \mathbf{S}^{(k-i,l)} + \sum_{j=1}^l \binom{l}{j} w^{(0,j)} \mathbf{S}^{(k,l-j)} \\ &\quad + \sum_{i=1}^k \binom{k}{i} \sum_{j=1}^l \binom{l}{j} w^{(i,j)} \mathbf{S}^{(k-i,l-j)} \end{aligned}$$

Desta segue que

$$\begin{aligned} \mathbf{S}^{(k,l)} &= \frac{1}{w} \left(\mathbf{A}^{(k,l)} - \sum_{i=1}^k \binom{k}{i} w^{(i,0)} \mathbf{S}^{(k-i,l)} \right. \\ &\quad \left. - \sum_{j=1}^l \binom{l}{j} w^{(0,j)} \mathbf{S}^{(k,l-j)} - \sum_{i=1}^k \binom{k}{i} \sum_{j=1}^l \binom{l}{j} w^{(i,j)} \mathbf{S}^{(k-i,l-j)} \right) \quad (2.70) \end{aligned}$$

2.10 Curvatura Gaussiana

A curvatura Gaussiana fornece uma idéia de como se desenvolvem os raios instantâneos da superfície em um determinado ponto. Precisamente esta representa o produto do inverso do raio instantâneo das direções dos eixos do espaço paramétrico.

Neste trabalho se faz uso da curvatura Gaussiana para controlar a deformação da malha de elementos finitos no processo de otimização. Devido ao fato de que a cada iteração a malha é reposicionada se deve ter controle para que os elementos não sofram uma deformação que gere erros inaceitáveis. O controle é feito pela curvatura Gaussiana.

A seguir são apresentados os conceitos necessários para o computo da curvatura Gaussiana.

2.10.1 Primeira forma fundamental

Seja ϕ um mapeamento do \mathbb{R}^2 ao \mathbb{R}^3 . Este pode ser considerado como uma parametrização de uma superfície determinada pela imagem de ϕ .

Sejam ∂_i , para $i = 1, 2$ as derivadas da parametrização em relação às variáveis de \mathbb{R}^2 . Ao produto interno entre $\langle \partial_i, \partial_j \rangle$ se denomina coeficientes da *primeira forma fundamental*, $g_{i,j}$.

A matriz $[g_{i,j}]$ é conhecida como tensor métrico.

Os coeficientes da primeira forma fundamental podem ser representados por $E = \langle \partial_1, \partial_1 \rangle$, $F = \langle \partial_1, \partial_2 \rangle$ e $G = \langle \partial_2, \partial_2 \rangle$.

2.10.2 Segunda forma fundamental

Os coeficientes da *segunda forma fundamental*, $b_{i,j}$, de uma superfície estão dados por $\langle \hat{\mathbf{n}}, \partial_{i,j} \rangle$, onde $\hat{\mathbf{n}} = \frac{\partial_1 \times \partial_2}{\|\partial_1 \times \partial_2\|}$ e $\partial_{i,j}$ são as derivadas segundas da parametrização em relação às variáveis de \mathbb{R}^2 .

Os coeficientes da segunda forma fundamental podem ser representados por $L = \langle \partial_{1,1}, \hat{\mathbf{n}} \rangle$,

$$M = \langle \partial_{1,2}, \hat{\mathbf{n}} \rangle \text{ e } N = \langle \partial_{2,2}, \hat{\mathbf{n}} \rangle.$$

2.10.3 Computo da curvatura Gaussiana

A curvatura Gaussiana de uma superfície é uma função escalar $K_G(u, v)$ que indica a curvatura de uma superfície em seus pontos.

A curvatura Gaussiana se determina mediante o quociente

$$K_G(u, v) = \frac{b_{1,1}b_{2,2} - b_{1,2}^2}{g_{1,1}g_{2,2} - g_{1,2}^2} \quad (2.71)$$

onde $g_{1,1}$, $g_{1,2}$ e $g_{2,2}$ correspondem aos coeficientes da primeira forma fundamental e $b_{1,1}$, $b_{1,2}$ e $b_{2,2}$ correspondem aos coeficientes da segunda forma fundamental.

Por exemplo, uma esfera de raio r tem-se $K_G(u, v) = \frac{1}{r^2}$ para todo (u, v) .

Outra forma de determinar a curvatura Gaussiana é mediante o produto das curvaturas principais k_1 e k_2 , que são os valores próprios do arranjo matricial $[b_{i,j}] - k[g_{i,j}] = 0$, i.e.,

$$\det \begin{bmatrix} L - kE & M - kF \\ M - kF & N - kG \end{bmatrix} = 0 \quad (2.72)$$

2.11 Modificação de forma

Uma curva ou uma superfície NURBS resulta completamente descrita ao conhecer seu vetor de nós, \mathcal{U} e \mathcal{V} , seu grau, p e q , seus pontos de controle, $\mathbf{P}_{i,j}$, e seus pesos, $w_{i,j}$. Ainda, trabalhando em coordenadas homogêneas, em vez de $\mathbf{P}_{i,j}$ e $w_{i,j}$, é necessário conhecer seus pontos de controle em coordenadas homogêneas $\mathbf{P}_{i,j}^w$.

Obviamente, modificando qualquer uma destas características que definem a entidade, esta será modificada. Entretanto o que se busca é uma modificação *natural*, *intuitiva* e suficientemente *local*. As técnicas de modificação de forma baseadas em geometrias NURBS são propostas pela comunidade da computação gráfica. Logo, tais técnicas foram adaptadas para satisfazer necessidades científicas.

Existem inúmeras técnicas de modificação de forma, independente do tipo de parametrização, entretanto nenhuma delas foi criada para otimização de forma. Usualmente na

otimização de forma, as variáveis são as coordenadas de alguns nós da malha de elementos finitos, e a partir da nova posição dos nós, por interpolação ou por aproximação é obtida a nova geometria e logo se faz a malha da nova geometria. Claramente, este processo é muito custoso computacionalmente, i.e., este é lento e ainda requer uma comunicação iterativa com um código computacional de CAGD e com outro de geração de malha.

A partir das seguintes premissas o código foi elaborado:

- Código independente de qualquer código computacional de CAGD;
- Código independente de qualquer gerador de malha;
- Código baseado em NURBS;
- Código que necessite baixa memória de armazenamento e processamento;
- Código que de alguma maneira controle a deformação sobre os elementos.

Parece difícil satisfazer tantas exigências, entretanto ao longo deste últimos vinte anos, na computação gráfica foram desenvolvidos algoritmos altamente eficientes baseados em NURBS. Em particular, Sederberg e Parry (1986), elaboraram a deformação livre de forma (*Free-form deformation*) baseada em sólidos com parametrização Bézier. A princípio a deformação livre de forma satisfaz as duas primeiras exigências, onde as outras três tiveram de ser acrescentadas.

O método de deformação livre de forma é descrito por Sederberg e Parry (1986) como se a geometria a ser deformada estivesse imersa em um meio puramente plástico, onde tal meio é deformado e assim este infere uma nova configuração à geometria imersa. Como já explanado, esta deformação da geometria vem dada pela modificação das características que definem a entidade. Na implementação do código computacional feita neste trabalho a deformação da geometria vem dada pela modificação das características que definem a entidade, entretanto esta deformação não decorre de uma deformação de um meio puramente plástico auxiliar mas sim de como um outro código computacional auxiliar (código de otimização) demanda a modificação da geometria, i.e., quem define como se deve dar a modificação da geometria é um outro código computacional auxiliar, ou um usuário que defina mudanças nas características que descrevem a entidade geométrica.

Formalmente, o método de deformação livre de forma é muito simples. Sederberg e Parry (1986) enunciam um corolário que é a base do método, tal diz que a geometria paramétrica deformada continua parametrizada após a deformação livre de forma. Este resultado é bastante óbvio, já que a mudança de forma vem dada pela mudança dos parâmetros que definem a geometria.

Para sólidos, a geometria é deformada pelo deslocamento dos pontos de controle do sólido Bézier. Para superfícies, existem dois procedimentos. O primeiro é baseado na utilização de um sólido auxiliar, onde os pontos de controle da superfície, imersos no sólido, são deslocados a partir da deformação do sólido auxiliar. O segundo procedimento é análogo ao procedimento descrito para sólidos.

A deformação de uma superfície, baseada na utilização de um sólido auxiliar, é bastante limitada quando se faz uso de parametrização NURBS, já que apenas podem ser deslocados os pontos de controle e seus respectivos pesos permaneceram inalterados. Entretanto, se a superfície for uma isosuperfície do sólido, a deformação que o sólido infere à superfície pode ser dada pelo deslocamento dos pontos de controle e pela mudança dos pesos.

O código desenvolvido baseado em NURBS permite a modificação dos pontos de controle em coordenadas homogêneas, assim são deslocados os pontos de controle e alterado os seus respectivos pesos.

Cabe salientar que o meio puramente plástico é necessário caso o problema de engenharia que se quer tratar envolva escoamento de fluidos, pois existe a necessidade de modificar a forma tanto do domínio do fluido como o domínio da estrutura (para interação fluido-estrutura) ou do obstáculo (para problemas sem interação).

A terceira exigência visa suprir a pouca capacidade de representação da parametrização Bézier e *B-spline*, e suprir a necessidade de um esquema de modificação local, que parametrizações Bézier não fornecem. Funções de base, como explanado, formam base de um espaço vetorial de maior dimensão, o qual é, obviamente, vantajoso quando se busca uma geometria ótima, sendo esta complexa ou não. Por esta razão foram implementados os algoritmos NURBS para computar curvas e superfícies e suas derivadas até uma ordem prescrita d .

2.11.1 Algoritmo de modificação de forma – deformação livre de forma –

A modificação de forma abordada neste trabalho, consiste simplesmente na modificação da malha de elementos finitos, i.e., como resultado final da modificação da geometria se quer a modificação da malha. Para tal feito, se faz uso da malha no espaço paramétrico. Conforme discutido, para modificar a forma, fixando os graus p e q , os vetores de nós \mathcal{U} e \mathcal{V} , ainda podem ser deslocados os pontos de controle $\mathbf{P}_{i,j}$ e os pesos $w_{i,j}$. Ainda, trabalhando em coordenadas homogêneas, simplesmente se modifica com $\mathbf{P}_{i,j}^w$.

Ao modificar os $\mathbf{P}_{i,j}^w$, a malha de elementos finitos no espaço paramétrico permanece inalterada. Logo, simplesmente se poderia mapear toda a malha do espaço paramétrico ao espaço Euclidiano para se obter a malha deformada, entretanto no que concerne à propriedade P.NURBS. 26 este mapeamento total é totalmente desnecessário. Segundo propriedade P.NURBS. 26 basta reposicionar na geometria deformada os nós da malha de elementos finitos que pertencem ao subdomínio modificado, i.e., modificando $\mathbf{P}_{i,j}^w$, os nós da malha pertencentes à $[u_i, u_{i+p+1}) \times [v_j, v_{j+q+1})$ são reposicionados. Para exemplificar esta propriedade, considerando um grau na direção u igual à três ($p = 3$), um grau na direção v igual à dois ($q = 2$) e os vetores de nós conforme representado na figura 2.34. Ainda, movendo os pontos de controle $\mathbf{P}_{2,3}^w$ e $\mathbf{P}_{5,4}^w$, os nós da malha de elementos finitos pertencentes aos subdomínios $[u_2, u_6) \times [v_3, v_6)$ e $[u_5, u_9) \times [v_4, v_7)$ são reposicionados na superfície modificada (estes subdomínios estão hachurados na figura 2.34). Estes subdomínios, claramente não possuem intersecção vazia.

O algoritmo de modificação de forma se desenvolve da seguinte maneira

Leitura de dados

- Leitura das entidades, tais como superfícies;
- Leitura da malha de elementos finitos no espaço paramétrico;
- Leitura da malha de elementos finitos no espaço Euclidiano;
- Índices dos pontos de controle (em coordenadas homogêneas) a deslocar;

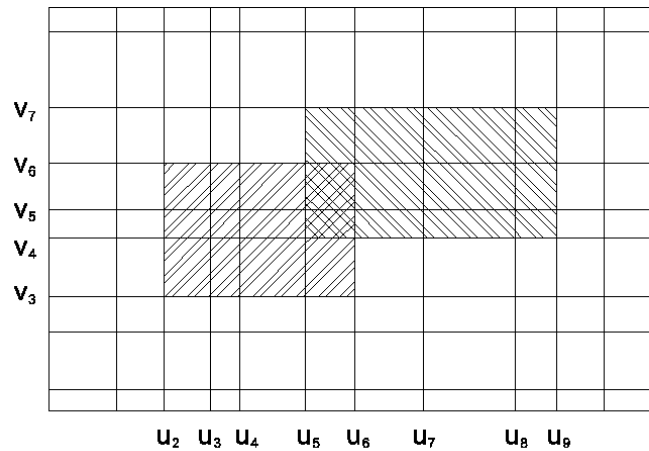


Figura 2.34: Subdomínios à serem reavaliados.

Deslocado os $\mathbf{P}_{i,j}^w$ (pelo usuário ou código de otimização)

Pré-processamento

- A partir dos índices dos pontos de controle a deslocar, são criadas variáveis vetores (dentro de uma variável tipo derivado), onde o número de variáveis criadas é igual ao número de subdomínios a serem modificados;
- É feita a contagem dos nós (da malha de elementos finitos) pertencentes a cada subdomínio a ser modificado, assim alocando dinamicamente cada variável vetor;
- Em cada variável vetor é armazenado o conjunto de nós (da malha de elementos finitos) pertencentes ao correspondente subdomínio;

Este pré-processamento descrito é realizado para cada superfície a ser modificada, como não se conhece o número de superfícies, assim como não se conhece o número de subdomínios a serem trabalhados, se fez o uso da variável tipo derivado. Onde esta contém todas as informações de todas as superfícies, e o pré-processamento é aplicado sobre esta variável, assim obtendo os conjuntos de nós (da malha de elementos finitos) a serem reposicionados de cada superfície. Desta maneira tem-se alocado na memória apenas os nós a serem reposicionados. Sendo assim, não se compromete a memória de armazenamento, e tem-se uma alta velocidade na modificação de forma, devido ao fato de que apenas são trabalhados os nós a serem reposicionados, e não toda a malha.

Processamento

- Avaliam-se todos os nós (da malha de elementos finitos) pertencentes aos vetores que representam os subdomínios a serem reavaliados, para cada superfície;
- Os nós da malha de elementos finitos são sobrescritos pelos nós reposicionados na nova geometria.

Observar que simplesmente são avaliados os nós reposicionados na nova geometria. Como o código computacional se ocupa de avaliar nós (da malha de elementos finitos) indexados aos subdomínios “marcados” a serem reavaliados, não existe o risco de reavaliar um mesmo subdomínio duas vezes (já que conforme a figura 2.34, pode existir a intersecção entre os subdomínios modificados por dois pontos de controle diferentes).

Em todos os exemplos executados, para exemplificar o código de deformação livre de forma, onde os pontos de controle foram deslocados aleatoriamente, o tempo mensurado pelo cronômetro do Fortran foi zero. Mesmos para os exemplos em que a malha de elementos finitos era extremamente densa. Com isto a quarta exigência, anteriormente descrita, foi satisfeita.

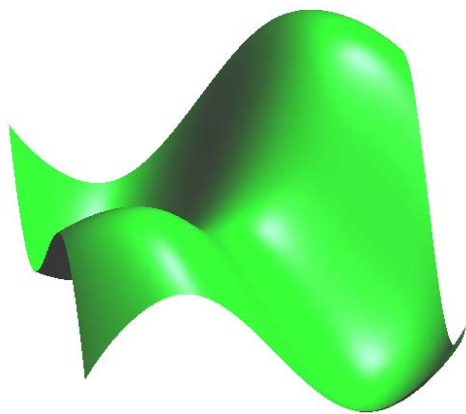
A quinta exigência, “código que de alguma maneira controle a deformação sobre os elementos”, é feita com a curvatura Gaussiana. A partir de alguns pontos selecionados, é calculada a curvatura Gaussiana antes da deformação e após, e assim o usuário estabelece uma porcentagem limite de deformação, i.e.,

$$\frac{\|k_{(u,v)}^{inicial} - k_{(u,v)}^{final}\|}{\|k_{(u,v)}^{inicial}\|} < tol \quad (2.73)$$

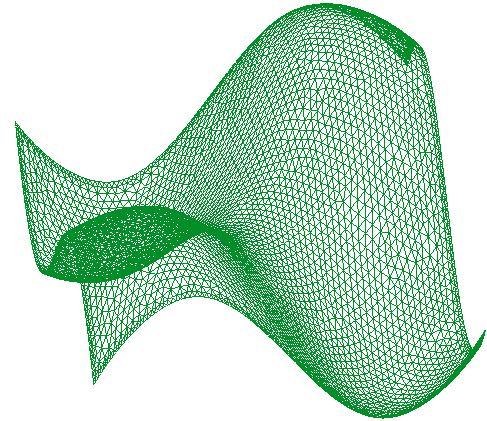
Este controle, não é muito sofisticado, entretanto na literatura pesquisada à cerca de otimização de forma (tanto de cascas como aerodinâmica) nada se encontrou sobre tal controle.

2.12 Exemplos de deformação livre de forma

Para ilustrar o código computacional desenvolvido em Fortran 2003, são apresentadas algumas geometrias às quais foram aplicadas deformações prescritas pelo usuário.

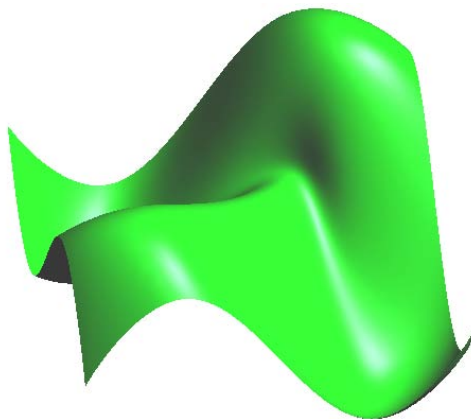


(a) Superfície NURBS (1)

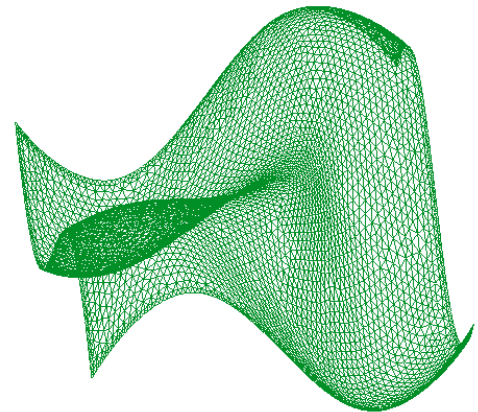


(b) Malha de elementos finitos (1)

Figura 2.35: Geometria original (1).



(a) Sup. NURBS deformada (1)

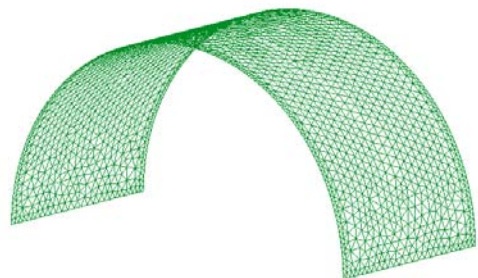


(b) Malha deformada (1)

Figura 2.36: Geometria deformada (1).

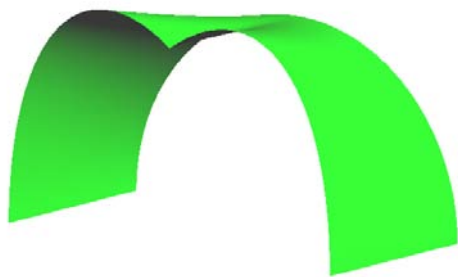


(a) Superfície NURBS (2)

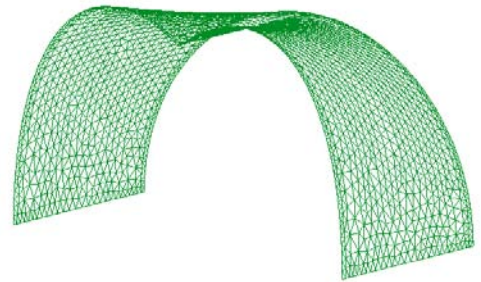


(b) Malha de elementos finitos (2)

Figura 2.37: Geometria original (2).

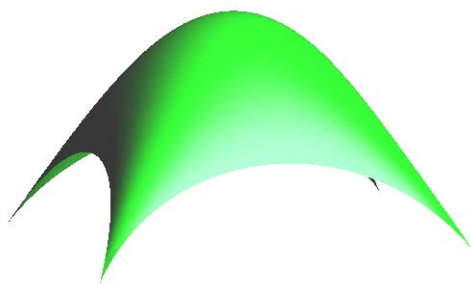


(a) Sup. NURBS deformada (2)

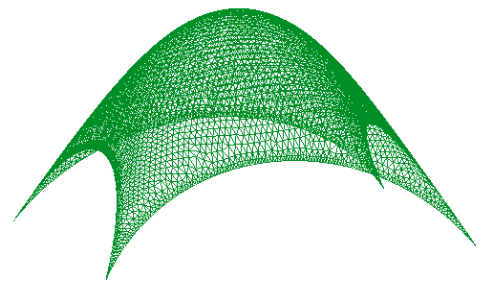


(b) Malha deformada (2)

Figura 2.38: Geometria deformada (2).

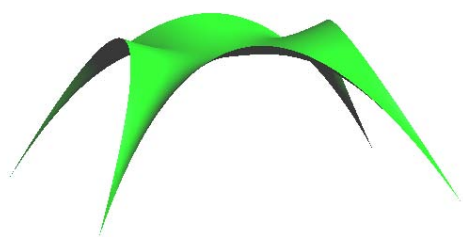


(a) Superfície NURBS (3)

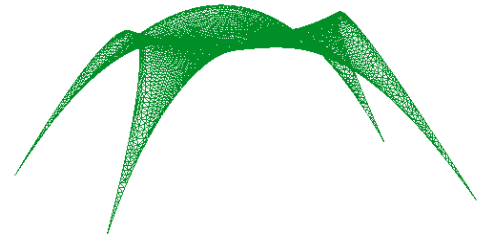


(b) Malha de elementos finitos (3)

Figura 2.39: Geometria original (3).

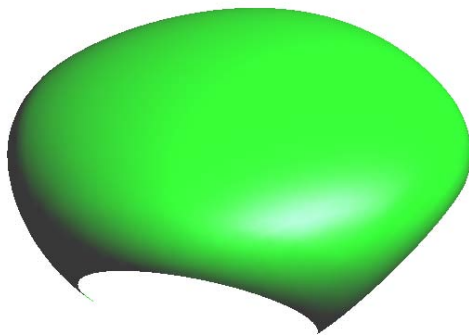


(a) Sup. NURBS deformada (3)

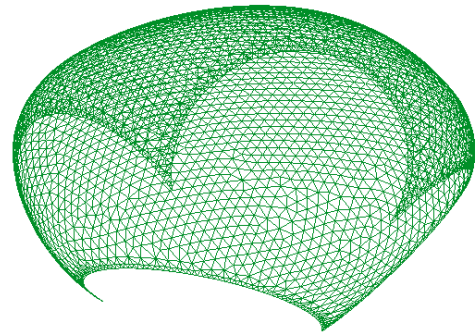


(b) Malha deformada (3)

Figura 2.40: Geometria deformada (3).

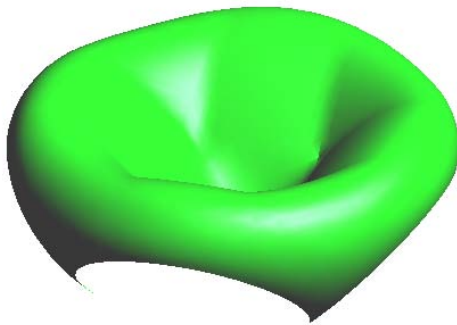


(a) Superfície NURBS (4)

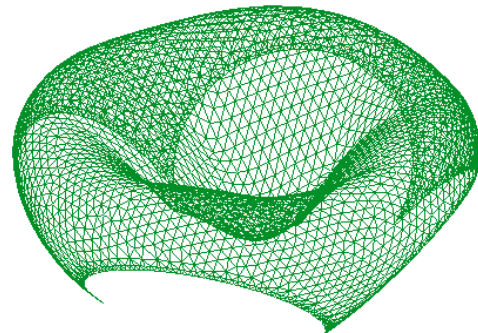


(b) Malha de elementos finitos (4)

Figura 2.41: Geometria original (4).

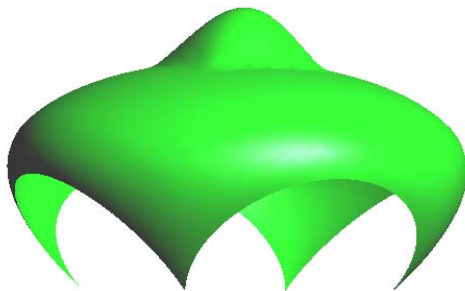


(a) Sup. NURBS deformada (4)

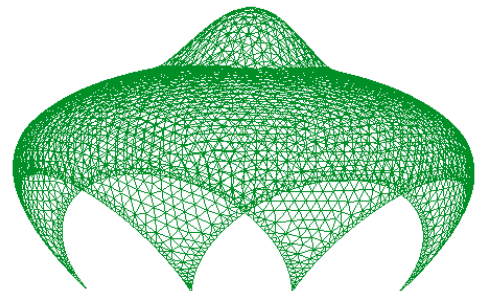


(b) Malha deformada (4)

Figura 2.42: Geometria deformada (4).

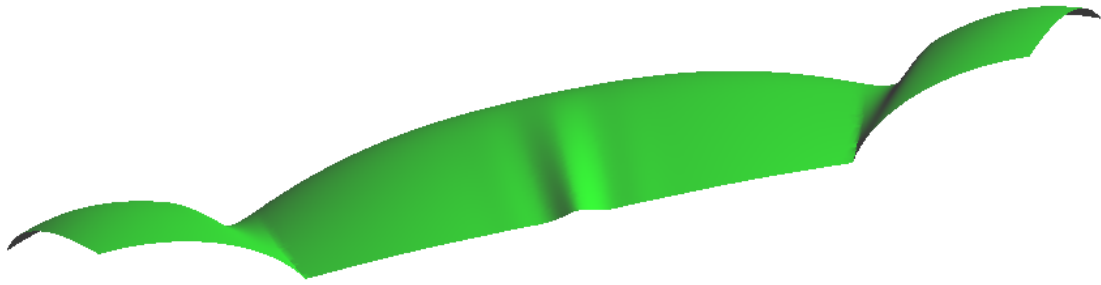


(a) Sup. NURBS deformada 2 (4)

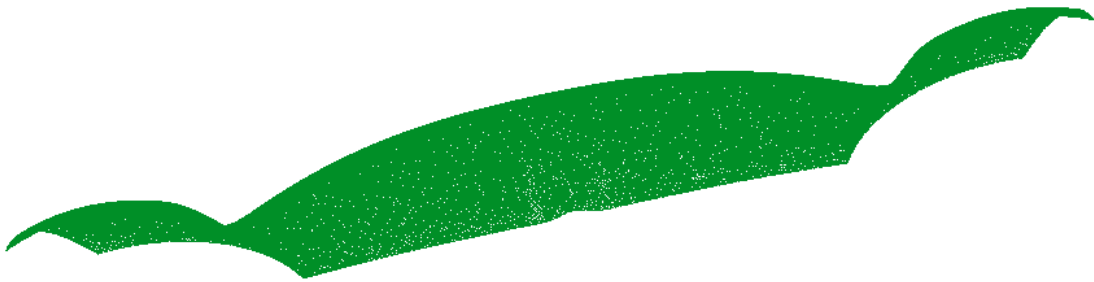


(b) Malha deformada 2 (4)

Figura 2.43: Geometria deformada 2 (4).

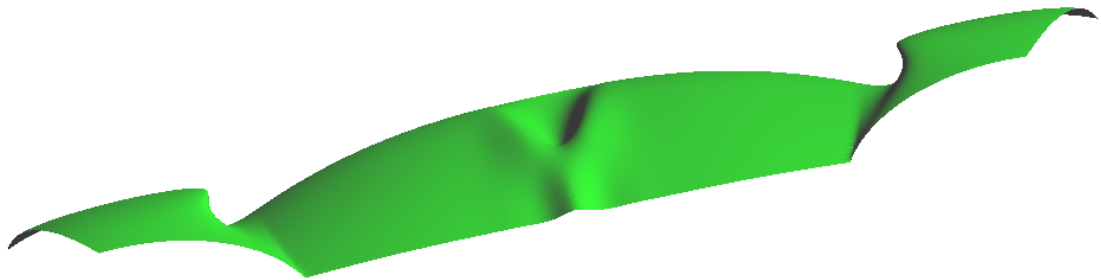


(a) Superfície NURBS (5)

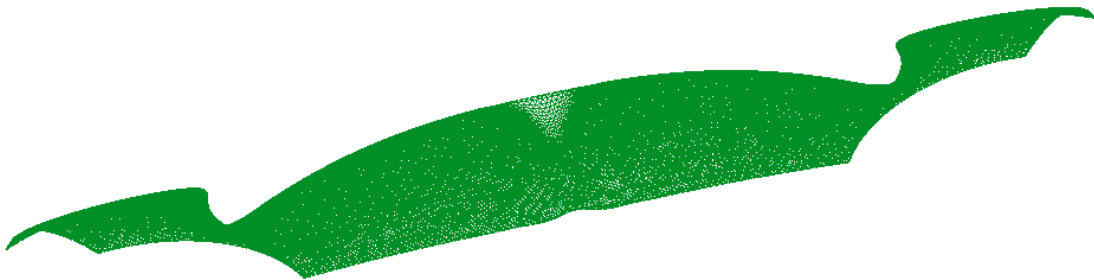


(b) Malha de elementos finitos (5)

Figura 2.44: Geometria original (5).

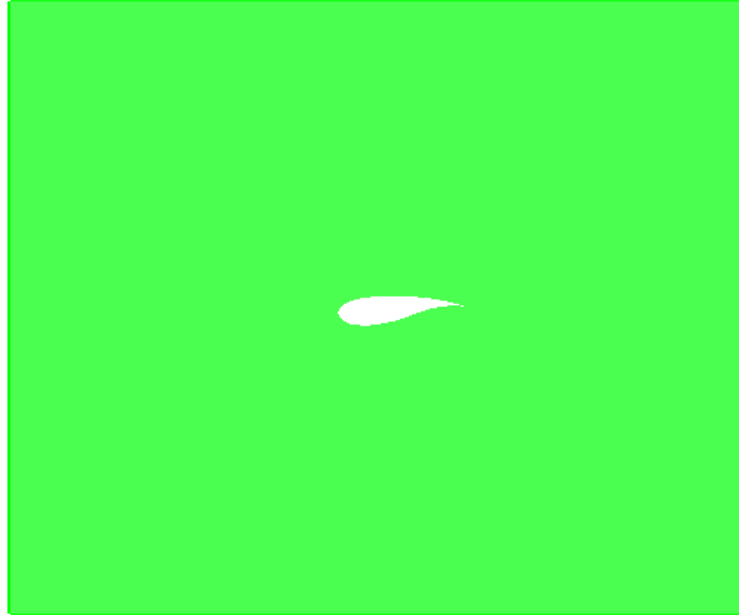


(a) Sup. NURBS deformada (5)

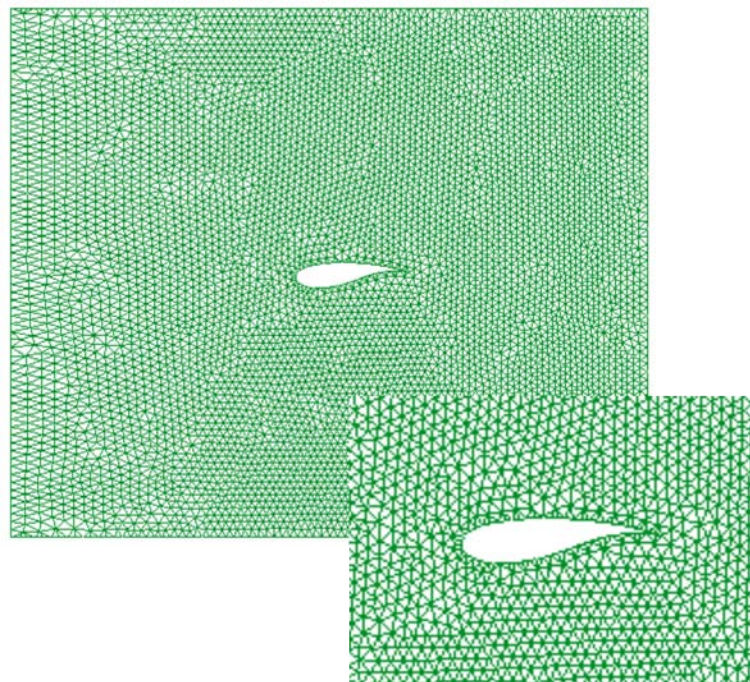


(b) Malha deformada (5)

Figura 2.45: Geometria deformada (5).

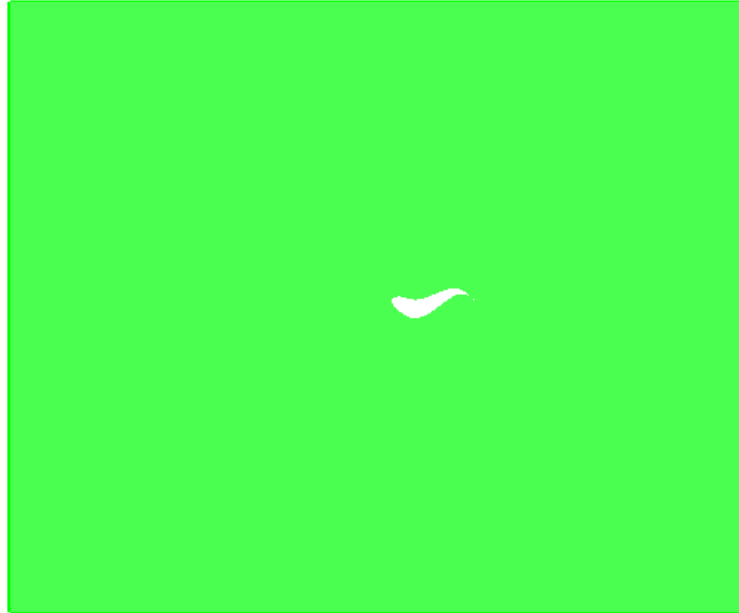


(a) Superfície NURBS (6)

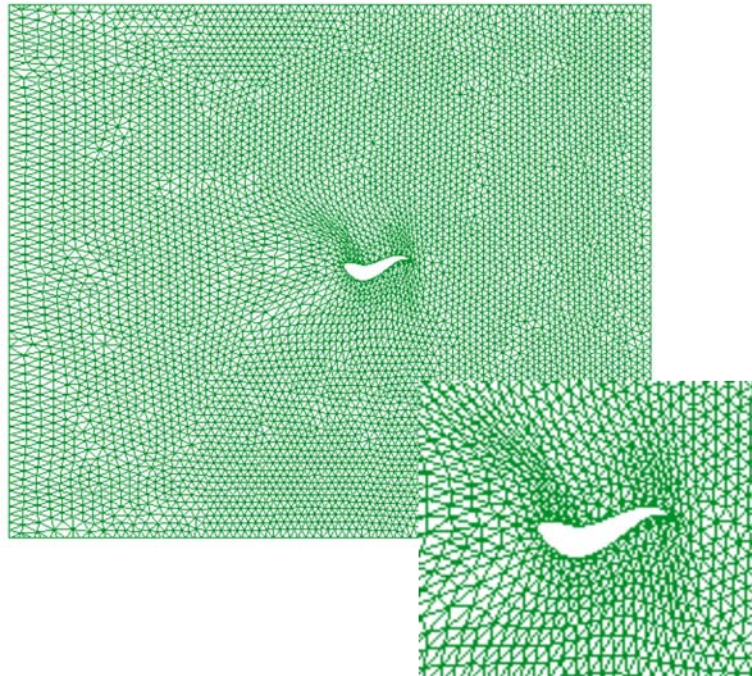


(b) Malha de elementos finitos (6)

Figura 2.46: Geometria original (6).



(a) Sup. NURBS deformada (6)



(b) Malha deformada (6)

Figura 2.47: Geometria deformada (6).

Capítulo 3

Otimização

3.1 Introdução

Os problemas que envolvem minimização ou maximização com ou sem restrições já são bem conhecidos na engenharia, estes talvez sejam os mais presentes devido ao fato de que sempre se busca uma solução ótima.

Em geral se busca minimizar uma função a várias variáveis na qual o seu domínio é restrito; entretanto, o problema não se restringe a funções, também o problema se apresenta para funcionais, no cálculo variacional.

Ao longo do capítulo 3, a notação será a seguinte: \mathbf{x}_0 para o ponto de partida, \mathbf{x}^* para o ponto de ótimo, e \mathbf{x}_k para o ponto da k -ésima iteração. Em geral todas as funções estão definidas no \mathbb{R}^n , são não-lineares e escritas na forma implícita.

3.2 Teoria de otimização sem restrições

Na otimização sem restrições, se minimiza uma *função objetivo* que depende de variáveis reais, sem restrições sobre qualquer variável. A formulação matemática é

$$\min_{\mathbf{x}} f(\mathbf{x}) \tag{3.1}$$

onde $\mathbf{x} \in \mathbb{R}^n$ com $n \geq 1$ e $f : \mathbb{R}^n \xrightarrow{f} \mathbb{R}$ é uma função suave.

3.2.1 Mínimo local

Definição 3.1 (Mínimo global) *No problema de minimização sem restrições tem-se que \mathbf{x}^* é ponto de mínimo global $\Leftrightarrow f(\mathbf{x}) \geq f(\mathbf{x}^*)$, $\forall \mathbf{x} \in \mathbb{R}^n$.*

Definição 3.2 (Mínimo local) *No problema de minimização sem restrições tem-se que \mathbf{x}^* é ponto de mínimo local $\Leftrightarrow f(\mathbf{x}) \geq f(\mathbf{x}^*)$, $\forall \mathbf{x} \in \mathcal{N}(\mathbf{x}^*, \delta)$, sendo δ um vetor real arbitrário; onde $\mathcal{N}(\mathbf{x}^*, \delta)$ é um entorno de \mathbf{x}^* de raio δ .*

3.2.2 Condições do ponto de ótimo sem restrições

Analisando a definição 3.2, parece que a única maneira de se encontrar o ponto de ótimo, \mathbf{x}^* , é examinando seu entorno. A seguir são apresentados os teoremas para realizar o reconhecimento de um mínimo.

Teorema 3.1 (Taylor) *Supondo que $f : \mathbb{R}^n \xrightarrow{f} \mathbb{R}$ é diferencialmente contínua e que $\delta \in \mathbb{R}^n$; sendo $\nabla(\cdot)$ o operador gradiente, tem-se que*

$$f(\mathbf{x} + \delta) = f(\mathbf{x}) + \nabla f(\mathbf{x} + t\delta)^T \delta \quad (3.2)$$

para um $t \in (0,1)$. Além disso, se f for duas vezes diferencialmente contínua, tem-se que

$$\nabla f(\mathbf{x} + \delta) = \nabla f(\mathbf{x}) + \int_0^1 \nabla^2 f(\mathbf{x} + t\delta) \delta dt \quad (3.3)$$

logo

$$f(\mathbf{x} + \delta) = f(\mathbf{x}) + \nabla f(\mathbf{x})^T \delta + \frac{1}{2} \delta^T \nabla^2 f(\mathbf{x} + t\delta) \delta \quad (3.4)$$

Teorema 3.2 (Condições necessárias de primeira ordem) *Se \mathbf{x}^* é um ponto de mínimo local de f e f é continuamente diferenciável em um entorno aberto de \mathbf{x}^* , logo $\nabla f(\mathbf{x}^*) = \mathbf{0}$.*

demonstração: Por redução ao absurdo, se assume $\nabla f(\mathbf{x}^*) \neq \mathbf{0}$. Definindo o vetor $\delta = -\nabla f(\mathbf{x}^*)$, note que $\delta^T \nabla f(\mathbf{x}^*) = -\|\nabla f(\mathbf{x}^*)\|^2 < 0$. Como ∇f é contínua no entorno de \mathbf{x}^* , há um escalar $T > 0$ tal que

$$\delta^T \nabla f(\mathbf{x}^* + t\delta) < 0, \quad \forall t \in [0, T]$$

para qualquer $\bar{t} \in (0, T]$, pelo teorema 3.1 tem-se que

$$f(\mathbf{x}^* + \bar{t}\delta) = f(\mathbf{x}^*) + \bar{t}\delta^T \nabla f(\mathbf{x}^* + t\delta), \quad \forall t \in (0, \bar{t})$$

logo, $f(\mathbf{x}^* + \bar{t}\delta) < f(\mathbf{x}^*)$ para qualquer $\bar{t} \in (0, T]$, assim se encontrou uma direção no ponto \mathbf{x}^* , na qual f decresce, logo \mathbf{x}^* não minimiza a função, e tem-se uma contradição.

Teorema 3.3 (Condições necessárias de segunda ordem) *Se \mathbf{x}^* é um ponto de mínimo local de $f(\mathbf{x})$ e $\nabla^2 f(\mathbf{x})$ (Hessiano) é contínua em um entorno aberto de \mathbf{x}^* , logo $\nabla f(\mathbf{x}^*) = \mathbf{0}$ e $\nabla^2 f(\mathbf{x}^*)$ positivo semi-definido.*

demonstração: Do teorema 3.2 tem-se que $\nabla f(\mathbf{x}^*) = \mathbf{0}$. Por redução ao absurdo, se assume $\nabla^2 f(\mathbf{x}^*)$ não é positivo semi-definido. Logo tomando δ tal que $\delta^T \nabla^2 f(\mathbf{x}^*) \delta < 0$. Como $\nabla^2 f$ é contínua no entorno de \mathbf{x}^* , há um escalar $T > 0$ tal que

$$\delta^T \nabla^2 f(\mathbf{x}^* + t\delta) \delta < 0, \quad \forall t \in [0, T]$$

Realizando a expansão de Taylor em torno de \mathbf{x}^* , tem-se para qualquer $\bar{t} \in (0, T]$ e $t \in (0, \bar{t})$ que

$$f(\mathbf{x}^* + \bar{t}\delta) = f(\mathbf{x}^*) + \bar{t}\delta^T \nabla f(\mathbf{x}^*) + \frac{1}{2}\bar{t}^2 \delta^T \nabla^2 f(\mathbf{x}^* + t\delta) \delta < f(\mathbf{x}^*)$$

Como no teorema 3.2, se encontrou uma direção no ponto \mathbf{x}^* , na qual f decresce, logo \mathbf{x}^* não minimiza a função, e tem-se uma contradição.

Teorema 3.4 (Condições suficientes de segunda ordem) *Supondo que $\nabla^2 f(\mathbf{x})$ é contínuo em um entorno aberto de \mathbf{x}^* e que $\nabla f(\mathbf{x}^*) = \mathbf{0}$ e $\nabla^2 f(\mathbf{x}^*)$ é positivo definido. Logo \mathbf{x}^* é um ponto de mínimo local de f .*

demonstração: Como o Hessiano é contínuo e positivo definido em \mathbf{x}^* , existe um raio $r > 0$ tal que $\nabla^2 f(\mathbf{x})$ é positivo definido para qualquer \mathbf{x} pertencente a uma esfera aberta $\mathcal{D} = \{\mathbf{z} \mid \|\mathbf{z} - \mathbf{x}^*\| < r\}$. Tomando um vetor δ diferente de zero, com $\|\delta\| < r$, tem-se $(\mathbf{x}^* + \delta) \in \mathcal{D}$ e que

$$\begin{aligned} f(\mathbf{x}^* + \delta) &= f(\mathbf{x}^*) + \delta^T \nabla f(\mathbf{x}^*) + \frac{1}{2}\delta^T \nabla^2 f(\mathbf{z}) \delta \\ &= f(\mathbf{x}^*) + \frac{1}{2}\delta^T \nabla^2 f(\mathbf{z}) \delta \end{aligned}$$

onde $\mathbf{z} = \mathbf{x}^* + t\delta$ para qualquer $t \in (0, 1)$. Desde de que $\mathbf{z} \in \mathcal{D}$, tem-se que $\delta^T \nabla^2 f(\mathbf{z}) \delta$ é positivo, logo $f(\mathbf{x}^* + \delta) > f(\mathbf{x}^*)$, assim resulta demonstrado.

Teorema 3.5 (Mínimo de função convexa) *Se f for convexa, qualquer mínimo local \mathbf{x}^* é um mínimo global de f . E ainda, se f for diferencialmente contínua, então qualquer ponto, \mathbf{x}^* , estacionário é um mínimo global de f .*

demonstração: Supondo que \mathbf{x}^* é um mínimo local mas não global. Então se pode encontrar um ponto $\mathbf{z} \in \mathbb{R}^n$ tal que $f(\mathbf{z}) < f(\mathbf{x}^*)$. Considerando o segmento de reta formado pelos pontos \mathbf{x}^* e \mathbf{z} , tem-se que

$$\mathbf{x} = \lambda \mathbf{z} + (1 - \lambda) \mathbf{x}^*, \quad \lambda \in (0, 1] \quad (3.5)$$

pelo fato de f ser convexa, tem-se que

$$f(\mathbf{x}) \leq \lambda f(\mathbf{z}) + (1 - \lambda) f(\mathbf{x}^*) < f(\mathbf{x}^*) \quad (3.6)$$

Qualquer entorno \mathcal{N} de \mathbf{x}^* contém um segmento de reta da expressão (3.5), logo o ponto $\mathbf{x} \in \mathcal{N}$ sempre satisfaz (3.6). Logo \mathbf{x}^* não é um mínimo local. Para a segunda parte do teorema, se supõe que \mathbf{x}^* não é um mínimo global e se escolhe um \mathbf{z} como acima foi proposto.

$$\begin{aligned} \nabla f(\mathbf{x}^*)^T (\mathbf{z} - \mathbf{x}^*) &= \frac{d}{d\lambda} f(\mathbf{x}^* + \lambda(\mathbf{z} - \mathbf{x}^*)) \Big|_{\lambda=0} \\ &= \lim_{\lambda \rightarrow 0^+} \frac{f(\mathbf{x}^* + \lambda(\mathbf{z} - \mathbf{x}^*)) - f(\mathbf{x}^*)}{\lambda} \\ &\leq \lim_{\lambda \rightarrow 0^+} \frac{\lambda f(\mathbf{z}) + (1 - \lambda) f(\mathbf{x}^*) - f(\mathbf{x}^*)}{\lambda} \\ &= f(\mathbf{z}) - f(\mathbf{x}^*) < 0 \end{aligned}$$

Logo, $\nabla f(\mathbf{x}^*) \neq 0$ e \mathbf{x}^* não é um ponto estacionário.

Estes resultados providenciam os fundamentos dos algoritmos para otimização sem restrições, algoritmos também necessários como parte dos algoritmos de otimização com restrições.

3.2.3 Problemas não suaves

O foco deste trabalho é realizado sobre funções suaves, onde as derivadas segundas existem e são contínuas. Cabe salientar que existem problemas interessantes nos quais as funções podem não ser suaves, como por exemplo, a função que relaciona a forma da estrutura com as frequências naturais.

A forma mais simples de contornar este problema é criar subdomínios (funções por partes), nas quais as fronteiras são as descontinuidades da função original. Ainda existem alternativas muito mais elegantes e eficientes discutidas por Hiriart-Urruty e Lemaréchal (1993) que não serão tratadas.

3.2.4 Estratégias: busca linear e região confiável

A partir de um ponto \mathbf{x}_0 , os algoritmos de otimização geram uma sequência de iterações $\{\mathbf{x}_k\}_{k=0}^{\infty}$ que termina quando nenhum progresso a mais pode ser realizado ou quando este ponto parece ser a solução aproximada com suficiente acurácia. Na decisão de como se mover de uma iteração \mathbf{x}_k para a próxima iteração, os algoritmos usam informações à cerca de f sobre \mathbf{x}_k , e possivelmente também informações sobre iterações recente em $\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{k-1}$. Estes usam estas informações para encontrar uma nova iteração \mathbf{x}_{k+1} com um valor de f menor do que na iteração anterior. Existem algoritmos *não monótonos*, que não insistem no decrescimento de f em todos os passos, mas estes requerem que f decresça após um número m prescrito de iterações. Isto leva à $f(\mathbf{x}_k) < f(\mathbf{x}_{k-m})$.

Existem duas estratégias fundamentais para se mover da iteração atual \mathbf{x}_k para a nova iteração \mathbf{x}_{k+1} .

Na estratégia da *busca linear*, o algoritmo escolhe uma direção δ_k e busca ao longo desta, a partir da iteração atual \mathbf{x}_k , uma nova posição com um menor valor funcional. A distância percorrida sobre a direção δ_k pode ser aproximada pela solução de uma minimização unidimensional; sendo α_k o comprimento deste passo tem-se

$$\min_{\alpha_k > 0} f(\mathbf{x}_k + \alpha_k \delta_k) \quad (3.7)$$

A solução de (3.7) resulta no máximo benefício na direção δ_k , mas uma minimização exata é cara e desnecessária. Em vez disto, o algoritmo de busca linear gera um número limitado de passos, até encontrar um ponto que aproxime o mínimo de (3.7). Sobre o novo ponto, uma nova direção e um novo comprimento do passo são computados, e o processo é repetido.

Na estratégia da *região confiável*, a informação coletada à cerca de f é usada para construir uma *função modelo* m_k tal que seu comportamento em um entorno do ponto atual \mathbf{x}_k seja

similar ao da função objetivo f . Como o modelo m_k é realizado em um entorno do ponto atual \mathbf{x}_k , se restringe a busca do mínimo neste entorno de \mathbf{x}_k , resolvendo o seguinte subproblema

$$\min_{\delta} m_k(\mathbf{x}_k + \delta), \quad (\mathbf{x}_k + \delta) \in \text{região confiável} \quad (3.8)$$

Se o candidato à solução não produz um decréscimo suficiente de f , se conclui que a região é muito ampla. Usualmente, a região confiável é uma esfera definida por $\|\delta\| \leq \Delta_k$, onde o escalar $\Delta_k > 0$ é chamado de raio da região confiável.

O modelo m_k em (3.8), é usualmente definido por uma função quadrática (expansão de Taylor) da forma

$$m_k(\mathbf{x}_k + \delta) = f_k + \delta^T \nabla f_k + \frac{1}{2} \delta^T \mathbf{B}_k \delta \quad (3.9)$$

onde f_k , ∇f_k e \mathbf{B}_k são um escalar, um vetor e uma matriz, respectivamente. ∇f_k é o gradiente enquanto a matriz \mathbf{B}_k é o Hessiano $\nabla^2 f_k$ ou alguma aproximação do mesmo.

A principal diferença entre as duas estratégias aqui apresentadas é que o tamanho da região confiável afeta a direção da próxima iteração, enquanto que na busca linear somente existe uma direção de busca. Outro aspecto que diferencia estas duas abordagens é a ordem de escolha entre a *direção* e a *distância* para se mover na próxima iteração. Na busca linear se parte de uma direção δ_k fixa e se identifica uma distância apropriada, chamada de α_k . Na região confiável, se escolhe uma distância máxima – raio da região confiável Δ_k – e logo se procura uma direção e um passo que atendam o melhor desempenho possível sujeitos à restrição imposta pelo raio. Se este passo resulta ser insatisfatório, se reduz o raio Δ_k e se recomputa o processo.

Direção na busca linear

A direção de decréscimo $-\nabla f_k$ é a escolha mais óbvia para a direção de busca no método de busca linear. Esta direção é a de maior decréscimo de f neste ponto; para justificar esta escolha se aplica o teorema de Taylor (teorema 3.1), o qual revela que para uma direção de busca δ e uma distância α tem-se

$$f(\mathbf{x}_k + \alpha\delta) = f(\mathbf{x}_k) + \alpha\delta^T \nabla f(\mathbf{x}_k) + \frac{1}{2} \alpha^2 \delta^T \nabla^2 f(\mathbf{x}_k + t\delta) \delta \quad t \in (0, \alpha)$$

desta maneira, para descobrir a direção de maior decréscimo se busca obter o menor valor possível do produto escalar $\delta^T \nabla f(\mathbf{x}_k)$. Como $\delta^T \nabla f_k = \|\delta\| \cdot \|\nabla f_k\| \cdot \cos \theta$ e com

$\|\delta\| = 1$ o $\cos \theta$ deve resultar no menor número possível, ou seja, -1 . E isto ocorre para o valor de $\theta = \pi \text{ rad}$, assim revelando que a direção de δ deve ser $-\nabla f_k$, logo

$$\delta = -\frac{\nabla f_k}{\|\nabla f_k\|}$$

O método de maior declividade é uma busca linear que se move ao longo de $\delta_k = -\nabla f_k$ a cada passo. Uma das vantagens deste método é que este requer o computo do gradiente mas não o das derivadas segunda.

Os métodos de busca linear podem usar direções de busca diferentes à direção de maior decréscimo de f . Em geral, qualquer direção de decréscimo pode ser usada – qualquer direção que faça um ângulo estritamente menor que $\frac{\pi}{2}$ radianos com $-\nabla f_k$ – se garante um decréscimo de f , com um passo suficientemente pequeno. Isto pode ser verificado a partir do teorema de Taylor (teorema 3.1); a partir de (3.4) tem-se

$$f(\mathbf{x}_k + \epsilon \delta_k) = f(\mathbf{x}_k) + \epsilon \delta_k^T \nabla f_k + O(\epsilon^2)$$

Aqui se distingue entre $O(\cdot)$ e $o(\cdot)$ ¹. Enquanto δ_k é uma direção de decréscimo de f , o ângulo θ_k entre δ_k e ∇f_k tem $\cos \theta_k < 0$, então

$$\delta_k^T \nabla f_k = \|\delta_k\| \|\nabla f_k\| \cos \theta_k$$

Deste segue que $f(\mathbf{x}_k + \epsilon \delta_k) < f(\mathbf{x}_k)$ para todo valor de ϵ suficientemente pequeno.

Outra importante direção de busca – talvez a mais importante delas – é a direção de *Newton*. Esta deriva da série de Taylor de segunda ordem aproximando $f(\mathbf{x}_k + \delta)$, a qual resulta em

$$f(\mathbf{x}_k + \delta) \approx f_k + \delta^T \nabla f_k + \frac{1}{2} \delta^T \nabla^2 f_k \delta \stackrel{\text{def}}{=} m_k(\delta) \quad (3.10)$$

Assumindo que $\nabla^2 f_k$ é positivo definida, se obtém a direção de Newton achando o vetor δ que minimiza $m_k(\delta)$. Por simplicidade, se assume a derivada de $m_k(\delta)$ ser zero, então se obtém de forma explícita

$$\delta_k^N = -\nabla^2 f_k^{-1} \nabla f_k \quad (3.11)$$

A direção de Newton é confiável quando a diferença entre a função modelo e a função original não for grande. Comparando (3.10) com (3.4) tem-se que o terceiro termo da

¹ $o(s)$ tem o significado usual da análise matemática. Enquanto $o(s)$ representa termos cuja ordem de grandeza é pequena, comparada com s , $O(s)$ representa termos da ordem de grandeza de s , isto é, $\lim_{s \rightarrow 0} \frac{o(s)}{s} = 0$ e $\lim_{s \rightarrow 0} \frac{O(s)}{s}$ é finito

expansão de Taylor $\nabla^2 f(\mathbf{x}_k + t\delta)$ foi substituído por $\nabla^2 f_k = \nabla^2 f(\mathbf{x}_k)$. se $\nabla^2 f(\cdot)$ for suficientemente suave, esta diferença introduz uma perturbação de somente $O(\|\delta\|^3)$, então quando $\|\delta\|$ for pequeno, a aproximação $f(\mathbf{x}_k + t\delta) \approx m_k(\delta)$ é precisa, (Nocedal e Wright, 1999).

A direção de Newton pode ser usada na busca linear quando $\nabla^2 f_k$ for positivo definido. A maior parte das implementações de busca linear com a direção de Newton usa um passo de comprimento unitário $\alpha = 1$, o qual é possível ajustar caso a redução do valor de f não seja relevante. Quando o Hessiano não for positivo definido, a direção de Newton não estará definida, já que $\nabla^2 f_k^{-1}$ não está definido.

Métodos que usam a direção de Newton atendem uma rápida razão de convergência, tipicamente Q-quadrática. Quando se chega a um entorno da solução, a convergência com alta acurácia ocorre com apenas algumas iterações. O maior problema deste métodos é o computo do Hessiano, muitas vezes impreciso e caro.

A direção de busca *Quase-Newton* provê uma alternativa atrativa pelo fato de não precisar do computo do Hessiano e ainda atende uma razão de convergência Q-superlinear. Em vez de usar o “verdadeiro” Hessiano $\nabla^2 f_k$, este usa uma aproximação \mathbf{B}_k , a qual é atualizada a cada passo, levando em conta o conhecimento adicional obtido durante a ultima iteração. A atualização é realizada pelo fato de que mudanças no gradiente ao longo da direção de busca provêm informações à cerca das derivadas segundas de f ao longo da direção de busca. Usando (3.3) adicionando e subtraindo o termo $\nabla^2 f(\mathbf{x})\delta$ tem-se

$$\nabla f(\mathbf{x} + \delta) = \nabla f(\mathbf{x}) + \nabla^2 f(\mathbf{x})\delta + \int_0^1 [\nabla^2 f(\mathbf{x} + t\delta) - \nabla^2 f(\mathbf{x})] \delta dt$$

Como $\nabla f(\cdot)$ é contínua, o tamanho do último termo da integral é $o(\|\delta\|)$. Deixando $\mathbf{x} = \mathbf{x}_k$ e $\delta = \mathbf{x}_{k+1} - \mathbf{x}_k$, tem-se que

$$\nabla f_{k+1} = \nabla f_k + \nabla^2 f_k(\mathbf{x}_{k+1} - \mathbf{x}_k) + o(\|\mathbf{x}_{k+1} - \mathbf{x}_k\|)$$

Quando \mathbf{x}_k e \mathbf{x}_{k+1} estão contidos em um entorno da solução \mathbf{x}^* , com $\nabla^2 f$ positivo definido, o termo final desta expansão é dominado por $\nabla^2 f_{k+1}(\mathbf{x}_{k+1} - \mathbf{x}_k)$; então pode-se escrever que

$$\nabla^2 f_{k+1}(\mathbf{x}_{k+1} - \mathbf{x}_k) \approx \nabla f_{k+1} - \nabla f_k \quad (3.12)$$

Para a nova aproximação do Hessiano se escolhe um \mathbf{B}_{k+1} tal que satisfaça (3.12) do

verdadeiro Hessiano. Esta condição é conhecida como *equação secante*:

$$\mathbf{B}_{k+1}\mathbf{s}_k = \mathbf{y}_k \quad (3.13)$$

onde

$$\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k, \quad \mathbf{y}_k = \nabla f_{k+1} - \nabla f_k$$

Normalmente são impostas outras condições sobre \mathbf{B}_{k+1} , tais como simetria (motivado pelo fato do verdadeiro Hessiano ser simétrico) e que a diferença entre sucessivas aproximações tenha baixo posto, para atender questões de convergência. A aproximação inicial \mathbf{B}_0 deve ser pré-definida, no ponto de partida, (Nocedal e Wright, 1999).

As duas formas mais populares de atualizar a aproximação \mathbf{B}_0 do Hessiano são: a fórmula *simétrica de posto um*, ou SR1 do acrônimo em inglês *symmetric-rank-one*, definida por

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{(\mathbf{y}_k - \mathbf{B}_k\mathbf{s}_k)(\mathbf{y}_k - \mathbf{B}_k\mathbf{s}_k)^T}{(\mathbf{y}_k - \mathbf{B}_k\mathbf{s}_k)^T \mathbf{s}_k} \quad (3.14)$$

e a fórmula *BFGS*, nomeada após seus inventores, Broyden, Fletcher, Goldfarb e Shanno, a qual é denominada como fórmula *simétrica de posto dois* e definida por

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{\mathbf{B}_k\mathbf{s}_k\mathbf{s}_k^T\mathbf{B}_k}{\mathbf{s}_k^T\mathbf{B}_k\mathbf{s}_k} - \frac{\mathbf{y}_k\mathbf{y}_k^T}{\mathbf{y}_k^T\mathbf{s}_k} \quad (3.15)$$

Ambas satisfazem a equação da secante e ambas mantêm a simetria. A fórmula BFGS (3.15) gera uma aproximação positivo definida sempre que a aproximação inicial \mathbf{B}_0 for positivo definida e $\mathbf{s}_k^T\mathbf{y}_k > 0$.

A direção de quase-Newton vem dada pelo uso de \mathbf{B}_k no lugar do Hessiano exato na equação

$$\delta_k = -\mathbf{B}_k^{-1}\nabla f_k \quad (3.16)$$

Na prática, na implementação do método de quase-Newton não é preciso inverter \mathbf{B}_k , de fato é aplicado uma aproximação da inversa $\mathbf{H}_k \stackrel{\text{def}}{=} \mathbf{B}_k^{-1}$

$$\mathbf{H}_{k+1} = (\mathbf{I} - \mathbf{c}_k\mathbf{s}_k\mathbf{y}_k^T)\mathbf{H}_k(\mathbf{I} - \mathbf{c}_k\mathbf{s}_k^T\mathbf{y}_k) + \mathbf{c}_k\mathbf{s}_k\mathbf{s}_k^T, \quad \mathbf{c}_k = \frac{1}{\mathbf{y}_k^T\mathbf{s}_k} \quad (3.17)$$

A ultima classe de direção de busca é gerada pelo *método não linear de gradientes conjugados*. Esta toma a seguinte forma

$$\delta_k = -\nabla f(\mathbf{x}_k) + \beta_k\delta_{k-1}$$

onde β_k é um escalar que assegura que δ_k e δ_{k-1} sejam conjugados em relação ao Hessiano. O método de gradientes conjugados teve origem na solução de sistemas de equações lineares do tipo $\mathbf{Ax} = \mathbf{b}$, onde a matriz dos coeficientes \mathbf{A} é simétrica e positivo definida. O problema de resolver este sistema linear é equivalente ao problema de minimizar a função convexa quadrática definida por

$$\phi(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{Ax} + \mathbf{b}^T \mathbf{x}$$

Em geral, direções geradas por gradientes conjugados não lineares são muito mais eficientes do que as direções de maior decrescimento e são mais simples de computar. Este método não atende a uma rápida razão de convergência comparado com o método de Newton e quase-Newton, mas a vantagem é que este não precisa armazenar nenhuma matriz, (Nocedal e Wright, 1999).

Todas estas direções de busca podem ser usadas diretamente nos algoritmos de busca linear. Exceto os gradientes conjugados estes têm aplicação análoga na estrutura dos algoritmos de região confiável.

Modelos para região confiável

Considerando $\mathbf{B}_k = \mathbf{0}$ em (3.9) e definindo a região confiável com o uso da norma Euclideana, o subproblema da região confiável (3.8) se torna

$$\min_{\delta} f_k + \delta^T \nabla f_k \quad \text{sujeito a } \|\delta\| \leq \Delta_k$$

Pode-se escrever a solução deste problema de forma fechada

$$\delta_k = \frac{\Delta_k \nabla f_k}{\|\nabla f_k\|}$$

Este caso simplesmente é o método de maior declividade com um comprimento de passo determinado pelo raio da região confiável; a abordagem por região confiável e busca linear são essencialmente as mesmas para este caso.

Algoritmos mais interessantes são obtido escolhendo \mathbf{B}_k sendo o Hessiano exato no modelo quadrático (3.9), pois não é preciso fazer nada quando $\nabla^2 f_k$ não for positivo definido, já que o subproblema (3.8) garante uma solução δ_k . O método de região confiável com a direção de Newton, na prática, é altamente efetivo.

Se o Hessiano \mathbf{B}_k no modelo quadrático (3.9) é definido pela aproximação quase-Newton, tem-se o método de região confiável quase-Newton.

3.2.5 Fator de escala

A performance de um algoritmo depende crucialmente de como o problema foi formulado. Uma importante questão na formulação do problema é o *fator de escala*. Na otimização sem restrições se diz que um problema é *pobremamente escalado* se uma perturbação da variável \mathbf{x} , numa certa direção, produz uma variação muito maior do valor de f , do que uma perturbação da variável \mathbf{x} em outra direção (Nocedal e Wright, 1999).

O fator de escala é alterado (muitas vezes intencionalmente) quando as unidades para representar as variáveis são trocadas. Também pode existir a necessidade de mudar o fator de escala durante o processo de otimização, o qual pode ser diferente para todas as variáveis. Alguns algoritmos são muito sensíveis ao fator de escala; estes algoritmos

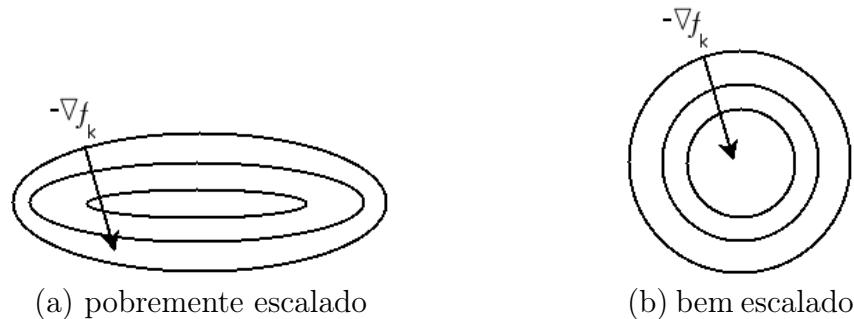


Figura 3.1: Fator de escala.

são os que seguem o sentido oposto do gradiente de f . Na figura 3.1 se percebe como o caso da esquerda (figura 3.1a problema pobremamente escalado) não produzirá uma grande redução de f na seguinte iteração, enquanto o caso da direita (figura 3.1b problema bem escalado) produzirá uma grande redução de f na seguinte iteração.

3.2.6 Razão de convergência

Um modo de medir a performance de um algoritmo é pela sua razão de convergência. Neste são apresentados alguns tipos de avaliação da convergência.

Sendo $\{\mathbf{x}_k\}$ uma sequência no \mathbb{R}^n que converge para \mathbf{x}^* . Se diz que a convergência é *Q-linear* se existe uma constante $r \in (0,1)$ tal que

$$\frac{\|\mathbf{x}_{k+1} - \mathbf{x}^*\|}{\|\mathbf{x}_k - \mathbf{x}^*\|} \leq r, \quad \forall k \text{ suficientemente grande} \quad (3.18)$$

O prefixo “Q” faz referência à “quociente”, pois este tipo de convergência é definido em termos do quociente de sucessivos erros.

A convergência é dita *Q-superlinear* se

$$\lim_{k \rightarrow \infty} \frac{\|\mathbf{x}_{k+1} - \mathbf{x}^*\|}{\|\mathbf{x}_k - \mathbf{x}^*\|} = 0 \quad (3.19)$$

A convergência *Q-quadrática* é uma razão mais rápida que as anteriores, e é obtida se

$$\frac{\|\mathbf{x}_{k+1} - \mathbf{x}^*\|}{\|\mathbf{x}_k - \mathbf{x}^*\|^2} \leq M, \quad \forall k \text{ suficientemente grande} \quad (3.20)$$

A velocidade de convergência depende de r e (mais fracamente) de M , e estes valores não dependem apenas do algoritmo, mas também das propriedades particulares de cada problema.

Obviamente, qualquer sequência que converge *Q-quadraticamente* também converge *Q-superlinearmente*, e qualquer sequência que converge *Q-superlinearmente* também converge *Q-linearmente*. Em geral, é dito que a convergência *Q-ordem* é p (com $p > 1$) se existe uma constante positiva M tal que

$$\frac{\|\mathbf{x}_{k+1} - \mathbf{x}^*\|}{\|\mathbf{x}_k - \mathbf{x}^*\|^p} \leq M, \quad \forall k \text{ suficientemente grande} \quad (3.21)$$

3.2.7 Razão de convergência R

Uma forma fraca de convergência, caracterizada pelo prefixo “R” (de raiz), é a que trata da total razão de decrescimento do erro, em vez do decrescimento de um simples passo do algoritmo. A convergência *R-linear* ocorre se existe uma sequência de escalares $\{\nu_k\}$ não negativos tais que

$$\|\mathbf{x}_k - \mathbf{x}^*\| \leq \nu_k, \quad \forall k \wedge \{\nu_k\} \text{ converge Q-linearmente para zero} \quad (3.22)$$

A sequência $\{\|\mathbf{x}_k - \mathbf{x}^*\|\}$ é dita *dominada* por $\{\nu_k\}$. Sequências $\{\|\mathbf{x}_k - \mathbf{x}^*\|\}$ dominadas por uma sequência Q-superlinear, Q-quadrática são convergências do tipo R-superlinear e R-quadrática, respectivamente.

A maior parte das análises de convergência, de algoritmos de otimização, são tratados com Q-convergência.

3.3 Teoria de otimização com restrições

Na otimização com restrições, se minimiza uma função objetivo que depende de variáveis reais, com restrições sobre qualquer variável. A formulação matemática é

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) \quad \text{sujeito a} \quad \begin{cases} c_i(\mathbf{x}) = 0, & i \in \mathcal{E} \\ c_i(\mathbf{x}) \geq 0, & i \in \mathcal{I} \end{cases} \quad (3.23)$$

onde f , c_i estão definidas no \mathbb{R}^n ; e \mathcal{E} e \mathcal{I} são dois conjuntos finitos de índices. O vetor \mathbf{x} contém as chamadas *variáveis de projeto*, f é a *função objetivo*, $c_i, i \in \mathcal{E}$ são as *restrições de igualdade* e $c_i, i \in \mathcal{I}$ são as *restrições de desigualdade*.

Um ponto \mathbf{x} que satisfaz todas as restrições de desigualdades e igualdades é chamado de *viável* ou *possível* e a região definida pela interseção de todas as funções é chamada de *região viável*. Uma restrição de desigualdade define dois subdomínios, pela sua fronteira, um subdomínio viável e outro inviável. Um ponto sobre esta fronteira define dita restrição como sendo *ativa*; um ponto contido no subdomínio viável define dita restrição como sendo *inativa*; finalmente, um ponto contido no subdomínio inviável define dita restrição como sendo *violada*. A restrição de igualdade satisfeita, define tal restrição como ativa.

Definição 3.3 (Região viável) *A região viável é definida como*

$$\Omega = \{\mathbf{x} \mid c_i(\mathbf{x}) = 0, \quad i \in \mathcal{E}; \quad c_i(\mathbf{x}) \geq 0, \quad i \in \mathcal{I}\} \quad (3.24)$$

De forma mais compacta se pode escrever (3.23) da seguinte forma

$$\min_{\mathbf{x} \in \Omega} f(\mathbf{x}) \quad (3.25)$$

Na seção 3.2, foi caracterizado o ponto de mínimo de \mathbf{x}^* para o problema de otimização sem restrições. Nesta seção 3.3, se introduz uma outra definição à cerca do ponto de mínimo.

Definição 3.4 (Mínimo local estrito) *No problema de minimização tem-se que \mathbf{x}^* é ponto de mínimo local estrito $\Leftrightarrow f(\mathbf{x}) > f(\mathbf{x}^*), \quad \forall \mathbf{x} \in \mathcal{N}(\mathbf{x}^*, \delta)$, sendo δ um vetor real arbitrário.*

3.3.1 Condições do ponto de ótimo de primeira ordem

Proposições das condições necessárias de primeira ordem

Em geral, o *Lagrangiano* ou *Função Lagrangiana* para o problema de otimização com restrições, é definido como

$$\mathcal{L}(\mathbf{x}, \lambda) = f(\mathbf{x}) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i c_i(\mathbf{x}) \quad (3.26)$$

onde λ_i são os multiplicadores de Lagrange.

Definição 3.5 (Conjunto ativo) O conjunto ativo $\mathcal{A}(\mathbf{x})$ sobre todo \mathbf{x} viável é definido como a união do conjunto de índices \mathcal{E} com os índices das restrições de desigualdade ativas, isto produz

$$\mathcal{A}(\mathbf{x}) = \mathcal{E} \cup \{i \in \mathcal{I} \mid c_i(\mathbf{x}) = 0\} \quad (3.27)$$

O vetor $\nabla c_i(\mathbf{x})$ é usualmente chamado de *normal* à restrição c_i em \mathbf{x} , pois este é perpendicular às curvas de contorno de c_i em \mathbf{x} , e no caso da restrição de desigualdade, este vetor “aponta” na direção da região viável desta restrição. É possível, entretanto, que $\nabla c_i(\mathbf{x})$ desapareça devido à representação de $c_i(\mathbf{x})$; logo o termo $\lambda_i \nabla c_i(\mathbf{x})$ será o vetor nulo para todos os valores de λ_i e portanto perderá o rol no gradiente do Lagrangiano $\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \lambda)$.

Usualmente se faz uma consideração chamada de *qualificação das restrições* para se assegurar de que não ocorra um comportamento degenerado no valor de \mathbf{x} em questão.

Definição 3.6 (LICQ) Dado o ponto $\mathbf{x}^* \in \Omega$ e o conjunto ativo $\mathcal{A}(\mathbf{x}^*)$, se diz que a qualificação das restrições linearmente independente (LICQ do acrônimo em inglês: *linear independence constraint qualification*) é assegurada se os gradientes das restrições do conjunto ativo $\{\nabla c_i(\mathbf{x}^*), i \in \mathcal{A}(\mathbf{x}^*)\}$ forem linearmente independente.

Observar que esta condição assegura que todos os gradientes das restrições ativas sejam diferentes do vetor nulo.

Teorema 3.6 (Condições necessárias de primeira ordem) *Supondo que $\mathbf{x}^* \in \Omega$ é um mínimo local de (3.23) e que é assegurado a LICQ em \mathbf{x}^* . Então existe um vetor multiplicador de Lagrange λ^* , com componentes λ_i^* , $i \in \mathcal{E} \cup \mathcal{I}$, tal que as seguintes condições são satisfeitas em $(\mathbf{x}^*, \lambda^*)$*

$$\nabla_x \mathcal{L}(\mathbf{x}^*, \lambda^*) = \mathbf{0}, \quad (3.28a)$$

$$c_i(\mathbf{x}^*) = 0, \quad \forall i \in \mathcal{E}, \quad (3.28b)$$

$$c_i(\mathbf{x}^*) \geq 0, \quad \forall i \in \mathcal{I}, \quad (3.28c)$$

$$\lambda_i^* \geq 0, \quad \forall i \in \mathcal{I}, \quad (3.28d)$$

$$\lambda_i^* c_i(\mathbf{x}^*) = 0, \quad \forall i \in \mathcal{E} \cup \mathcal{I}. \quad (3.28e)$$

As condições (3.28a), (3.28b), (3.28c), (3.28d) e (3.28e) são conhecidas como as condições de *Karush–Kuhn–Tucker*, ou condições de *KKT*. Como os multiplicadores das restrições inativas vale zero, se pode reescrever

$$\mathbf{0} = \nabla_x \mathcal{L}(\mathbf{x}^*, \lambda^*) = \nabla f(\mathbf{x}^*) - \sum_{i \in \mathcal{A}(\mathbf{x}^*)} \lambda_i^* \nabla c_i(\mathbf{x}^*) \quad (3.29)$$

Um caso particular, complementar, merece sua definição:

Definição 3.7 (Complementário estrito) *Dado o ponto $\mathbf{x}^* \in \Omega$ e um vetor λ_i^* satisfazendo as condições de KKT, se diz que o complementar estrito é assegurado se exatamente um, entre λ_i^* e $c_i(\mathbf{x}^*)$, é zero para cada índice $i \in \mathcal{I}$. Em outras palavras, tem-se que $\lambda_i^* > 0$ para cada $i \in \mathcal{I} \cap \mathcal{A}(\mathbf{x}^*)$.*

A definição 3.7 expressa que se um mínimo local for condicionado por um restrição de desigualdade, então dita restrição vale zero em \mathbf{x}^* (obviamente), mas o multiplicador de Lagrange deverá ser positivo. Entretanto, se um mínimo local não for condicionado por tal restrição de desigualdade, dita restrição será positiva, mas o multiplicador de Lagrange valerá zero. A condição (3.28e) é chamada de condição complementar.

3.3.2 Derivação das condições de primeira ordem

Sequência viável e direção limitante

A primeira definição introduzida é a *sequência viável* e sua associada *direção limitante*.

Definição 3.8 (Sequência viável e direção limitante)

- (i) Uma sequência $\{\mathbf{x}_k\}_{\mathbb{N}}$ com $\mathbf{x}_k \in \mathbb{R}^n \wedge k \in \mathbb{N}$ é definida como uma sequência viável em relação a um ponto viável $\mathbf{x}^* \in \mathbb{R}^n$, se existe $k_0 \in \mathbb{N}$ tal que as seguintes propriedades são satisfeitas

$$\mathbf{x}_k \neq \mathbf{x}^*, k \in \mathbb{N} \quad (3.30a)$$

$$\lim_{k \rightarrow \infty} \mathbf{x}_k = \mathbf{x}^* \quad (3.30b)$$

$$\mathbf{x}_k \in \Omega, k \geq k_0 \quad (3.30c)$$

O conjunto de todas as sequências viáveis em relação a um ponto viável $\mathbf{x}^* \in \Omega$ são denotadas por $\mathcal{T}(\mathbf{x}^*)$.

- (ii) Um vetor $\delta \in \mathbb{R}^n$ é definido como uma direção limitante de uma sequência viável $\{\mathbf{x}_k\}_{\mathbb{N}} \subset \mathcal{T}(\mathbf{x}^*)$, se existe uma subsequência $\{\mathbf{x}_k\}_{\mathbb{N}'}$, $\mathbb{N}' \subset \mathbb{N}$ tal que

$$\lim_{k \rightarrow \infty} \frac{\mathbf{x}_k - \mathbf{x}^*}{\|\mathbf{x}_k - \mathbf{x}^*\|} = \delta, \quad \text{com } k \in \mathbb{N}' \quad (3.31)$$

Observar que uma direção limitante simplesmente expressa uma direção, a partir do ponto atual \mathbf{x}_k , na qual nenhuma restrição ativa será violada, i.e., direção pela qual se mantém a viabilidade. Outras definições podem ser encontradas, tal como, a interseção entre uma hipersfera de raio um com centro em \mathbf{x}_k com a região viável.

Definição 3.9 (Solução local) Um ponto viável $\mathbf{x}^* \in \Omega$ é solução local de (3.23), se para toda sequência viável $\{\mathbf{x}_k\}_{\mathbb{N}} \subset \mathcal{T}(\mathbf{x}^*)$ existe um $k_0 \in \mathbb{N}$ tal que

$$f(\mathbf{x}_k) \geq f(\mathbf{x}^*), k \geq k_0 \quad (3.32)$$

Teorema 3.7 (Caracterização da solução) Se $\mathbf{x}^* \in \Omega$ é solução local de (3.23), então toda sequência viável $\{\mathbf{x}_k\}_{\mathbb{N}} \subset \mathcal{T}(\mathbf{x}^*)$ satisfaz

$$\nabla f(\mathbf{x}^*)^T \delta \geq 0 \quad (3.33)$$

onde δ é qualquer direção limitante de uma sequência viável.

demonstração: Por redução ao absurdo, se assume que existe uma sequência viável $\{\mathbf{x}_k\}_{\mathbb{N}} \subset \mathcal{T}(\mathbf{x}^*)$ e uma direção limitante δ , tal que $\nabla f(\mathbf{x}^*)^T \delta < 0$, sendo $\{\mathbf{x}_k\}_{\mathbb{N}'}$ uma subsequência de $\{\mathbf{x}_k\}_{\mathbb{N}}$ em relação a um ponto viável $\mathbf{x}^* \in \Omega$. Pelo teorema de Taylor (teorema 3.1), tem-se para $k \in \mathbb{N}'$ que

$$\begin{aligned} f(\mathbf{x}_k) &= f(\mathbf{x}^*) + \nabla f(\mathbf{x}^*)^T (\mathbf{x}_k - \mathbf{x}^*) + o(\|\mathbf{x}_k - \mathbf{x}^*\|) \\ &= f(\mathbf{x}^*) + \|\mathbf{x}_k - \mathbf{x}^*\| \nabla f(\mathbf{x}^*)^T \delta + o(\|\mathbf{x}_k - \mathbf{x}^*\|) \end{aligned}$$

Como $\nabla f(\mathbf{x}^*)^T \delta < 0$ e tendo em vista que os termos de maior ordem são dominados pelo termo de primeira ordem já que existe $k \in \mathbb{N}'$, tem-se que

$$f(\mathbf{x}_k) \approx f(\mathbf{x}^*) + \|\mathbf{x}_k - \mathbf{x}^*\| \nabla f(\mathbf{x}^*)^T \delta, \quad \forall k \geq k_0$$

Logo

$$f(\mathbf{x}_k) > f(\mathbf{x}^*)$$

e \mathbf{x}^* não é uma solução local. (\mathbf{x}_{k_0} deve pertencer a um entorno aberto de \mathbf{x}^*)

3.3.3 Caracterizando as direções limitantes: qualificação das restrições

O teorema 3.7 é bastante geral, entretanto este requer o conhecimento de todas as direções limitantes para todas as sequências viáveis $\mathcal{T}(\mathbf{x}^*)$; então este não é muito útil como proposição. Nesta seção 3.3.3 é mostrado que as *qualificação das restrições* permitem caracterizar as propriedades sobressalentes de $\mathcal{T}(\mathbf{x}^*)$, e logo tornar a condição (3.33) fácil de verificar.

Uma qualificação das restrições usada frequentemente, aqui já definida, é a *qualificação das restrições linearmente independente* (LICQ) dada na definição (3.6). O seguinte lema mostra que quando a LICQ é assegurada, existe uma forma organizada para caracterizar o conjunto de todas as possíveis direções limitantes δ em termos dos gradientes $\nabla c_i(\mathbf{x}^*)$ das restrições ativas em \mathbf{x}^* .

Introduzindo a notação A , para representar a matriz na qual as linhas são os gradientes das restrições ativas no ponto de ótimo.

$$A = \left[\nabla c_i(\mathbf{x}^*)^T \right]_{i \in \mathcal{A}(\mathbf{x}^*)} \quad (3.34)$$

Lema 3.1 (Caracterização das direções limitantes) *As duas seguintes proposições são verdadeiras.*

(i) *Se $\delta \in \mathbb{R}^n$ é uma direção limitante de uma sequência viável, logo*

$$\delta^T \nabla c_i(\mathbf{x}^*) = 0, \quad \forall i \in \mathcal{E}, \quad \delta^T \nabla c_i(\mathbf{x}^*) \geq 0, \quad \forall i \in \mathcal{A}(\mathbf{x}^*) \cap \mathcal{I} \quad (3.35)$$

(ii) *Se (3.35) se mantém com $\|\delta\| = 1$ e a condição LICQ é satisfeita, então $\delta \in \mathbb{R}^n$ é uma direção limitante de alguma sequência viável.*

demonstração: Para a primeira proposição, sendo $\{\mathbf{x}_k\}_{\mathbb{N}} \subset \mathcal{T}(\mathbf{x}^*)$ uma sequência viável com uma direção limitante δ tal que $\mathbb{N}' \subset \mathbb{N}$

$$\lim_{k \in \mathbb{N}'} \frac{\mathbf{x}_k - \mathbf{x}^*}{\|\mathbf{x}_k - \mathbf{x}^*\|} = \delta$$

Isto produz

$$\mathbf{x}_k = \mathbf{x}^* + \delta \|\mathbf{x}_k - \mathbf{x}^*\| + o(\|\mathbf{x}_k - \mathbf{x}^*\|) \quad (3.36)$$

Tomando $i \in \mathcal{E}$ e usando o teorema de Taylor, tem-se que

$$\begin{aligned} 0 &= \frac{1}{\|\mathbf{x}_k - \mathbf{x}^*\|} c_i(\mathbf{x}_k) \\ &= \frac{1}{\|\mathbf{x}_k - \mathbf{x}^*\|} \left[c_i(\mathbf{x}^*) + \|\mathbf{x}_k - \mathbf{x}^*\| \nabla c_i(\mathbf{x}^*)^T \delta + o(\|\mathbf{x}_k - \mathbf{x}^*\|) \right] \\ &= \nabla c_i(\mathbf{x}^*)^T \delta + \frac{o(\|\mathbf{x}_k - \mathbf{x}^*\|)}{\|\mathbf{x}_k - \mathbf{x}^*\|} \end{aligned}$$

Tomando o limite $k \rightarrow \infty$, o último termo desaparece e tem-se que $\nabla c_i(\mathbf{x}^*)^T \delta = 0$, conforme o proposto.

Para as restrições de desigualdade ativas $i \in \mathcal{A}(\mathbf{x}^*) \cap \mathcal{I}$ tem-se

$$\begin{aligned} 0 &\leq \frac{1}{\|\mathbf{x}_k - \mathbf{x}^*\|} c_i(\mathbf{x}_k) \\ &= \frac{1}{\|\mathbf{x}_k - \mathbf{x}^*\|} \left[c_i(\mathbf{x}^*) + \|\mathbf{x}_k - \mathbf{x}^*\| \nabla c_i(\mathbf{x}^*)^T \delta + o(\|\mathbf{x}_k - \mathbf{x}^*\|) \right] \\ &= \nabla c_i(\mathbf{x}^*)^T \delta + \frac{o(\|\mathbf{x}_k - \mathbf{x}^*\|)}{\|\mathbf{x}_k - \mathbf{x}^*\|} \end{aligned}$$

Tomando o limite $k \rightarrow \infty$, o último termo desaparece e tem-se que $\nabla c_i(\mathbf{x}^*)^T \delta \geq 0$, conforme o proposto.

Para segunda preposição, se usa o teorema da função implícita. Pela definição de LICQ, tem-se que a matrix $A_{m \times n}$ das restrições ativas, possui posto completo na sua linha, i.e., posto m . Sendo Z uma matriz na qual suas colunas são uma base do espaço nulo de A (ou do núcleo de A)

$$Z \in \mathbb{R}^{n \times (n-m)}, \quad Z \text{ possui posto completo } (n-m) \text{ na sua coluna,} \quad AZ = \mathbf{0} \quad (3.37)$$

Sendo δ uma direção que satisfaz (3.35), e $\{t_k\}_{\mathbb{N}}$ uma sequência de reais positivos tal que $\lim_{k \rightarrow \infty} t_k = 0$. Definindo o sistema parametrizado $R: \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$ por

$$R(\mathbf{x}, t) = \begin{bmatrix} c(\mathbf{x}) - tA\delta \\ Z^T(\mathbf{x} - \mathbf{x}^* - t\delta) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (3.38)$$

assumindo que existe $k_0 \in \mathbb{N}$ tal que para cada $t = t_k$, $k \geq k_0$ o sistema (3.38) admite uma solução $\mathbf{x} = \mathbf{x}_k$ para um pequeno $t > 0$ dada uma sequência viável $\{\mathbf{x}_k\}_{\mathbb{N}}$ que aproxima \mathbf{x}^* .

Claramente, para $t = 0$, a solução de (3.38) é $\mathbf{x} = \mathbf{x}^*$, e o Jacobiano de R neste ponto é

$$\nabla_x R(\mathbf{x}^*, 0) = \begin{bmatrix} A \\ Z^T \end{bmatrix} \quad (3.39)$$

a qual não é singular. De acordo com o teorema da função implícita, o sistema (3.38) tem solução única para todos os valores de t_k suficientemente pequenos, i.e., existe $k_0 \in \mathbb{N}$ tal que $k \geq k_0$ e $t = t_k$ que assegura a unicidade da solução $\mathbf{x} = \mathbf{x}_k$. De (3.35) e (3.38) tem-se que

$$i \in \mathcal{E} \quad \therefore \quad c_i(\mathbf{x}_k) = t_k \nabla c_i(\mathbf{x}_k)^T \delta = 0 \quad (3.40a)$$

$$i \in \mathcal{A}(\mathbf{x}^*) \cap \mathcal{I} \quad \therefore \quad c_i(\mathbf{x}_k) = t_k \nabla c_i(\mathbf{x}_k)^T \delta \geq 0 \quad (3.40b)$$

então isto prova que \mathbf{x}_k realmente é viável.

Agora será verificado que $\mathbf{x}_k = \mathbf{x}(t_k) \neq \mathbf{x}^*$, por redução ao absurdo, assume-se $\mathbf{x}(\bar{t}) = \mathbf{x}^*$ para algum $\bar{t} > 0$, i.e.,

$$\begin{bmatrix} c(\mathbf{x}^*) - \bar{t}A\delta \\ -Z^T(\bar{t}\delta) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (3.41)$$

como $c_i(\mathbf{x}^*) = 0$ e (3.39) é não singular, a única solução $\delta = \mathbf{0}$ contradiz $\|\delta\| = 1$. Este fato apresenta δ como uma direção limitante da sequência viável $\{\mathbf{x}\}_{\mathbb{N}}$. Usando $R(\mathbf{x}_k, t_k)$

para todo k , pelo teorema de Taylor tem-se que

$$\begin{aligned} \mathbf{0} = R(\mathbf{x}_k, t_k) &= \begin{bmatrix} c(\mathbf{x}_k) - t_k A \delta \\ Z^T(\mathbf{x}_k - \mathbf{x}^* - t_k \delta) \end{bmatrix} \\ &= \begin{bmatrix} A(\mathbf{x}_k - \mathbf{x}^*) + o(\|\mathbf{x}_k - \mathbf{x}^*\|) - t_k A \delta \\ Z^T(\mathbf{x}_k - \mathbf{x}^* - t_k \delta) \end{bmatrix} \\ &= \begin{bmatrix} A \\ Z^T \end{bmatrix} (\mathbf{x}_k - \mathbf{x}^* - t_k \delta) + o(\|\mathbf{x}_k - \mathbf{x}^*\|) \end{aligned}$$

Dividindo esta expressão por $\|\mathbf{x}_k - \mathbf{x}^*\|$ e usando a não singularidade de $\begin{bmatrix} A \\ Z^T \end{bmatrix}$ tem-se que

$$\lim_{k \rightarrow \infty} \left[\delta_k - \frac{t_k}{\|\mathbf{x}_k - \mathbf{x}^*\|} \delta \right] = 0 \quad \text{com } \delta_k = \frac{\mathbf{x}_k - \mathbf{x}^*}{\|\mathbf{x}_k - \mathbf{x}^*\|} \quad (3.42)$$

Desde que $\|\delta_k\| = 1$, para todo $k \in \mathbb{N}$, e $\|\delta\| = 1$, tem-se

$$\lim_{k \rightarrow \infty} \frac{t_k}{\|\mathbf{x}_k - \mathbf{x}^*\|} = 1 \quad (3.43)$$

logo, $\lim_{k \rightarrow \infty} \delta_k = \delta$, conforme o proposto.

O conjunto de direções definido por (3.35) possuem um rol central nas condições do ponto ótimo.

Definição 3.10 (Cone tangente e normal à região viável) *Um vetor \mathbf{w} é tangente à região viável Ω em $\mathbf{x} \in \Omega$, se para toda sequência $\{\mathbf{x}_k\}_{\mathbb{N}}$, $\mathbf{x}_k \in \Omega$, $k \in \mathbb{N}$, com $\lim_{k \rightarrow \infty} \mathbf{x}_k = \mathbf{x}$ e $\{t_k\}_{\mathbb{N}}$ uma sequência de reais positivos tal que $\lim_{k \rightarrow \infty} t_k = 0$, $k \in \mathbb{N}$, existe uma sequência $\{\mathbf{w}_k\}_{\mathbb{N}}$, $\mathbf{w}_k \in \mathbb{R}^n$, $k \in \mathbb{N}$, com $\lim_{k \rightarrow \infty} \mathbf{w}_k = \mathbf{w}$ tal que $\mathbf{x}_k + t_k \mathbf{w}_k \in \Omega$, $k \in \mathbb{N}$.*

O conjunto $\mathcal{T}_\Omega(\mathbf{x})$ de todos os vetores tangentes à Ω em $\mathbf{x} \in \Omega$ é um cone, i.e., a seguinte propriedade é satisfeita

$$\mathbf{w} \in \mathcal{T}_\Omega(\mathbf{x}) \quad \Rightarrow \quad \alpha \mathbf{w} \in \mathcal{T}_\Omega(\mathbf{x}), \quad \forall \alpha > 0$$

$\mathcal{T}_\Omega(\mathbf{x})$ é chamado de cone tangente à Ω em \mathbf{x} .

O cone normal $\mathcal{N}_\Omega(\mathbf{x})$ é o complemento ortogonal ao cone tangente, ou seja

$$\mathcal{N}_\Omega(\mathbf{x}) = \{ \mathbf{v} \in \mathbb{R}^n \mid \mathbf{v}^T \mathbf{w} \leq 0, \quad \forall \mathbf{w} \in \mathcal{T}_\Omega(\mathbf{x}) \}$$

Definição 3.11 (Direção limitante e cone tangente) Dado um ponto \mathbf{x}^* e o conjunto dos índices das restrições ativas $\mathcal{A}(\mathbf{x})$, o conjunto F_1 é definido por

$$F_1 = \left\{ \alpha \delta \mid \alpha > 0, \begin{array}{ll} \delta^T \nabla c_i(\mathbf{x}^*) = 0, & \forall i \in \mathcal{E}, \\ \delta^T \nabla c_i(\mathbf{x}^*) \geq 0, & \forall i \in \mathcal{A}(\mathbf{x}^*) \cap \mathcal{I} \end{array} \right\}$$

O conjunto de vetores de direções limitantes a partir de \mathbf{x} viável é chamado de cone de direções limitantes, F_1 com $\alpha = 1$.

3.3.4 Multiplicadores de Lagrange

O lema 3.1 expressa que quando a LICQ é assegurada, o cone F_1 é o conjunto de todos os múltiplos positivos de todas as direções limitantes de todas as sequências viáveis. Ainda, a condição (3.33) do teorema 3.7, como já foi mencionado é inviável de ser verificada. O seguinte lema fornece uma alternativa prática de ser verificada, fazendo o uso dos multiplicadores de Lagrange.

Lema 3.2 (Inexistência da direção descendente) Não existe direção descendente $\delta \in F_1$ para a qual $\nabla f(\mathbf{x}^*)^T \delta < 0$, se e somente se existe um vetor $\lambda \in \mathbb{R}^m$ tal que

$$\nabla f(\mathbf{x}^*) = \sum_{i \in \mathcal{A}(\mathbf{x}^*)} \lambda_i \nabla c_i(\mathbf{x}^*) = A(\mathbf{x}^*)^T \lambda, \quad \lambda_i \geq 0 \quad \forall i \in \mathcal{A}(\mathbf{x}^*) \cap \mathcal{I} \quad (3.44)$$

demonstração: Esta demonstração exige duas demonstrações, i.e., se existe λ então não existe δ ; se não existe δ então existe λ . Definindo o cone \mathcal{N} por

$$\mathcal{N} = \left\{ s \mid s = \sum_{i \in \mathcal{A}(\mathbf{x}^*)} \lambda_i \nabla c_i(\mathbf{x}^*), \quad \lambda_i \geq 0, \quad \forall i \in \mathcal{A}(\mathbf{x}^*) \cap \mathcal{I} \right\} \quad (3.45)$$

Logo a condição (3.44) é equivalente a $\nabla f(\mathbf{x}^*) \in \mathcal{N}$.

suficiência: Assegurando (3.44) e escolhendo um δ que satisfaça (3.35) ($\delta \in F_1$), tem-se

$$\nabla f(\mathbf{x}^*)^T \delta = \sum_{i \in \mathcal{E}} \lambda_i \underbrace{\nabla c_i(\mathbf{x}^*)^T \delta}_{=0} + \sum_{i \in \mathcal{A}(\mathbf{x}^*) \cap \mathcal{I}} \lambda_i \underbrace{\nabla c_i(\mathbf{x}^*)^T \delta}_{\geq 0} \geq 0 \quad (3.46)$$

Assim resulta demonstrada a proposição à frente.

necessariedade: Verifica-se que $\nabla f(\mathbf{x}^*)$ não satisfaz (3.44), i.e, $\nabla f(\mathbf{x}^*) \notin \mathcal{N}$, logo

pode-se encontrar um vetor δ para o qual $\nabla f(\mathbf{x}^*)^T \delta < 0$ e (3.35) é assegurada. Sendo $\hat{\mathbf{s}}$ um vetor que representa a melhor aproximação de $\nabla f(\mathbf{x}^*)$ em \mathcal{N} , i.e.,

$$\|\hat{\mathbf{s}} - \nabla f(\mathbf{x}^*)\|^2 = \min_{\mathbf{s} \in \mathcal{N}} \|\mathbf{s} - \nabla f(\mathbf{x}^*)\|^2 \quad (3.47)$$

Desde que $\hat{\mathbf{s}} \in \mathcal{N}$, tem-se que $t\hat{\mathbf{s}} \in \mathcal{N}$ para todos os escalares $t \geq 0$.

Desde que $\|t\hat{\mathbf{s}} - \nabla f(\mathbf{x}^*)\|^2$ é minimizado em $t = 1$, tem-se

$$\begin{aligned} \frac{d}{dt} \|t\hat{\mathbf{s}} - \nabla f(\mathbf{x}^*)\|^2 \Big|_{t=1} = 0 & \quad \therefore \quad (-2\hat{\mathbf{s}}^T \nabla f(\mathbf{x}^*) + 2t\hat{\mathbf{s}}^T \hat{\mathbf{s}}) \Big|_{t=1} = 0 \\ & \quad \therefore \quad \hat{\mathbf{s}}^T (\hat{\mathbf{s}} - \nabla f(\mathbf{x}^*)) = 0 \end{aligned} \quad (3.48)$$

Agora, sendo \mathbf{s} outro vetor qualquer de \mathcal{N} , sendo \mathcal{N} por definição convexo, logo

$$\|\hat{\mathbf{s}} + \theta(\mathbf{s} - \hat{\mathbf{s}}) - \nabla f(\mathbf{x}^*)\|^2 \geq \|\hat{\mathbf{s}} - \nabla f(\mathbf{x}^*)\|^2, \quad \forall \theta \in [0,1]$$

e

$$2\theta(\mathbf{s} - \hat{\mathbf{s}})^T (\hat{\mathbf{s}} - \nabla f(\mathbf{x}^*)) + \theta^2 \|\mathbf{s} - \hat{\mathbf{s}}\|^2 \geq 0 \quad (3.49)$$

dividindo (3.49) por θ e levando θ ao limite $\theta \rightarrow 0^+$, tem-se $(\mathbf{s} - \hat{\mathbf{s}})^T (\hat{\mathbf{s}} - \nabla f(\mathbf{x}^*)) \geq 0$.

Logo, pela expressão (3.48)

$$\mathbf{s}^T (\hat{\mathbf{s}} - \nabla f(\mathbf{x}^*)) \geq 0, \quad \forall \mathbf{s} \in \mathcal{N} \quad (3.50)$$

Deixando $\delta = \hat{\mathbf{s}} - \nabla f(\mathbf{x}^*)$, satisfaz (3.35) e $\delta^T \nabla f(\mathbf{x}^*) < 0$, logo da expressão (3.48) tem-se

$$\delta^T \nabla f(\mathbf{x}^*) = \delta^T (\hat{\mathbf{s}} - \delta) = (\hat{\mathbf{s}} - \nabla f(\mathbf{x}^*))^T \hat{\mathbf{s}} - \delta^T \delta = -\|\delta\|^2 < 0 \quad (3.51)$$

logo δ é uma direção descendente. Analisando a expressão (3.45), obviamente tem-se

$$i \in \mathcal{E} \therefore \nabla c_i(\mathbf{x}^*) \in \mathcal{N} \quad \wedge \quad -\nabla c_i(\mathbf{x}^*) \in \mathcal{N};$$

$$i \in \mathcal{A}(\mathbf{x}^*) \cap \mathcal{I} \therefore \nabla c_i(\mathbf{x}^*) \in \mathcal{N}$$

logo, com (3.50) e $\delta = \hat{\mathbf{s}} - \nabla f(\mathbf{x}^*)$, e pela particular escolha $\mathbf{s} = \nabla c_i(\mathbf{x}^*)$ e $\mathbf{s} = -\nabla c_i(\mathbf{x}^*)$ tem-se

$$i \in \mathcal{E} \therefore \delta^T \nabla c_i(\mathbf{x}^*) \geq 0 \quad \wedge \quad -\delta^T \nabla c_i(\mathbf{x}^*) \geq 0 \therefore \delta^T \nabla c_i(\mathbf{x}^*) = 0;$$

$$i \in \mathcal{A}(\mathbf{x}^*) \cap \mathcal{I} \therefore \delta^T \nabla c_i(\mathbf{x}^*) \geq 0$$

Então, δ satisfaz a condição (3.35). Assim resulta demonstrada a preposição reversa.

3.3.5 Demonstração do Teorema 3.6

Combinando os lemas 3.1 e 3.2 se consegue produzir as condições KKT descritas no teorema 3.6. Supondo que $\mathbf{x}^* \in \mathbb{R}^n$ é um ponto viável sobre o qual a LICQ é assegurada. O teorema diz que se \mathbf{x}^* é uma solução local de (3.23), então existe um vetor $\lambda^* \in \mathbb{R}^m$ que satisfaz as condições (3.28a), (3.28b), (3.28c), (3.28d), (3.28e) (KKT).

Primeiro foi mostrado que existem multiplicadores $\lambda_i, i \in \mathcal{A}(\mathbf{x}^*)$ tal que satisfazem (3.44). O teorema 3.7 expressa que $\delta^T \nabla f(\mathbf{x}^*) \geq 0$ para todos os vetores δ que são direções limitantes de sequências viáveis. Do lema 3.1, se a LICQ é assegurada, o conjunto de todas as possíveis direções limitantes é exatamente o conjunto de todos os vetores que satisfazem (3.35). Colocando estas duas proposições juntas, se encontram todas as direções δ que satisfaçam (3.35) e ainda $\delta^T \nabla f(\mathbf{x}^*) \geq 0$. Logo do lema 3.2, tem-se que existe um vetor λ que satisfaz (3.44), conforme proposto.

Definindo o vetor λ^* por

$$\lambda^* = \begin{cases} \lambda_i, & i \in \mathcal{A}(\mathbf{x}^*) \\ 0, & \text{caso contrário} \end{cases} \quad (3.52)$$

Esta escolha, juntamente com a solução local \mathbf{x}^* satisfaz as condições KKT. Com estas considerações, se verificam, uma a uma, as condições KKT.

- A condição (3.28a) vem dada por (3.44) e as definições (3.26) da função Lagrangiana e (3.52) de λ_i^* ;
- As condições (3.28b) e (3.28c) são satisfeitas desde que \mathbf{x}^* seja um ponto viável;
- A condição (3.28d) é satisfeita, já que de (3.44) tem-se $\lambda_i^* \geq 0$ para $i \in \mathcal{A}(\mathbf{x}^*) \cap \mathcal{I}$, enquanto que de (3.52) tem-se $\lambda_i^* = 0$ para $i \in \mathcal{I} \setminus \mathcal{A}(\mathbf{x}^*)$. Logo, $\lambda_i^* \geq 0$;
- A condição (3.28e) é satisfeita, já que para $i \in \mathcal{A}(\mathbf{x}^*) \cap \mathcal{I}$ tem-se $c_i(\mathbf{x}^*) = 0$, enquanto que para $i \in \mathcal{I} \setminus \mathcal{A}(\mathbf{x}^*)$ tem-se $\lambda_i^* = 0$.

Isto completa a demonstração.

3.3.6 Condições do ponto de ótimo de segunda ordem

Na descrição das condições do ponto de ótimo de primeira ordem – condições de KKT – foi relatado como devem resultar as derivadas primeiras de f e das restrições ativas c_i sobre \mathbf{x}^* . Uma vez satisfeitas tais condições, o “movimento” sobre qualquer vetor $\mathbf{w} \in F_1$ produz um acréscimo na aproximação de primeira ordem da função objetivo, i.e., $\mathbf{w}^T \nabla f(\mathbf{x}^*) > 0$, ou este valor é mantido, i.e., $\mathbf{w}^T \nabla f(\mathbf{x}^*) = 0$.

A necessidade de usar derivadas de segunda ordem para verificar a existência um ponto de ótimo ocorre quando nada mais se pode saber, no que concerne, das derivadas de primeira ordem, i.e., quando $\mathbf{w}^T \nabla f(\mathbf{x}^*) = 0$ nada se sabe sobre o crescimento ou decréscimo da função ao longo desta direção. Logo, para completar com a caracterização do ponto de ótimo falta saber quais as implicações das derivadas segundas da função objetivo e das restrições sobre ponto de ótimo.

Condições de segunda ordem são obtidas examinando os termos das derivadas segundas na expansão de Taylor da função objetivo e das restrições, esta informação adicional resolve o problema à cerca do crescimento de f . Essencialmente, as condições de segunda ordem fornecem informações sobre a curvatura da função Lagrangiana nas direções “indecisas” – as direções $\mathbf{w} \in F_1$ tal que $f(\mathbf{x}^*) = 0$.

Como já mencionado, se assume que todas as funções envolvidas sejam suaves e pelo menos duas vezes diferenciáveis.

Dado F_1 pela definição 3.11 e algum vetor multiplicador de Lagrange λ^* satisfazendo as condições KKT, se define um subconjunto $F_2(\lambda^*)$ de F_1 por

$$F_2(\lambda^*) = \{\mathbf{w} \in F_1 \mid \nabla c_i(\mathbf{x}^*)^T \mathbf{w} = 0, \forall i \in \mathcal{A}(\mathbf{x}^*) \cap \mathcal{I} \text{ com } \lambda^* > 0\} \quad (3.53)$$

De forma equivalente

$$\mathbf{w} \in F_2(\lambda^*) \Leftrightarrow \begin{cases} \nabla c_i(\mathbf{x}^*)^T \mathbf{w} = 0, & \forall i \in \mathcal{E} \\ \nabla c_i(\mathbf{x}^*)^T \mathbf{w} = 0, & \forall i \in \mathcal{A}(\mathbf{x}^*) \cap \mathcal{I} \text{ com } \lambda^* > 0 \\ \nabla c_i(\mathbf{x}^*)^T \mathbf{w} \geq 0, & \forall i \in \mathcal{A}(\mathbf{x}^*) \cap \mathcal{I} \text{ com } \lambda^* = 0 \end{cases} \quad (3.54)$$

O conjunto $F_2(\lambda^*)$ que contém os vetores \mathbf{w} tende a manter as restrições de desigualdade ativas, para as quais a componente λ_i^* é positiva, bem como para as restrições de igualdade.

Da definição (3.54) e do fato que $\lambda_i^* = 0$ para toda componente inativa $i \in \mathcal{I} \setminus \mathcal{A}(\mathbf{x}^*)$, segue

$$\mathbf{w} \in F_2(\lambda^*) \quad \therefore \quad \lambda_i^* \nabla c_i(\mathbf{x}^*)^T \mathbf{w} = 0, \quad \forall i \in \mathcal{E} \cup \mathcal{I} \quad (3.55)$$

Logo, das condições KKT (3.28a) e pela definição da função Lagrangiana (3.26), tem-se

$$\mathbf{w} \in F_2(\lambda^*) \quad \therefore \quad \mathbf{w}^T \nabla f(\mathbf{x}^*) = \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i^* \mathbf{w}^T \nabla c_i(\mathbf{x}^*) = 0 \quad (3.56)$$

Conforme o requerido, o conjunto $F_2(\lambda^*)$ contém direções de F_1 das quais não se sabe, por informações obtidas das derivadas primeiras, se a função objetivo f irá crescer ou decrescer ao longo da mesma.

Teorema 3.8 (Condições necessárias de segunda ordem) *Supondo que $\mathbf{x}^* \in \Omega$ é um mínimo local de (3.23) e que é assegurado a LICQ em \mathbf{x}^* . Sendo $\lambda^* \in \mathbb{R}^n$ um vetor multiplicador de Lagrange tal que as condições KKT são satisfeitas, e sendo $F_2(\lambda^*)$ o conjunto definido em (3.53), então*

$$\mathbf{w}^T \nabla_{xx} \mathcal{L}(\mathbf{x}^*, \lambda^*) \mathbf{w} \geq 0, \quad \forall \mathbf{w} \in F_2(\lambda^*) \quad (3.57)$$

Em outras palavras o Hessiano da função Lagrangiana deve ser positivo semi-definido.

demonstração: A idéia desta demonstração é criar uma sequência viável $\{\mathbf{x}_k\}_{\mathbb{N}} \in \mathcal{T}(\mathbf{x}^*)$ com uma direção limitante $\frac{\mathbf{w}}{\|\mathbf{w}\|}$ tal que $f(\mathbf{x}_k) \geq f(\mathbf{x}^*)$ para $k \geq k_0$, implique em (3.57).

Desde que $\mathbf{w} \in F_2(\lambda^*) \subset F_1$, se consegue fazer uso da técnica usada na demonstração do lema 3.1 para construir uma sequência viável $\{\mathbf{x}_k\}_{\mathbb{N}} = \{\mathbf{x}(t_k)\}$ tal que

$$\lim_{k \in \mathbb{N}'} \frac{\mathbf{x}_k - \mathbf{x}^*}{\|\mathbf{x}_k - \mathbf{x}^*\|} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$$

Em particular, tem-se de (3.40a) que

$$c_i(\mathbf{x}_k) = \frac{t_k}{\|\mathbf{w}\|} \nabla c_i(\mathbf{x}^*)^T \mathbf{w}, \quad \forall i \in \mathcal{A}(\mathbf{x}^*) \quad (3.58)$$

onde $\{t_k\}_{\mathbb{N}}$ é uma sequência de escalares reais positivos decrescentes tendendo a zero.

Ainda, de (3.43) tem-se que

$$\|\mathbf{x}_k - \mathbf{x}^*\| = t_k + o(t_k) \quad (3.59)$$

e por substituição em (3.42) resulta

$$\mathbf{x}_k - \mathbf{x}^* = \frac{t_k}{\|\mathbf{w}\|} \mathbf{w} + o(t_k) \quad (3.60)$$

Das condições KKT e da expressão (3.58), tem-se

$$\begin{aligned}
 \mathcal{L}(\mathbf{x}_k, \lambda^*) &= f(\mathbf{x}_k) - \sum_{i \in \mathcal{E} \cup \mathcal{I}} \lambda_i^* c_i(\mathbf{x}_k) \\
 &= f(\mathbf{x}_k) - \sum_{i \in \mathcal{A}(\mathbf{x})} \underbrace{\lambda_i^* \nabla c_i(\mathbf{x}_k)^T \mathbf{w}}_{=0 \text{ de (3.58)}} \\
 &= f(\mathbf{x}_k)
 \end{aligned} \tag{3.61}$$

Por outro lado, a expansão de Taylor em torno de \mathbf{x}^* da função Lagrangiana produz

$$\begin{aligned}
 \underbrace{\mathcal{L}(\mathbf{x}_k, \lambda^*)}_{=f(\mathbf{x}_k) \text{ de (3.61)}} &= \underbrace{\mathcal{L}(\mathbf{x}^*, \lambda^*)}_{=f(\mathbf{x}^*) \text{ de KKT}} + (\mathbf{x}_k - \mathbf{x}^*)^T \underbrace{\nabla_x \mathcal{L}(\mathbf{x}^*, \lambda^*)}_{=0 \text{ de KKT}} \\
 &+ \frac{1}{2} (\mathbf{x}_k - \mathbf{x}^*)^T \nabla_{xx} \mathcal{L}(\mathbf{x}^*, \lambda^*) (\mathbf{x}_k - \mathbf{x}^*) + o(\|\mathbf{x}_k - \mathbf{x}^*\|^2)
 \end{aligned} \tag{3.62}$$

ainda

$$\mathcal{L}(\mathbf{x}_k, \lambda^*) = f(\mathbf{x}^*) + \frac{1}{2} (\mathbf{x}_k - \mathbf{x}^*)^T \nabla_{xx} \mathcal{L}(\mathbf{x}^*, \lambda^*) (\mathbf{x}_k - \mathbf{x}^*) + o(\|\mathbf{x}_k - \mathbf{x}^*\|^2) \tag{3.63}$$

Tomando (3.59) e (3.60), a expressão (3.62) resulta em

$$f(\mathbf{x}_k) = f(\mathbf{x}^*) + \frac{1}{2} \frac{t_k^2}{\|\mathbf{w}\|^2} \mathbf{w}^T \nabla_{xx} \mathcal{L}(\mathbf{x}^*, \lambda^*) \mathbf{w} + o(t_k^2) \tag{3.64}$$

Logo, se $\mathbf{w}^T \nabla_{xx} \mathcal{L}(\mathbf{x}^*, \lambda^*) \mathbf{w} < 0$, então $f(\mathbf{x}_k) < f(\mathbf{x}^*)$ para todo $k > k_0 \mid k_0 \in \mathbb{N}$, logo \mathbf{x}^* não é um mínimo local. Assim se pode concluir que (3.57) é satisfeita.

Requerendo que o Hessiano da função Lagrangiana $\nabla_{xx} \mathcal{L}(\mathbf{x}^*, \lambda^*)$ seja positivo definido sobre $F_2(\lambda^*)$, se constitui as condições do ponto de ótimo. Observar que a LICQ não é requerida.

Teorema 3.9 (Condições suficientes de segunda ordem) *Supondo que $\mathbf{x}^* \in \Omega$ é um ponto viável e que existe um vetor multiplicador de Lagrange $\lambda^* \in \mathbb{R}^m$ tal que as condições KKT são satisfeitas. Ainda, supondo que*

$$\mathbf{w}^T \nabla_{xx} \mathcal{L}(\mathbf{x}^*, \lambda^*) \mathbf{w} > 0, \quad \forall \mathbf{w} \in F_2(\lambda^*) \tag{3.65}$$

Então, \mathbf{x}^ é um mínimo local estrito.*

demonstração: A idéia desta demonstração é mostrar que para toda sequência viável $\{\mathbf{x}_k\}_{\mathbb{N}} \subset \mathcal{T}(\mathbf{x}^*)$ tem-se

$$f(\mathbf{x}_k) > f(\mathbf{x}^*), \quad \text{para um } k > k_0 \mid k_0 \in \mathbb{N} \tag{3.66}$$

Pelo lema 3.1 (i) e a definição 3.10 tem-se que toda direção limitante δ de $\{\mathbf{x}_k\}_{\mathbb{N}}$ satisfaz $\delta \in F_1$. Deixando δ ser uma direção limitante e $\mathbb{N}' \subset \mathbb{N}$ para todo $k \in \mathbb{N}$ tem-se que

$$\mathbf{x}_k - \mathbf{x}^* = \|\mathbf{x}_k - \mathbf{x}^*\| \delta + o(\|\mathbf{x}_k - \mathbf{x}^*\|) \quad (3.67)$$

Da definição da função Lagrangiana tem-se

$$\mathcal{L}(\mathbf{x}_k, \lambda^*) = f(\mathbf{x}_k) - \sum_{i \in \mathcal{A}(\mathbf{x}^*)} \lambda_i^* c_i(\mathbf{x}_k) \leq f(\mathbf{x}_k) \quad (3.68)$$

enquanto a expansão de Taylor (3.63) do teorema 3.8 se mantém verdadeira.

Agora, dividindo a demonstração em dois casos $\delta \notin F_2(\lambda^*)$ e $\delta \in F_2(\lambda^*)$.

Caso I: $\delta \notin F_2(\lambda^*)$

No que concerne à expressão (3.54), existe um índice $j \in \mathcal{A}(\mathbf{x}^*) \cap \mathcal{I}$ tal que esta expressão se mantém estritamente positiva, i.e.,

$$\lambda_j^* \nabla c_j(\mathbf{x}^*)^T \delta > 0 \quad (3.69)$$

enquanto que para os índices $i \in \mathcal{A}(\mathbf{x}^*)$, tem-se

$$\lambda_i^* \nabla c_i(\mathbf{x}^*)^T \delta \geq 0 \quad (3.70)$$

Pelo teorema de Taylor, e da expressão (3.67) tendo $k \in \mathbb{N}$, para o índice j tem-se

$$\begin{aligned} \lambda_j^* c_j(\mathbf{x}_k) &= \underbrace{\lambda_j^* c_j(\mathbf{x}^*)}_{=0} + \lambda_j^* \nabla c_j(\mathbf{x}^*)^T (\mathbf{x}_k - \mathbf{x}^*) + o(\|\mathbf{x}_k - \mathbf{x}^*\|) \\ &= \|\mathbf{x}_k - \mathbf{x}^*\| \lambda_j^* \nabla c_j(\mathbf{x}^*)^T \delta + o(\|\mathbf{x}_k - \mathbf{x}^*\|) \end{aligned}$$

Logo, da expressão (3.68)

$$\begin{aligned} \mathcal{L}(\mathbf{x}_k, \lambda^*) &= f(\mathbf{x}_k) - \sum_{i \in \mathcal{A}(\mathbf{x}^*)} \lambda_i^* c_i(\mathbf{x}_k) \\ &\leq f(\mathbf{x}_k) - \|\mathbf{x}_k - \mathbf{x}^*\| \lambda_j^* \nabla c_j(\mathbf{x}^*)^T \delta + o(\|\mathbf{x}_k - \mathbf{x}^*\|) \end{aligned} \quad (3.71)$$

Pela expansão de Taylor da expressão (3.63), se mantém

$$\mathcal{L}(\mathbf{x}_k, \lambda^*) = f(\mathbf{x}^*) + O(\|\mathbf{x}_k - \mathbf{x}^*\|^2) \quad (3.72)$$

O termo $O(\|\mathbf{x}_k - \mathbf{x}^*\|^2)$ é positivo, conforme (3.65).

Combinando as expressões (3.72) e (3.71)

$$f(\mathbf{x}_k) \geq f(\mathbf{x}^*) + \|\mathbf{x}_k - \mathbf{x}^*\| \lambda_j^* \nabla c_j(\mathbf{x}^*)^T \delta + o(\|\mathbf{x}_k - \mathbf{x}^*\|) \quad (3.73)$$

Logo, por (3.69) tem-se $f(\mathbf{x}_k) > f(\mathbf{x}^*)$ para todo $k \in \mathbb{N}$

Caso II: $\delta \in F_2(\lambda^*)$ Combinando as expressões (3.67), (3.68) e (3.63) resulta

$$\begin{aligned} f(\mathbf{x}_k) &\geq f(\mathbf{x}^*) + \frac{1}{2}(\mathbf{x}_k - \mathbf{x}^*)^T \nabla_{xx} \mathcal{L}(\mathbf{x}^*, \lambda^*)(\mathbf{x}_k - \mathbf{x}^*) + o(\|\mathbf{x}_k - \mathbf{x}^*\|^2) \\ &= f(\mathbf{x}^*) + \frac{1}{2}\|\mathbf{x}_k - \mathbf{x}^*\|^2 \underbrace{\delta^T \nabla_{xx} \mathcal{L}(\mathbf{x}^*, \lambda^*) \delta}_{>0 \text{ de (3.65)}} + o(\|\mathbf{x}_k - \mathbf{x}^*\|^2) \end{aligned} \quad (3.74)$$

Logo, tem-se $f(\mathbf{x}_k) > f(\mathbf{x}^*)$ para todo $k \in \mathbb{N}$

Finalmente, desde que \mathbf{x}_k , com $k \in \mathbb{N}$ pode ser designado por uma subsequência viável $\mathbb{N}' \subset \mathbb{N}$ tal que $\{\mathbf{x}_k\}_{\mathbb{N}'}$ converge com uma direção limitante δ , tem-se que $f(\mathbf{x}_k) > f(\mathbf{x}^*)$ para todo $k \geq k_0$.

3.3.7 Problemas não suaves

O fato da função objetivo e as restrições serem suaves é uma importante questão na caracterização da solução, assim como nos problemas de minimização sem restrições. Este assegura que a função objetivo e as restrições apresentem um comportamento razoavelmente previsível e permite que os algoritmos façam uma boa escolha na busca linear.

O problema da não suavidade das restrições pode ser facilmente contornado pela substituição da restrição não suave por várias restrições suaves. Assim como na otimização sem restrições de problemas não suaves se pode reformular o problema, de maneira tal que resulte um problema de otimização com restrições, onde todas as funções envolvidas sejam suaves (estes problemas ocorrem quando f é um máximo de um conjunto de funções).

3.3.8 Sensibilidade

A conveniência de usar multiplicadores de Lagrange pode não ser clara, entretanto o rol destes tem um significado bastante intuitivo. Cada valor dos multiplicadores de Lagrange λ_i^* , expressa a *sensibilidade* do valor ótimo $f(\mathbf{x}^*)$ na presença das restrições c_i . Em outras palavras, λ_i^* indica o quão forte f está “empurrando” ou “puxando” contra a particular restrição c_i .

Escolhendo uma restrição inativa $i \notin \mathcal{A}(\mathbf{x}^*)$ tal que $c_i(\mathbf{x}^*) > 0$, a solução \mathbf{x}^* e o valor da função $f(\mathbf{x}^*)$ são indiferentes à presença da restrição. Então da expressão (3.28e) tem-se que $\lambda_i^* = 0$, o multiplicador de Lagrange indica que tal restrição é insignificante. Perturbando esta restrição inativa, esta ainda continua inativa e \mathbf{x}^* continua sendo solução do problema de otimização.

Supondo agora que esta restrição i é ativa, $i \in \mathcal{A}(\mathbf{x}^*)$, e se perturba tal no sentido oposto do gradiente da restrição (isto significa que a perturbação desloca a fronteira da restrição para o lado da região inviável). Isto produz que $c_i(\mathbf{x}^*) \geq -\epsilon \|\nabla c_i(\mathbf{x}^*)\|$ em vez de $c_i(\mathbf{x}^*) \geq 0$. Supondo que a perturbação ϵ seja suficientemente pequena para que a solução $\mathbf{x}^*(\epsilon)$ mantenha o mesmo conjunto de restrições ativas, e que os multiplicadores de Lagrange não sejam muito afetados pela perturbação. Logo, tem-se que

$$\begin{aligned} -\epsilon \|\nabla c_i(\mathbf{x}^*)\| &= c_i(\mathbf{x}^*(\epsilon)) - c_i(\mathbf{x}^*) \approx (\mathbf{x}^*(\epsilon) - \mathbf{x}^*)^T \nabla c_i(\mathbf{x}^*) \\ 0 &= c_j(\mathbf{x}^*(\epsilon)) - c_j(\mathbf{x}^*) \approx (\mathbf{x}^*(\epsilon) - \mathbf{x}^*)^T \nabla c_j(\mathbf{x}^*) \\ &\quad \forall j \in \mathcal{A}(\mathbf{x}^*) \mid j \neq i \end{aligned}$$

De maneira análoga, o valor de $f(\mathbf{x}^*(\epsilon))$ pode ser estimado pela expressão (3.28a). Logo

$$\begin{aligned} f(\mathbf{x}^*(\epsilon)) - f(\mathbf{x}^*) &\approx (\mathbf{x}^*(\epsilon) - \mathbf{x}^*)^T \nabla f(\mathbf{x}^*) \\ &= \sum_{j \in \mathcal{A}(\mathbf{x}^*)} \lambda_j^* (\mathbf{x}^*(\epsilon) - \mathbf{x}^*)^T \nabla c_j(\mathbf{x}^*) \\ &\approx -\epsilon \|\nabla c_i(\mathbf{x}^*)\| \lambda_i^* \end{aligned}$$

Tomando o limite em ϵ , a família de soluções, satisfaz

$$\frac{df(\mathbf{x}^*(\epsilon))}{d\epsilon} = -\lambda_i^* \|\nabla c_i(\mathbf{x}^*)\| \quad (3.75)$$

Na análise de sensibilidade se pode concluir que se o valor $-\lambda_i^* \|\nabla c_i(\mathbf{x}^*)\|$ é grande, então o valor do ótimo é sensível ao deslocamento da i -ésima restrição. Caso $\lambda_i^* = 0$ para uma dada restrição i , a perturbação nesta, provavelmente não afetará o ótimo.

A análise de sensibilidade motiva a seguinte definição, a qual qualifica as restrições tendo ou não seu correspondente multiplicador de Lagrange igual a zero.

Definição 3.12 (Classificação das restrições de desigualdade) *Sendo \mathbf{x}^* a solução do problema (3.23), e supondo que as condições KKT são satisfeitas. Se diz que uma*

restrição de desigualdade c_i é fortemente ativa se $i \in \mathcal{A}(\mathbf{x}^*)$ e $\lambda_i^* > 0$. Se diz que uma restrição de desigualdade c_i é fracamente ativa se $i \in \mathcal{A}(\mathbf{x}^*)$ e $\lambda_i^* = 0$.

Observar que a análise de sensibilidade independe do fator de escala da restrição individual, i.e., multiplicando c_i por a , o novo problema será equivalente, já que o multiplicador de Lagrange resultará dividido por a e o gradiente da restrição resultará multiplicado por a , assim mantendo o produto $\lambda_i^* \|\nabla c_i(\mathbf{x}^*)\|$ idêntico. Entretanto, o mesmo não ocorre se o fator de escala é aplicado à função objetivo, para um fator de escala a os multiplicadores de Lagrange resultarão multiplicados por a e a sensibilidade à perturbação será multiplicada por a .

3.4 Programação quadrática

O problema de programação quadrática é caracterizado por uma função objetivo quadrática e restrições lineares

$$\min_{\mathbf{x}} q(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{G} \mathbf{x} + \mathbf{x}^T \delta \quad (3.76a)$$

$$\text{sujeito a } a_i^T \mathbf{x} = b_i, \quad i \in \mathcal{E} \quad (3.76b)$$

$$\text{e } a_i^T \mathbf{x} \geq b_i, \quad i \in \mathcal{I} \quad (3.76c)$$

onde \mathbf{G} é uma matriz simétrica $n \times n$, \mathcal{E} e \mathcal{I} são os conjuntos finitos de índices, e δ , \mathbf{x} e $\{a_i\}$, $i \in \mathcal{E} \cup \mathcal{I}$ são vetores de n componentes, b_i são escalares. Se a matrix \mathbf{G} , for positiva semi-definida, então o problema (3.76a) é *convexo* e existe apenas um mínimo. Existem algoritmos para problemas convexos, muito eficientes, possuindo um número finito de iterações, e ainda detectam os casos sem solução.

Segundo Nocedal e Wright (1999), este algoritmo é entendido em três etapas

- As restrições de igualdade são eliminadas pela substituição destas na função objetivo (problema padrão), i.e., as restrições de igualdade simplesmente são uma dependência entre as variáveis;
- Por simples operações algébricas, se consegue reduzir o problema a um problema

linear, ainda este é chamado de *problema de complementaridade linear*, já que a função objetivo toma a forma (3.28e);

- Recupera-se a solução para o espaço original com o computo das variáveis eliminadas.

3.5 Programação quadrática sequencial – SQP –

Os algoritmos de programação quadrática sequencial ainda são os algoritmos mais utilizados e mais robustos para problemas de programação não-linear. O algoritmo foi proposto por Wilson, e posteriormente o algoritmo foi abordado em uma forma quase-Newton. Han obteve um algoritmo convergente e Powell demonstrou que tal era superlinear.

O método *SQP* do acrônimo em inglês *sequential quadratic programming* surge da aplicação do método de Newton à função Lagrangiana, com $\delta_k = \mathbf{x}_k - \mathbf{x}_{k-1}$ e $\Delta\lambda_k = \lambda_k - \lambda_{k-1}$, i.e.,

$$\begin{bmatrix} \nabla_{xx}\mathcal{L}(\mathbf{x}_k, \lambda_k) & \nabla_{\lambda x}\mathcal{L}(\mathbf{x}_k, \lambda_k) \\ \nabla_{x\lambda}\mathcal{L}(\mathbf{x}_k, \lambda_k) & \nabla_{\lambda\lambda}\mathcal{L}(\mathbf{x}_k, \lambda_k) \end{bmatrix} \begin{bmatrix} \delta_k \\ \Delta\lambda_k \end{bmatrix} = - \begin{bmatrix} \nabla_x\mathcal{L}(\mathbf{x}_k, \lambda_k) \\ \nabla_\lambda\mathcal{L}(\mathbf{x}_k, \lambda_k) \end{bmatrix} \quad (3.77)$$

Denotando por W a componente $\nabla_{xx}\mathcal{L}(\mathbf{x}_k, \lambda_k)$ e sendo A definida por (3.34), a expressão (3.77) resulta

$$\begin{bmatrix} W & A^T \\ A & \mathbf{0} \end{bmatrix} \begin{bmatrix} \delta_k \\ \Delta\lambda_k \end{bmatrix} = - \begin{bmatrix} \nabla f(\mathbf{x}_k) \\ c_i(\mathbf{x}_k) \end{bmatrix} \quad (3.78)$$

Escrevendo as condições de Karush-Kuhn-Tucker, chega-se na seguinte expressão

$$\min_{\delta_k} \frac{1}{2} \delta_k^T \mathbf{W} \delta_k + \nabla f(\mathbf{x}_k)^T \delta_k \quad (3.79a)$$

$$\text{sujeito a } \nabla c_i(\mathbf{x}_k)^T \delta_k + c_i(\mathbf{x}_k) = 0 \quad (3.79b)$$

Observar que (3.79a) e (3.79b) formam o problema de programação quadrática no ponto \mathbf{x}_k . Executando este problema sequencialmente, tem-se a programação quadrática sequencial. Em geral, este método é empregado com a abordagem quase-Newton. Obviamente, este problema foi apresentado apenas para restrições de igualdade. Se este for o caso, então o problema se resume na obtenção da solução de um sistema de

equações lineares. Caso existam restrições de desigualdade, este resultaria em um sistema de inequações.

Para contornar este problema basta reescrever o problema da forma mais geral, conforme (3.23), e aproximar a função objetivo à uma forma quadrática e as restrições, tanto as de igualdade como as de desigualdade, à uma forma linear, e assim resolver o problema proposto na programação quadrática, i.e.,

$$\min_{\delta_k} \frac{1}{2} \delta_k^T \mathbf{W} \delta_k + \nabla f(\mathbf{x}_k)^T \delta_k \quad (3.80a)$$

$$\text{sujeito a } c_i(\mathbf{x}_k) + \nabla c_i(\mathbf{x}_k)^T \delta_k = 0, \quad i \in \mathcal{E} \quad (3.80b)$$

$$\text{e } c_i(\mathbf{x}_k) + \nabla c_i(\mathbf{x}_k)^T \delta_k \geq 0, \quad i \in \mathcal{I} \quad (3.80c)$$

Este subproblema, apenas resolve a obtenção da direção de busca. Logo, definida a direção de busca Wilson, Han e Powell sugeriram utilizar como função objetivo na busca linear, a função de *penalidade* exata, p . Esta função é definida por

$$p(\mathbf{x}) = f(\mathbf{x}) + \sum_{i \in \mathcal{A}(\mathbf{x}^*)} r_i \|c_i(\mathbf{x})\| \quad (3.81)$$

onde os r_i , fatores de penalidade, são definidos como $r_i \geq \lambda_i$. Esta função objetivo não tem apenas como finalidade diminuir o valor de f , mas também diminuir o grau de violação das restrições. Isto se consegue por meio da penalização incluída na função objetivo. Os r_i indicam o grau de penalização sobre cada restrição, i.e, ao violar uma restrição a função objetivo é penalizada, e maior será a penalização quanto maior for o multiplicador de Lagrange para tal restrição. Observar que o uso da função de penalidade representa uma minimização sem restrições, i.e., o problema restrito se tornou um problema irrestrito.

3.6 Algoritmo de otimização – SQP –

O algoritmo necessita de um ponto de partida \mathbf{x}_0 e uma matriz \mathbf{B}_0 simétrica e positivo definida como aproximação do Hessiano da função Lagrangiana. Em linhas gerais o algoritmo desenvolve-se da seguinte maneira

- **Passo 1:** Obter os vetores δ e λ no ponto \mathbf{x}_0 , a partir da solução do subproblema de programação quadrática (3.80a, 3.80b, 3.80c);

- **Passo 2:** Verificar os critérios de parada;

$$\begin{cases} \|\nabla f(\mathbf{x}_k)^T \delta_k\| < tol_1 \\ \max c_i(\mathbf{x}_k) < tol_2 \\ \sum c_i(\mathbf{x}_k) < tol_3 \end{cases} \quad (3.82)$$

onde o primeiro critério verifica o decréscimo de f na direção δ , o segundo critério verifica a restrição mais violada, e o terceiro critério verifica a soma das restrições violadas.

- **Passo 3:** Fazer a busca linear utilizando a função $p(\mathbf{x})$ definida em (3.81), sobre a direção δ , assim determinando o passo a ser efetuado;
- **Passo 4:** Atualizar a aproximação do Hessiano da função Lagrangiana, **BFGS**;
- **Passo 5:** Voltar ao **Passo 1** com o novo \mathbf{x}_k .

Em geral a aproximação do Hessiano da função Lagrangiana, começa com uma matriz com componentes proporcionais à matriz identidade, i.e.,

$$\mathbf{B}_0 = b_0 \mathbf{I} \quad (3.83)$$

onde b_0 é definida pelo usuário do programa; ainda a matriz \mathbf{B} deve ser reinicializada para não se ter influência de pontos muito distantes do ponto corrente. O reinício periódico é definido pelo usuário do programa.

Ainda, como dado de entrada é fornecido o vetor de fatores de escala para a variável. Este não pode ser modificado no andamento do código.

Nos exemplos apresentados no capítulo 4, se fez uso da subrotina DNCONF da biblioteca matemática IMSL, para Fortran. Esta é uma subrotina de SQP, baseada no código em Fortran desenvolvido por Schittkowski (1986), onde os gradientes são estimados por diferenças finitas.

Capítulo 4

Otimização de forma de cascas via deformação livre de forma baseado em NURBS

4.1 Introdução

Cascas pertencem ao tipo de elemento estrutural mais comum na natureza e eficiente em relação à tecnologia. Elas são utilizadas quando alta resistência, grandes vãos e quantidade mínima de materiais são requeridas ou então com função de contenção. Neste contexto, elas podem também ser denominadas estruturas ótimas, apresentando excelente performance estrutural e em muitos casos até arquitetonicamente belas. Entretanto, como consequência de sistemas otimizados, cascas podem ser extremamente sensíveis no que concerne ao seu comportamento mecânico bem como sua aparência estética. Qualquer projeto deve considerar esta sensibilidade para se tornar bem sucedido. Além da distribuição da espessura, toda a forma da casca está diretamente relacionada com esta sensibilidade.

É bem sabido que grandes reservatórios de paredes muito finas dependem de um princípio de carga baseado em um estado de tensões de membrana onde devem ser evitadas flexões ao máximo possível. Além disso, o estado de tensão deve refletir as características do material escolhido: uma casca de concreto armado deve estar sob compressão principalmente. Esta situação ideal, é claro, raramente pode ser obtida, se uma forma de casca é adotada intuitivamente. Neste caso, são necessários elementos estruturais

adicionais como reforços, pré-tensão, enrijecedores, etc., para colocar a casca na posição desejada. Em contraste, formas estruturais ou naturais tentam evitar a maior parte destes componentes de rigidez extra. Eles são obtidos através de uma aproximação inversa, na qual o objetivo de uma desejada resposta estrutural é prescrita, e.g. a distribuição da espessura e forma. Um dos métodos descritos, chamado otimização estrutural, aparenta ser a técnica mais versátil e generalista como ferramenta de projeto. Esta afirmação se baseia no fato que cada projeto segue essencialmente um processo de otimização. Especificamente no que concerne à encontrar a forma de uma estrutura de casca o objetivo pode ser definido como segue, i.e., encontrar a distribuição da espessura e forma de uma casca tal que:

- as condições de contorno e todos os possíveis casos de carregamento são considerados;
- propriedades dos materiais são levados em consideração;
- tensões e deslocamentos são limitados a certos valores;
- um estado de tensão de membrana praticamente uniforme é obtido;
- flambagem, deformações excessivas, e efeitos negativos são evitados;
- uma razoável vida útil (durabilidade) é garantida;

e se espera

- que custos de produção e serviços sejam justificados, e que o resultado seja esteticamente agradável.

Otimização estrutural é atualmente entendida como uma síntese de diversas disciplinas individuais como o método dos elementos finitos, diferenciação automática (baseada na teoria de grafos), NURBS (projeto auxiliado por computador – *Computer Aided Geometry Design – CAGD* –), otimização matemática, etc. Aparentemente, é um método computacional, conseqüentemente apenas aqueles requerimentos descritos acima podem ser parte do processo, o qual pode ser introduzido em uma formulação matemática.

Infelizmente, o termo “otimização” também é ilusório, uma vez que ele sugestiona que exista apenas uma única solução ótima. Em outras palavras, a criatividade é ainda mantida como parte do processo; felizmente, o processo é – nesta perspectiva generalizada – extremamente sensível aos parâmetros e permite uma ampla liberdade quanto ao projeto e *design*. Otimização estrutural é nada mais que um apoio ao projeto. Suas aplicações podem ser classificadas como:

- problemas de homogenização (estados de tensões uniformes etc.);
- uso otimizado de materiais (capacidade máxima de carregamento etc.);
- resposta estrutural otimizada (e.g. frequências).

4.2 Definição do problema de otimização estrutural

As declarações de um problema matemático abstrato de otimização não linear do tipo estudado no capítulo 3, têm de ser redefinidos em termos mecânicos. As funções e variáveis típicas em otimização de forma de cascas segundo Ramm et al. (1993) são:

variáveis de otimização:

- coordenadas dos nós selecionados;
- espessura dos nós selecionados;
- outros parâmetros da seção transversal.

No presente trabalho são propostas as seguintes variáveis de otimização:

- pontos de controle em coordenadas homogêneas $\mathbf{P}_{i,j}^w$, i.e., pontos de controle e pesos;
- parâmetros que descrevem a variação espessura.

funções objetivo:

- massa ou volume $f_w = \int \rho \, dV$; onde ρ = massa específica;

- energia de deformação $f_E = \frac{1}{2} \int \sigma : \epsilon \, dV$; onde σ, ϵ são os tensores de tensões e deformações, respectivamente;
- nível de tensão $f_s = \int (\sigma - \sigma_{avg})^2 \, dA$; onde σ_{avg} é um nível de tensão objetivo;
- função de ajuste $f_\lambda = \sum \frac{(\lambda - \lambda_{obj})^2}{\lambda_{obj}^2}$; onde λ_{obj} é o ajuste à uma determinada resposta;
- frequência natural $f_\omega = \omega$; onde ω é uma das frequências naturais da estrutura;
- fator de carga crítico $f_\lambda = -\lambda$; onde λ é o fator de carga crítico.

restrições:

- massa ou volume $c_w = 1 - \frac{w}{w_{all}} = 0$; onde w_{all} é o peso ou a massa da estrutura;
- deslocamentos $c_u = 1 - \frac{u}{u_{all}} \geq 0$; onde u é um deslocamento máximo permitido;
- tensões $c_s = 1 - \frac{\sigma}{\sigma_{all}} \geq 0$; onde σ_{all} é um nível de tensão máximo permitido;
- frequências $c_\omega = 1 - \frac{\omega}{\omega_{all}} \geq 0$; onde ω_{all} é uma frequência máxima permitida;
- fator de carga $c_\lambda = \frac{\lambda}{\lambda_{all}} - 1 \geq 0$; onde λ_{all} é um fator de carga crítico máximo permitido.

Além do objetivo mais comum, o peso, há outras funções objetivos de significância, como minimização de energia de deformação, o que é equivalente a maximizar a rigidez. Isso significa que as deformações por flexão em uma casca são minimizadas a um ponto em que são obtidas tensões de membrana. Nível de tensão baseado em uma tensão “alvo” prescrita pode ser aplicado para gerar uma casca principalmente em compressão. Ajuste para uma dada resposta, por exemplo, uma frequência específica ou um desejado espectro ou maximizar a carga de falha são objetivos clássicos em engenharia. Restrições de desigualdade são levadas em conta para analisar requisitos de segurança e confiabilidade as quais devem ser satisfeitas. Restrições típicas deste tipo são limites de tensões e deslocamentos. Se a rigidez ou o fator de carga crítica deve ser maximizado, uma massa estrutural prescrita é introduzida através de uma restrição de igualdade. Esta restrição proíbe um acúmulo de massa, o qual, de outra forma, produziria uma estrutura massiva irreal.

4.3 Algoritmo de otimização de forma de cascas via deformação livre de forma baseado em NURBS

O algoritmo desenvolvido é composto por três sub algoritmos básicos; estes são:

- Otimização – SQP –
- Mudança de forma – deformação livre de forma baseado em NURBS –
- Método de elementos finitos – elemento de casca –

O código em Fortran, em linhas gerais, se desenvolve da seguinte maneira: o método de elementos finitos é utilizado para avaliar a função objetivo, a mudança de forma é utilizada para deslocar a variável \mathbf{x} de \mathbf{x}_k para \mathbf{x}_{k+1} e a otimização fornece o vetor deslocamento de \mathbf{x}_k para \mathbf{x}_{k+1} . O vetor gradiente é estimado da seguinte maneira: avalia-se a função objetivo em um determinado ponto, e se perturba (individualmente) cada x_i de \mathbf{x} e por diferenças finitas se estima cada derivada parcial, assim formando o “gradiente”.

A entrada de dados é constituída pelos dados para o problema de elementos finitos, para a mudança de forma e para o processo de otimização.

Para o problema de elementos finitos os dados são:

- Malha da geometria;
- Dados do material, módulo de elasticidade, E , coeficiente de Poisson, ν , e massa específica ρ ;
- Condições de contorno: deslocamentos prescritos e esforços aplicados;

Para a mudança de forma os dados são:

- Dados da geometria: graus, p e q , pontos de controle, $\mathbf{P}_{i,j}$, pesos, $w_{i,j}$, (ou $\mathbf{P}_{i,j}^w$), vetores dos nós, \mathcal{U} e \mathcal{V} ;
- Malha do problema de elementos finitos no espaço paramétrico;

- Índices dos pontos de controle, $\mathbf{P}_{i,j}^w$, relativos aos $\mathbf{P}_{i,j}^w$ que se permite deslocar.

Para o problema de otimização os dados são:

- Variáveis de otimização $\mathbf{P}_{i,j}^w$ e em alguns problemas os parâmetros que descrevem a variação da espessura;
- Limite inferior e superior de cada variável;
- Fator de escala;
- Número máximo de iterações;
- Função objetivo (alguma das listadas anteriormente);
- Restrições de igualdade e desigualdade.

O código desenvolvido em Fortran se baseia em um algoritmo bastante simples. O esquema do algoritmo é apresentado na figura 4.1.

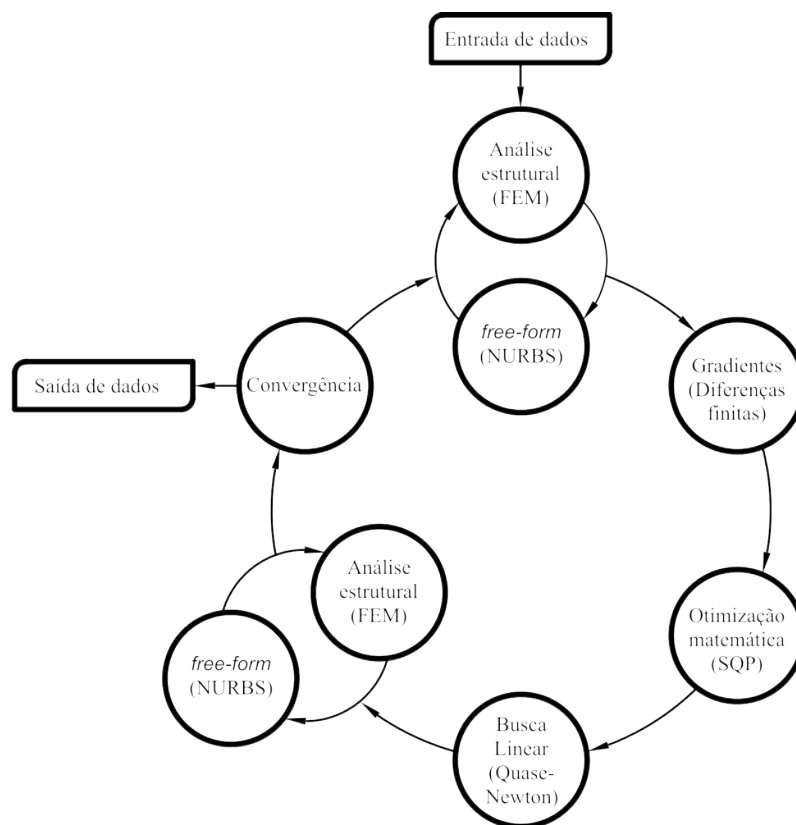


Figura 4.1: Esquema do algoritmo.

Após a entrada de dados (primeira etapa da figura 4.1) já descrita, perturba-se cada uma das variáveis x_i , sendo estas: parâmetros relativos à espessura, pontos de controle da geometria e seus respectivos pesos (segunda etapa da figura 4.1). Para cada perturbação se executa a análise de elementos finitos. Com estas informações são avaliadas as restrições, a função objetivo, e os respectivos gradientes são estimados por diferenças finitas (terceira etapa da figura 4.1). Estima-se o Hessiano BFGS e com os gradientes obtidos no passo anterior se determina a direção de busca (quarta etapa da figura 4.1). Com a direção de busca definida se minimiza a função $p(\mathbf{x})$ (3.81) nesta direção (quinta etapa da figura 4.1). Assim resulta determinado o *passo* a ser dado na direção de busca. Obtem-se a nova geometria e são avaliadas as restrições e a função objetivo (sexta etapa da figura 4.1). Para concluir são verificados os critérios de convergência; se não satisfeitos, realiza-se um novo ciclo, i.e., se parte para uma nova iteração. Caso os critérios de convergência forem satisfeitos efetua-se a saída de dados.

Como saída de dados se tem:

- Valor inicial e final da função objetivo, tensões, deformações, deslocamentos generalizados;
- Parametrização final da geometria;
- Parâmetros iniciais e finais que descrevem a variação da espessura;
- Malha final da geometria.

4.4 Exemplos de otimização de forma de cascas via deformação livre de forma baseado em NURBS

Alguns exemplos simples foram analisados com a finalidade de validar o código desenvolvido em Fortran de otimização de forma de cascas via deformação livre de forma baseado em NURBS. Nada se garante sobre a convexidade das funções objetivo utilizadas nos exemplos a seguir.

4.4.1 Exemplo 1

O primeiro exemplo trata da minimização do momento fletor de um arco circular. O carregamento imposto ao arco baseia-se em uma carga distribuída uniformemente de $1 \frac{N}{m}$ sobre uma linha situada no topo do arco com direção vertical, conforme apresentado na figura 4.2c. O arco está apoiado na extremidades.

A estrutura é de aço, onde foi considerado um módulo de elasticidade, $E = 205 \text{ GPa}$, coeficiente de Poisson, $\nu = 0,2$. Ainda a estrutura em arco possui um vão de dois metros e uma altura de um metro (raio de um metro), a largura vale $0,1 \text{ m}$ e a espessura $0,01 \text{ m}$.

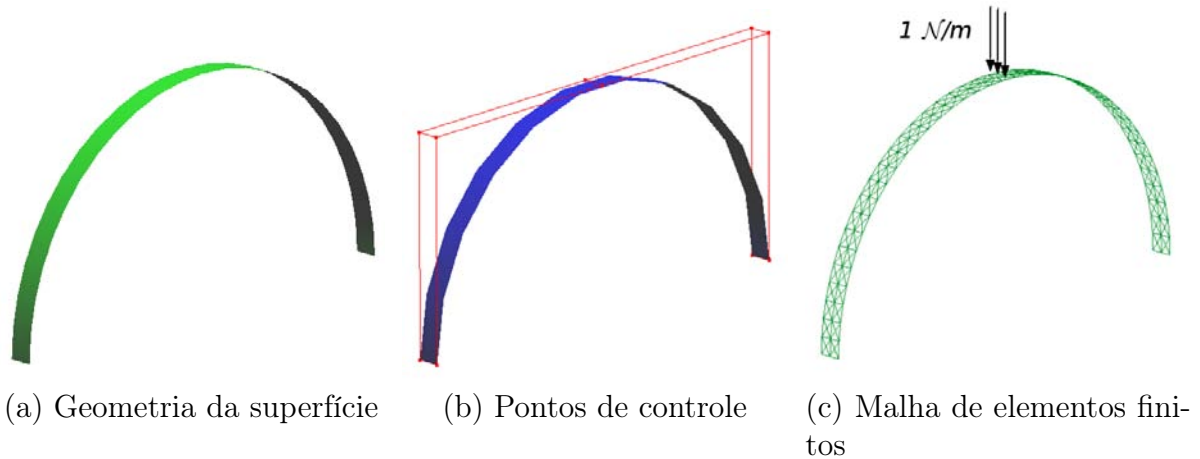


Figura 4.2: Exemplo 1.

Os pontos de controle da geometria são definidos como as variáveis do problema de otimização.

Formalmente a função objetivo trata de minimizar a diferença entre as tensões na face superior e inferior da casca, i.e.,

$$f(\sigma) = \sum_{i=1}^{nnos} \left\{ \left(\sigma_1^{sup} - \sigma_1^{inf} \right)^2 + \left(\sigma_2^{sup} - \sigma_2^{inf} \right)^2 \right\} \quad (4.1)$$

onde $nnos$ é o número de nós, sup faz referência à face superior e inf à face inferior, σ_1 faz referência à direção e_1 e σ_2 à direção e_2 , do elemento. As restrições impostas ao problema são: a altura deve permanecer constante, impõem-se restrições de simetria e a área em planta permanece constante.

Após seis iterações foi dada a convergência, com função objetivo $f_0(\sigma) = 0,6185 \cdot 10^9$ passando para $f_6(\sigma) = 0,3428 \cdot 10^2$. A geometria final obtida é apresentada na figura 4.3.

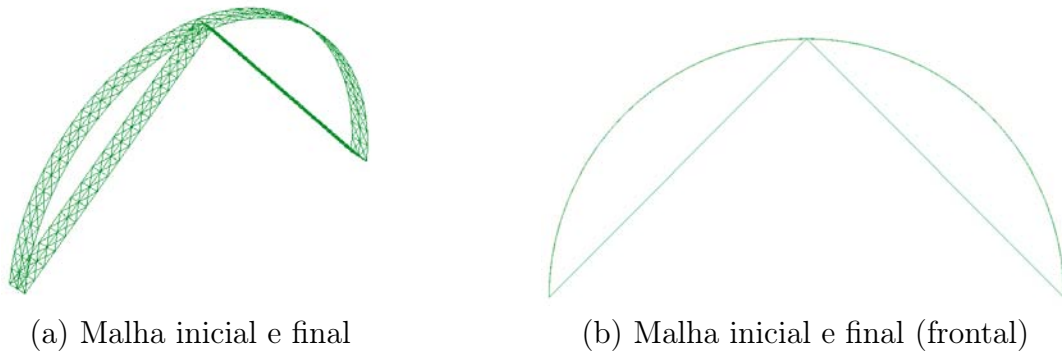


Figura 4.3: Geometria final 1.

Este problema possui solução analítica conhecida, sabe-se bem que as treliças estão isentas de momento fletor. O resultado que já era de se esperar foi confirmado; por mera inspeção visual se percebe que a geometria resultou ser um triângulo.

4.4.2 Exemplo 2

O segundo exemplo é idêntico ao primeiro exemplo no que concerne à geometria e ao problema de otimização, entretanto foi retirada a carga no topo e considerada apenas a carga de peso próprio, referente a um peso específico $\gamma = 78,6 \frac{kN}{m^3}$.

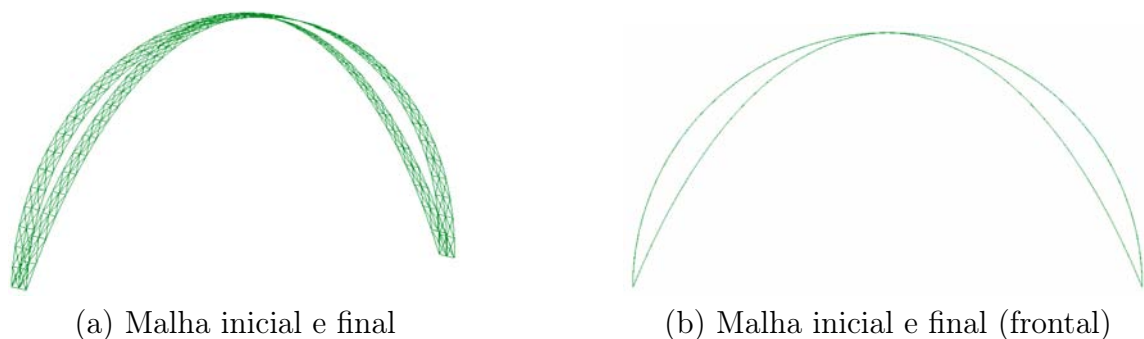


Figura 4.4: Geometria final 2.

Após doze iterações foi dada a convergência, com função objetivo $f_0(\sigma) = 0,2178 \cdot 10^9$ passando para $f_{12}(\sigma) = 0,5539 \cdot 10^4$. A geometria final obtida é apresentada na figura 4.4.

Este é outro problema que possui solução analítica conhecida, sabe-se bem que os arcos parabólico estão isentos de momento fletor se uma carga uniforme vertical for aplicada.

4.4.3 Exemplo 3

O terceiro exemplo trata da minimização da frequência fundamental do mesmo arco circular dos exemplos anteriores; entretanto a parametrização foi modificada com a inserção de duas linhas de nós (knots), em $u = 0,25$ e $u = 0,75$, para se obter uma parametrização mais maleável e assim obter geometrias mais complexas. A nova parametrização é apresentada na figura 4.5.

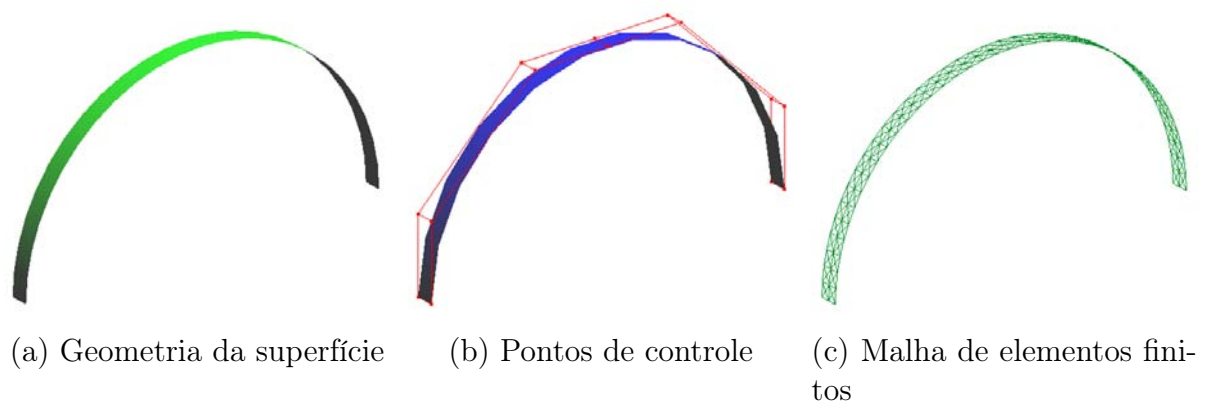


Figura 4.5: Exemplo 3.

Os pontos de controle da geometria são definidos como as variáveis do problema de otimização.

A função objetivo trata de minimizar a primeira frequência ω_0 da estrutura.

$$f(\omega) = \omega_0 \quad (4.2)$$

As restrições impostas ao problema são: a altura deve permanecer constante, restrições de simetria e a área em planta permanece constante.

Após uma iteração foi dada a convergência, com função objetivo $f_0(\omega) = 0,2935 \text{ rad/s}$ passando para $f_1(\omega) = 0,3261 \cdot 10^{-2} \text{ rad/s}$. A geometria final obtida é apresentada na figura 4.6.

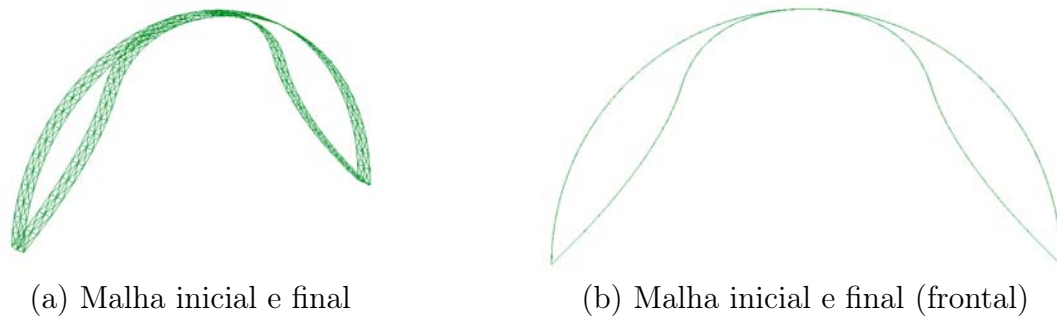


Figura 4.6: Geometria final 3.

4.4.4 Exemplo 4

O quarto exemplo é idêntico ao primeiro exemplo no que concerne à geometria, ao problema de otimização e ao problema de elementos finitos; entretanto para este exemplo foi considerada a não linearidade geométrica e a carga foi multiplicada por cem. Após doze iterações foi dada a convergência, com função objetivo $f_0(\sigma) = 0,6570 \cdot 10^{13}$ passando para $f_{12}(\sigma) = 0,6928 \cdot 10^3$. A geometria final obtida é apresentada na figura 4.7.

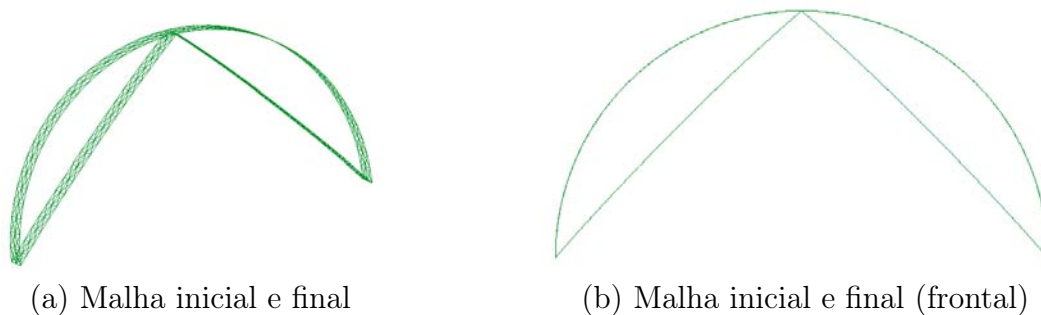


Figura 4.7: Geometria final 4.

O resultado é substancialmente diferente do primeiro exemplo já que a não linearidade geométrica é considerada. Ainda, observar que o resultado é uma geometria com curvatura Gaussiana positiva e não é um plano como foi obtido no primeiro exemplo.

4.4.5 Exemplo 5 – MIT –

Este exemplo foi apresentado por Ramm et al. (1993). O problema trata da otimização da casca do auditório Kresge construído em 1955 no MIT. A geometria da casca é um oitavo de uma esfera de raio $34,29 m$, apoiada nos três vértices, gerando um vão livre de $48,50 m$,



(a) Estrutura Kresge MIT (MIT)

(b) Construção do auditório (MIT)

Figura 4.8: Auditório Kresge MIT.

uma altura máxima de $14,50\text{ m}$, e uma altura na borda de $8,23\text{ m}$. A casca possui uma espessura de $8,9\text{ cm}$ no centro, que aumenta para $12,7\text{ cm}$ na direção central das vigas de bordo e aumenta para $49,5\text{ cm}$ na direção dos apoios. A viga de bordo possui uma largura de $25,4\text{ cm}$, uma altura de $50,8\text{ cm}$ no meio do vão, que aumenta para $91,4\text{ cm}$ na direção dos apoios. Apenas carga de peso próprio de $3,98\frac{\text{kN}}{\text{m}^2}$ correspondente ao volume inicial é considerada.

O material considerado nas análises possui um módulo de elasticidade, $E = 3 \cdot 10^4\frac{\text{N}}{\text{mm}^2}$, e um coeficiente de Poisson, $\nu = 0,2$.

Análise da geometria original

Neste exemplo é realizada a análise linear da casca original para comparar os resultados que Ramm et al. (1993) obtiveram. Cabe salientar que no trabalho de Ramm et al. (1993) foi usado o elemento de casca Mindlin isoparamétrico de oito nós. Este elemento possui deformação por corte, i.e., adequado para cascas de elevada espessura. Ainda, foram usados quarenta e dois elementos e as esquinas foram cortadas em uma distância de $1,20\text{ m}$ dos apoios; que segundo os autores, é necessário para evitar concentrações de tensões.

Por simplicidade, a espessura somente varia na direção dos apoios, com um valor máximo de $0,35\text{ m}$ e $0,089\text{ m}$ no centro. As vigas de bordo foram adotadas com uma largura de $0,25\text{ m}$ e sua altura varia de $0,51\text{ m}$ a $0,91\text{ m}$.

Foram realizadas duas análises iniciais, sem viga de bordo e com viga de bordo. Nas figuras 4.9a e 4.9b e estão apresentadas as malhas utilizadas, possuindo 990 e 3322 nós, para o caso sem viga e com viga, respectivamente.

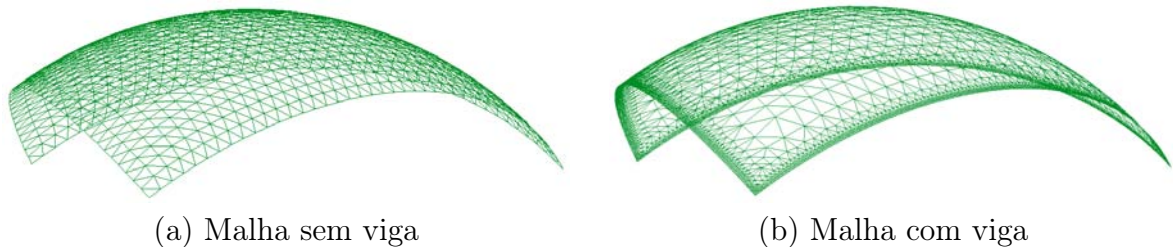


Figura 4.9: Malhas iniciais.

Na tabela 4.1 são apresentados os resultados e comparados com os obtidos por Ramm et al. (1993).

Tabela 4.1: Deslocamentos da geometria original.

	Sem Viga u_c	Sem Viga u_b	Com Viga u_c	Com Viga u_b
Presente trabalho	6,2 <i>cm</i>	26,9 <i>cm</i>	1,8 <i>cm</i>	5,5 <i>cm</i>
Ramm et al. (1993)	–	29,0 <i>cm</i>	1,7 <i>cm</i>	5,9 <i>cm</i>

onde u_c é o deslocamento do centro e u_b é o deslocamento no centro dos bordos.

Otimização

Na otimização foi usada a energia de deformação total como função objetivo, conforme a expressão (4.3), na tentativa de obter um estado puro de membrana, i.e., um estado de compressão total. Os pontos de controle da geometria e os parâmetros que definem a distribuição quadrática da espessura são definidos como as variáveis do problema de otimização. As restrições impostas ao problema são: a altura deve permanecer constante, restrições de simetria, a área em planta permanece constante, o volume permanece constante, a espessura mínima permitida é 8,9 *cm* e a espessura máxima permitida é 100 *cm*.

$$f_E = \frac{1}{2} \int \sigma : \epsilon \, dV \quad (4.3)$$

Na figura 4.10 são apresentados a geometria e seus respectivos pontos de controle.

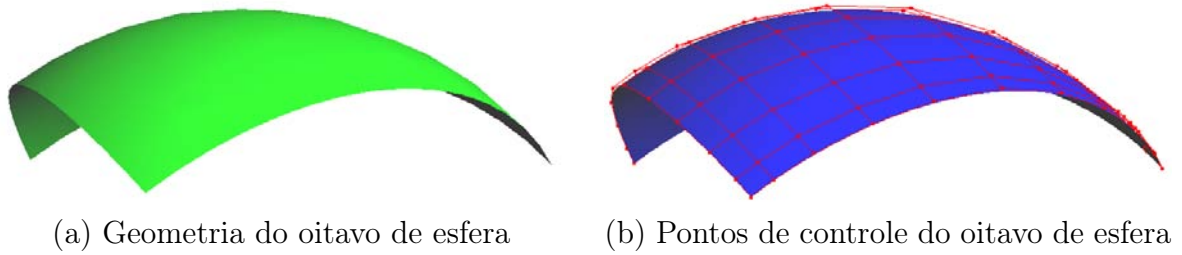


Figura 4.10: Geometria e pontos de controle – MIT –.

Após cento e quarenta e nove iterações foi dada a convergência para a otimização da casca sem viga de bordo, com função objetivo $f_0(\sigma) = 0,8511 \cdot 10^5$ passando para $f_{149}(\sigma) = 0,2621 \cdot 10^5$. Após cento e oito iterações foi dada a convergência para a otimização da casca com viga de bordo, com função objetivo $f_0(\sigma) = 0,4146 \cdot 10^5$ passando para $f_{108}(\sigma) = 0,1734 \cdot 10^5$.

Nas figuras 4.11 é apresentada a sequência de iterações e a geometria final do problema sem a viga.

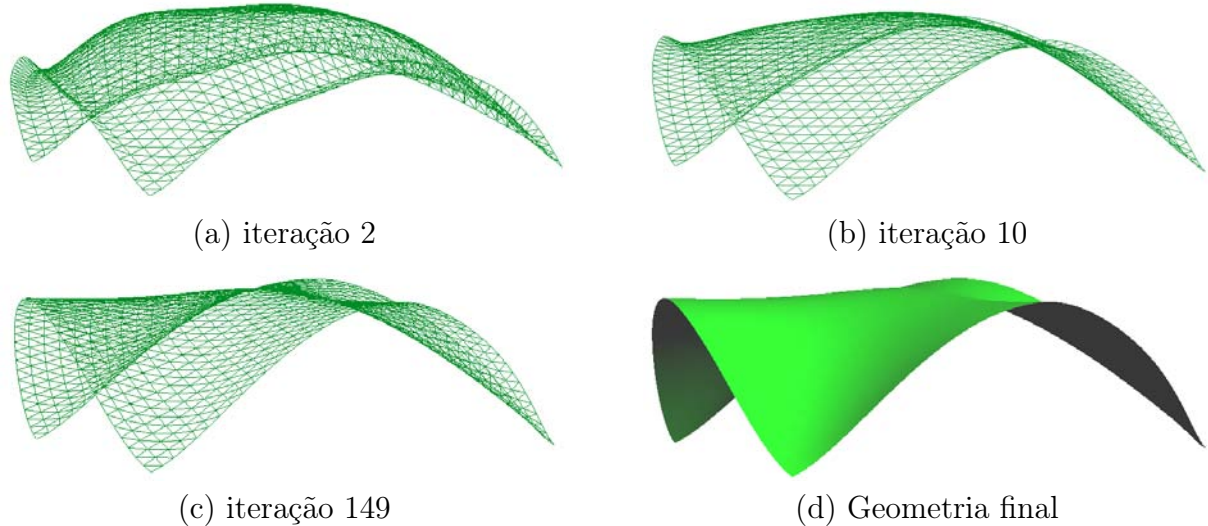


Figura 4.11: Sequência de iterações e geometria final (sem viga).

Observar que a geometria final é substancialmente diferente da inicial; ainda a curvatura Gaussiana nos bordos é negativa. Esta geometria, nos bordos, possui um aspecto mais parabólico em relação à geometria original.

Nas figuras 4.12 é apresentada a sequência de iterações e a geometria final do problema com a viga.

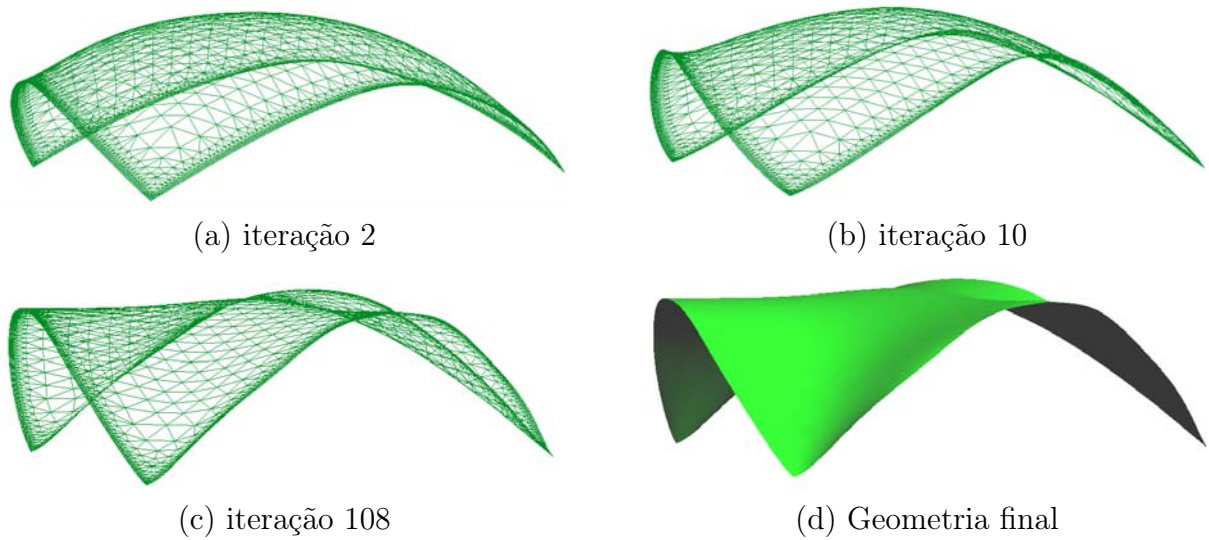


Figura 4.12: Sequência de iterações e geometria final (com viga).

Na tabela 4.2 são apresentados os resultados da geometria otimizada e comparados com os obtidos por Ramm et al. (1993) no que concerne aos deslocamentos no centro e na borda da casca.

Tabela 4.2: Deslocamentos da geometria otimizada.

	Sem Viga u_c	Sem Viga u_b	Com Viga u_c	Com Viga u_b
Presente trabalho	0,31 <i>cm</i>	0,15 <i>cm</i>	0,22 <i>cm</i>	0,09 <i>cm</i>
Ramm et al. (1993)	0,20 <i>cm</i>	0,30 <i>cm</i>	–	–

onde u_c é o deslocamento do centro e u_b é o deslocamento da borda.

Na tabela 4.3 são apresentados os resultados da geometria otimizada e comparados com os obtidos por Ramm et al. (1993) no que concerne às espessuras finais obtidas.

Tabela 4.3: Espessuras da geometria otimizada.

	Sem Viga e_c	Sem Viga e_a	Com Viga e_c	Com Viga e_a
Presente trabalho	8,9 <i>cm</i>	45,3 <i>cm</i>	8,9 <i>cm</i>	100,0 <i>cm</i>
Ramm et al. (1993)	8,9 <i>cm</i>	42,8 <i>cm</i>	8,9 <i>cm</i>	100,0 <i>cm</i>

onde e_c é a espessura do centro e e_a é a espessura no apoio.

Para o caso da casca com viga o fator de carga de flambagem na geometria original foi $\lambda = 15,9$ e Ramm et al. (1993) obtiveram $\lambda = 17,6$; para o caso da casca sem viga na geometria otimizada o fator foi $\lambda = 9,4$ e Ramm et al. (1993) obtiveram $\lambda = 11,3$.

Talvez a maior diferença entre as duas geometrias otimizadas é a curvatura nos bordos, para o caso do problema com viga de bordo, a curvatura no bordo é próxima de zero. Nas figuras 4.13 é apresentado o resultado obtido por Ramm et al. (1993).

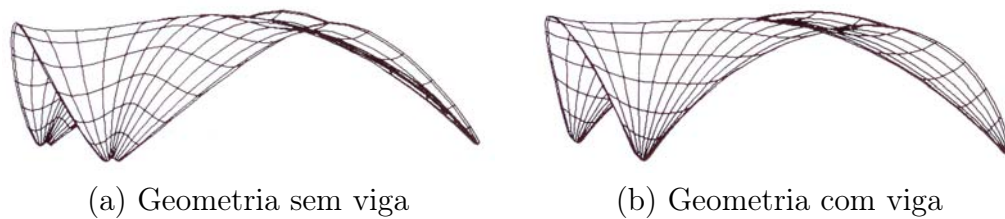


Figura 4.13: Geometria otimizada obtida por Ramm et al. (1993).

4.4.6 Últimos exemplos

Os últimos exemplos, apresentados a seguir, tratam da otimização de uma placa quadrada simplesmente apoiada nos quatro vértices. Esta possui um lado de $10,0\text{ m}$; onde o material utilizado possui um módulo de elasticidade, $E = 3 \cdot 10^4 \frac{N}{mm^2}$, e um coeficiente de Poisson, $\nu = 0,2$ e um peso específico, $\gamma = 25 \frac{kN}{m^3}$. A parametrização, NURBS, para a otimização é apresentada na figura 4.14, e na figura 4.14c é apresentada a malha inicial do problema.

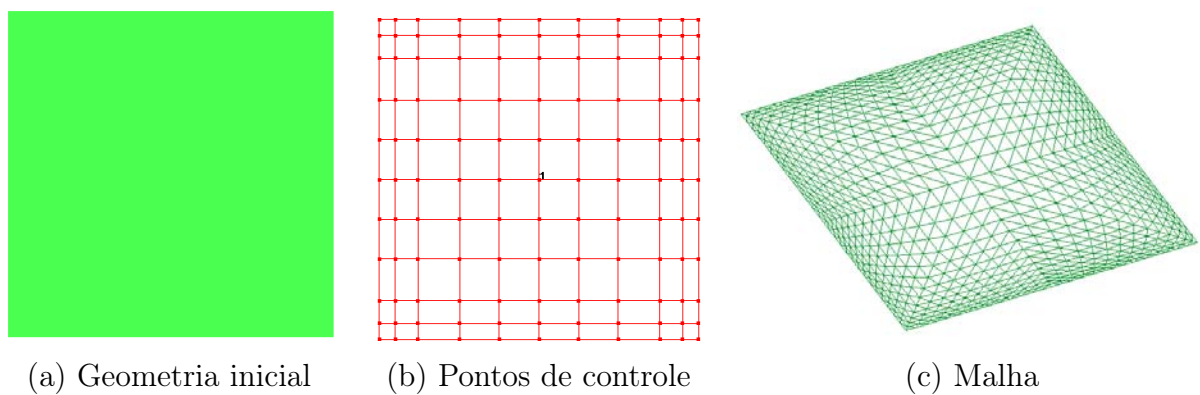


Figura 4.14: Geometria, pontos de controle e malha – Placa –.

Embora os exemplo apresentados a seguir sejam irreais mecanicamente, o objetivo fundamental é mostrar a eficiência dos algoritmos em situações críticas. As análises podem

conduzir a um resultado diferente do ótimo, devido ao fato de que não se considerou a não linearidade geométrica, i.e., quanto maior a diferença entre a análise linear e a não linear geométrica, maior será a diferença entre o ponto de ótimo da otimização linear e a otimização não linear geométrica.

Exemplo 6

Neste exemplo, como função objetivo buscou-se minimizar o deslocamento no centro da placa. As restrições impostas são: o deslocamento no meio do vão das bordas deve ser inferior a $0,25\text{ cm}$, restrições de simetria, a área em planta permanece constante, o volume permanece constante, a espessura mínima é $5,0\text{ cm}$ e a máxima é $45,0\text{ cm}$. A espessura varia quadraticamente do centro da placa na direção dos apoios. O carregamento é constituído pelo peso próprio mais uma carga uniforme de $2,5\frac{\text{kN}}{\text{m}^2}$. Na geometria original foi considerada a espessura de $7,0\text{ cm}$ no centro da placa e de $20,0\text{ cm}$ nos apoios.

A análise inicial da placa apresenta um deslocamento de $58,0\text{ cm}$ e $3,3\text{ cm}$, no centro da placa e no meio do vão das bordas, respectivamente. Esta é apresentada na figura 4.15.

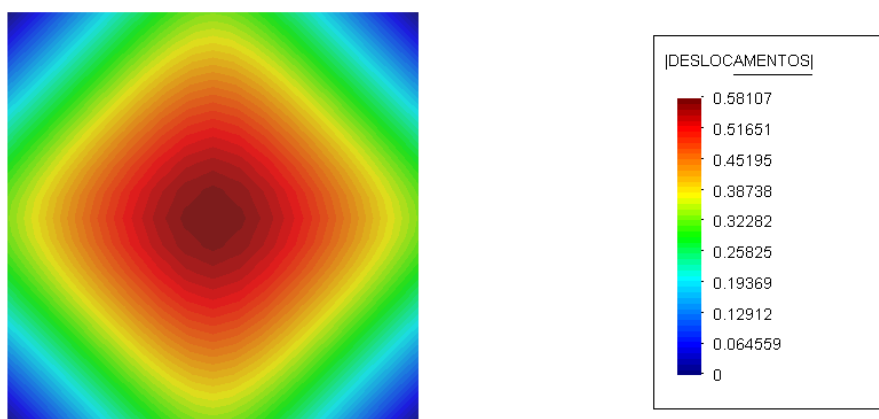


Figura 4.15: Deslocamentos da geometria inicial (6).

Após trinta e oito iterações foi dada a convergência para a otimização da casca, com função objetivo $f_0(\mathbf{u}) = 0,58$ passando para $f_{38}(\mathbf{u}) = 0,0018$. O deslocamento no centro da casca é $0,18\text{ cm}$ e no meio do vão das bordas $0,16\text{ cm}$. A espessura da casca resultou em $5,0\text{ cm}$ e $28,0\text{ cm}$, no centro da casca e nos apoios, respectivamente.

Nas figuras 4.16 é apresentada a sequência de iterações, a geometria e malha final do problema de otimização; a análise da geometria final nas figuras 4.17 e 4.18.

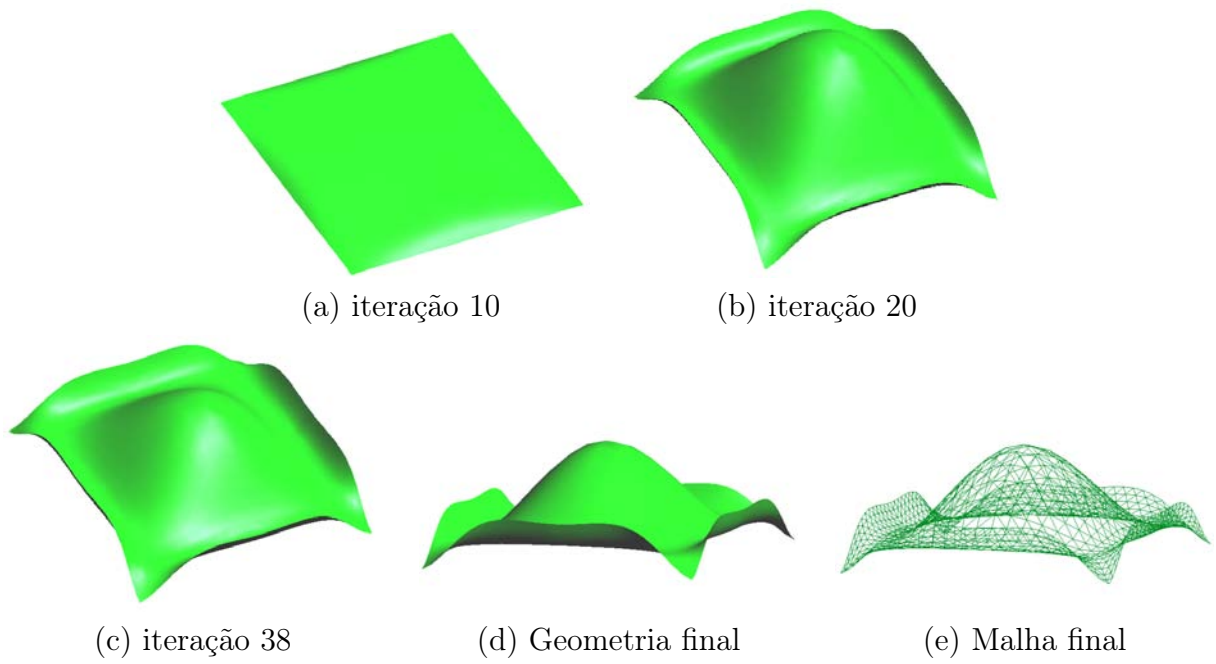


Figura 4.16: Sequência de iterações e geometria final (6).

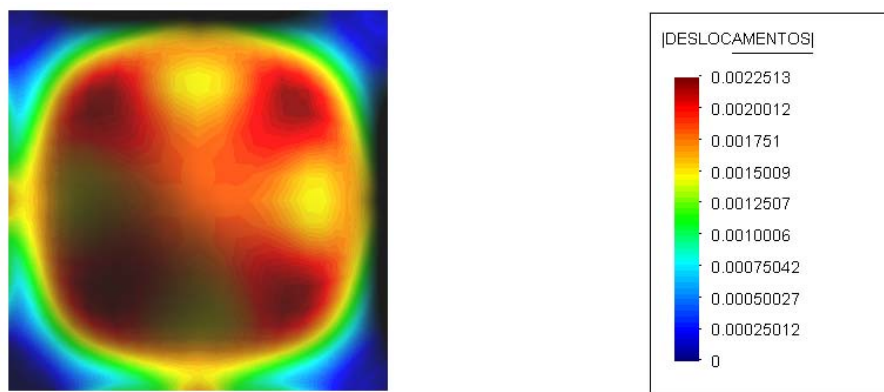


Figura 4.17: Deslocamentos da geometria final 1 (6).

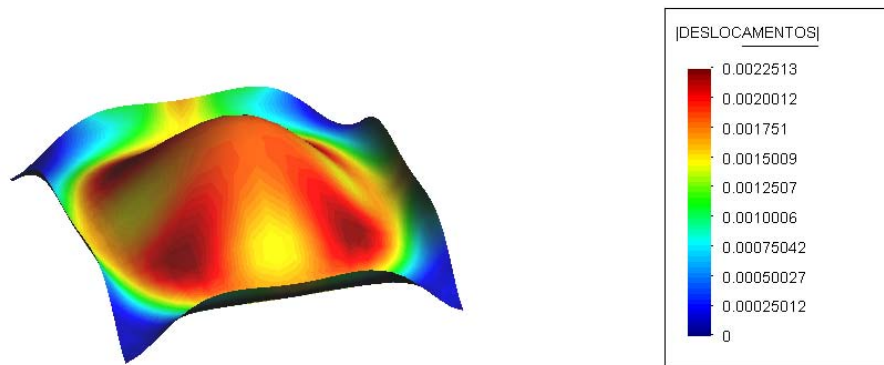


Figura 4.18: Deslocamentos da geometria final 2 (6).

Exemplo 7

Neste exemplo, como função objetivo buscou-se minimizar energia de deformação, descrita em (4.3). As restrições impostas ao problema são: o deslocamento no meio do vão das bordas e no centro da placa deve ser inferior à $0,5 \text{ cm}$, restrições de simetria, o volume permanece constante, a espessura mínima é $5,0 \text{ cm}$ e a máxima é $10,0 \text{ cm}$, com a espessura constante. Ainda, os pontos de controle só podem diminuir seu valor em z , i.e., deslocar-se para baixo. O carregamento é constituído por uma carga uniforme de $5,5 \frac{\text{kN}}{\text{m}^2}$ e a espessura foi de $5,0 \text{ cm}$. O módulo de elasticidade foi aumentado em dez vezes.

A análise inicial da placa apresenta um deslocamento de $43,7 \text{ cm}$ e $29,5 \text{ cm}$, no centro da placa e no meio do vão das bordas, respectivamente. Esta é apresentada na figura 4.19.

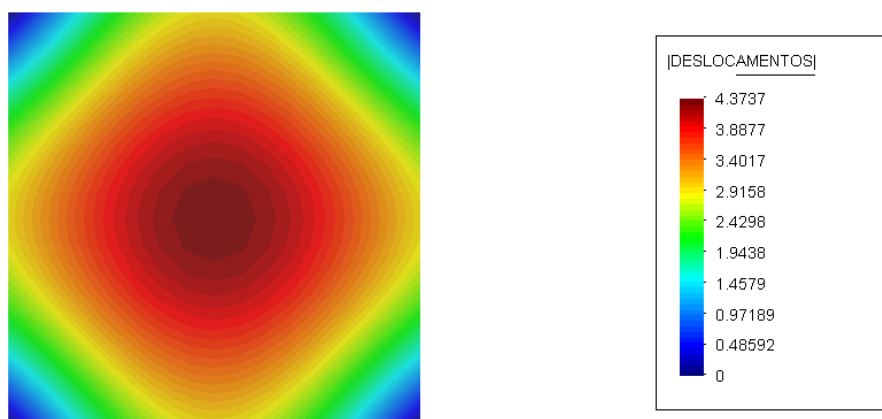


Figura 4.19: Deslocamentos da geometria inicial (7).

Após cento e vinte e nove iterações foi dada a convergência para a otimização da casca, com função objetivo $f_0(\sigma) = 0$ passando para $f_{129}(\sigma) = 2,40$. O deslocamento no centro da casca é $0,016 \text{ cm}$ e no meio do vão das bordas $0,015 \text{ cm}$. A espessura da casca resultou em $10,0 \text{ cm}$ para toda a casca. Obviamente a função objetivo aumento porque a placa está isenta de tensões de membrana, por esta razão as restrições de deslocamento possuem um rol essencial neste problema. Estas restrições “tiram” a geometria desta posição inicial, e pelo resultado final pode-se verificar que estas nem sequer participam mais do resultado final, já que estas possuem um multiplicador de Lagrange igual à zero, no que concerne à condição de complementariedade (3.28e).

Nas figuras 4.20 é apresentada a sequência de iterações, a geometria e malha final do problema de otimização; e a análise da geometria final nas figuras 4.21 e 4.22.

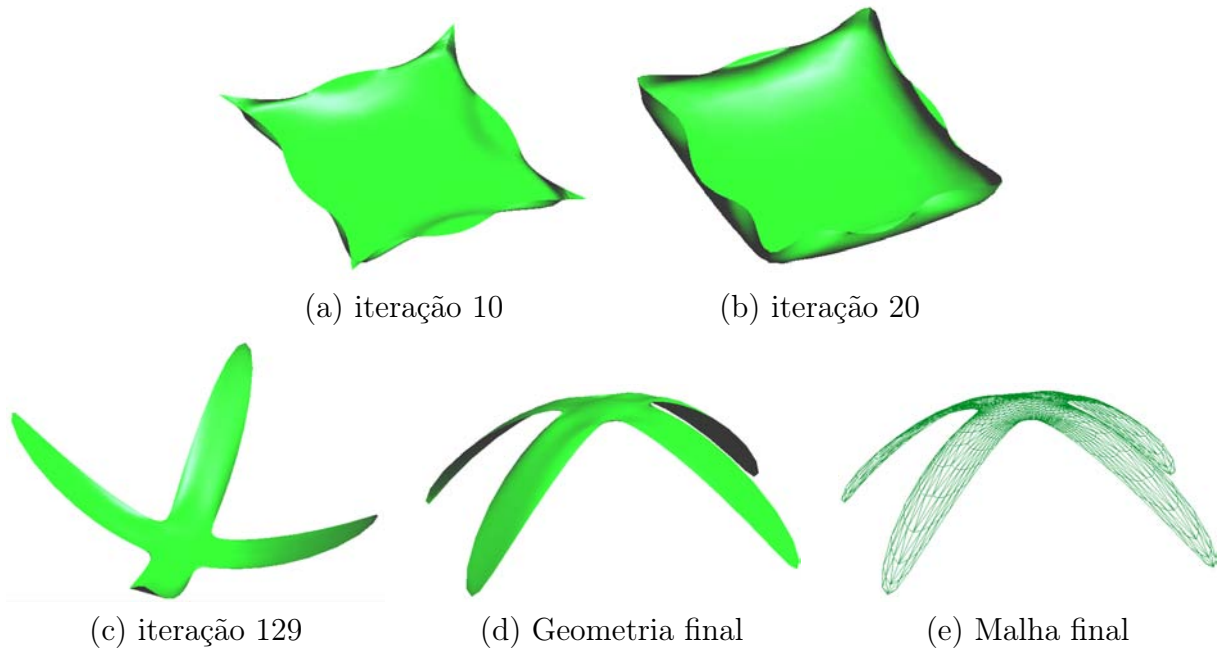


Figura 4.20: Sequência de iterações e geometria final (7).

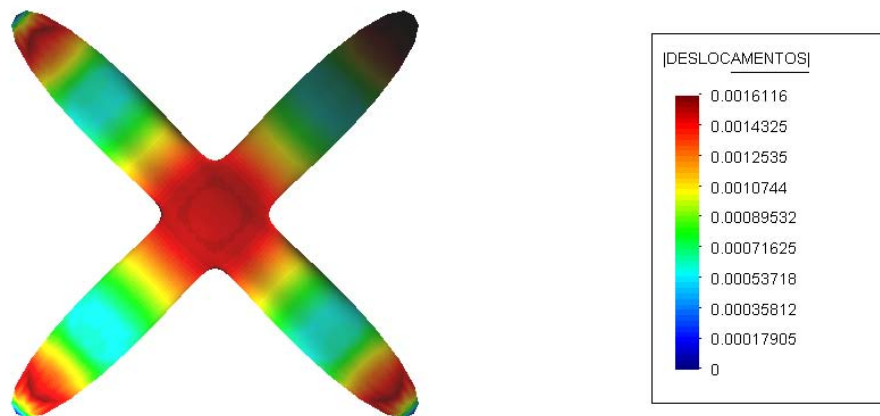


Figura 4.21: Deslocamentos da geometria final 1 (7).

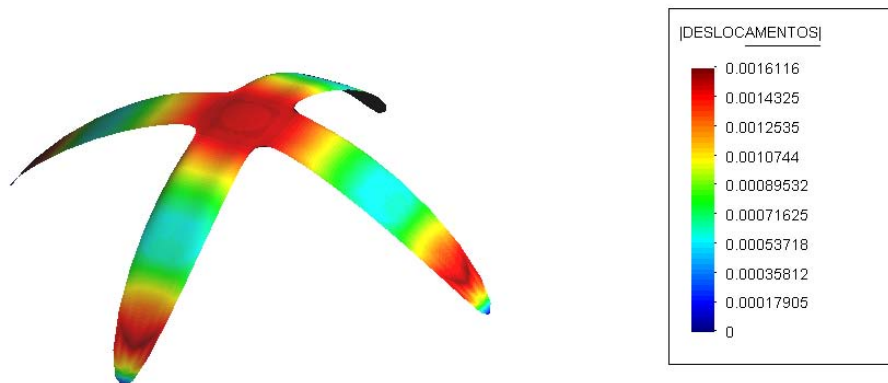


Figura 4.22: Deslocamentos da geometria final 2 (7).

Como os pontos de controle apenas podiam ser deslocados na direção de $-z$, o único estado de tensões que se pode obter é o de membrana à tração, no que concerne ao concreto armado, tal estado é indesejado. Entretanto, na análise final, a casca foi invertida e assim obteve-se um estado de membrana à compressão.

Exemplo 8

Este exemplo é basicamente uma modificação do anterior. As modificações em relação ao anterior são: a espessura possui distribuição quadrática, com valores iniciais de $7,0\text{ cm}$ e $20,0\text{ cm}$ no centro da casca e nos apoios, respectivamente. Em relação às restrições, foi adicionada uma restrição para que a área em planta permaneça constante e a espessura máxima permitida é $45,0\text{ cm}$. No restante do problema nada foi alterado.

A análise inicial da placa apresenta um deslocamento de $62,9\text{ cm}$ e $35,0\text{ cm}$, no centro da placa e no meio do vão das bordas, respectivamente. Esta é apresentada na figura 4.23.

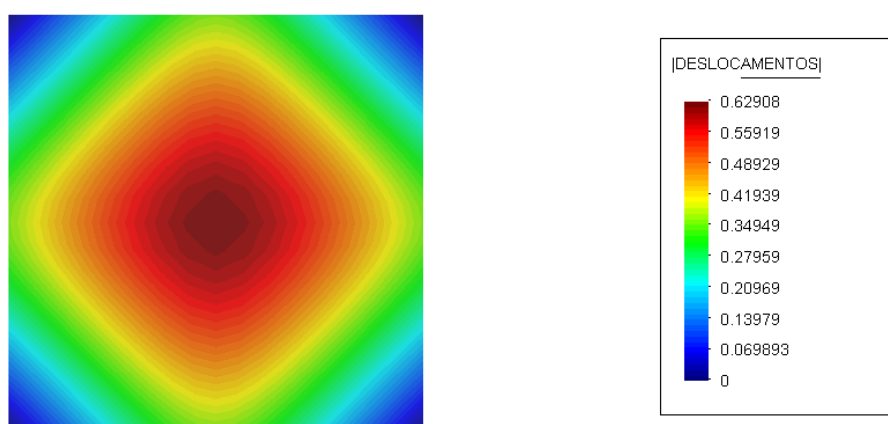


Figura 4.23: Deslocamentos da geometria inicial (8).

Após cento e sessenta e seis iterações foi dada a convergência para a otimização da casca, com função objetivo $f_0(\sigma) = 0$ passando para $f_{166}(\sigma) = 46,90$. O deslocamento no centro da casca é $0,014\text{ cm}$ e no meio do vão das bordas $0,017\text{ cm}$. A espessura da casca resultou em $5,0\text{ cm}$ e $45,0\text{ cm}$, no centro da casca e nos apoios, respectivamente.

Neste caso, também obteve-se um estado de membrana à compressão ao inverter a casca.

Nas figuras 4.24 é apresentada a sequência de iterações, a geometria e malha final do problema de otimização; e a análise da geometria final nas figuras 4.25 e 4.26.

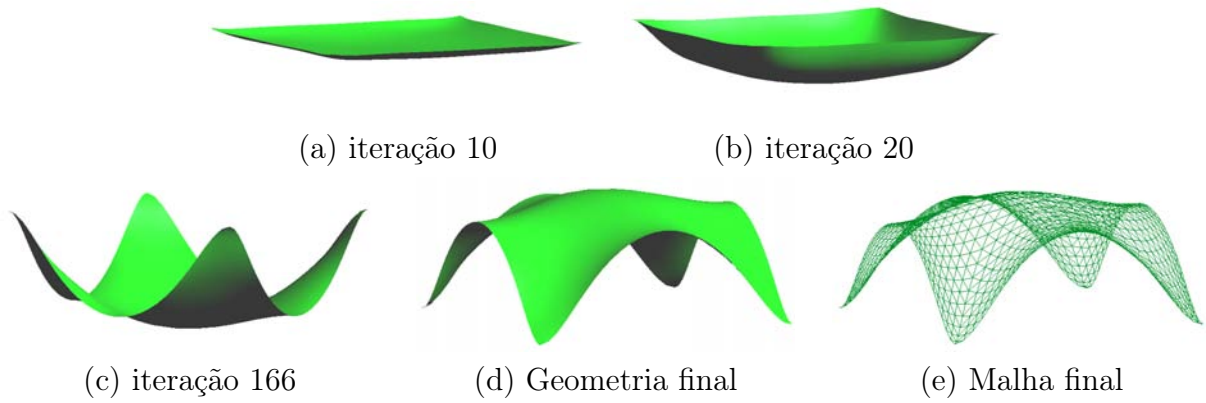


Figura 4.24: Sequência de iterações e geometria final (8).

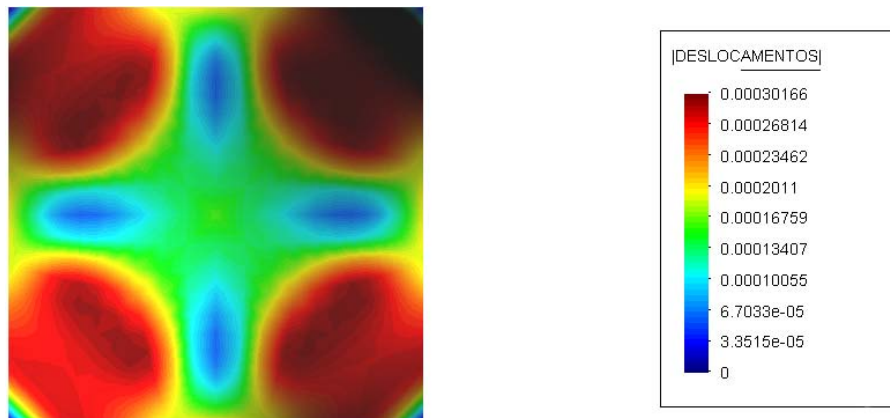


Figura 4.25: Deslocamentos da geometria final 1 (8).

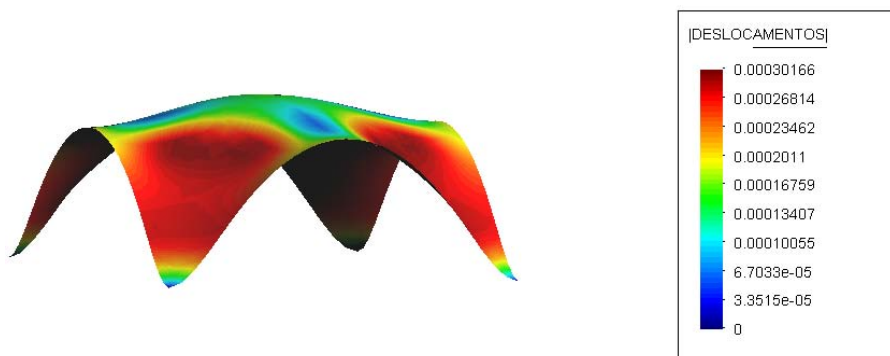


Figura 4.26: Deslocamentos da geometria final 2 (8).

Exemplo 9

Este exemplo apresenta o mesmo problema de otimização que o anterior, a mesma geometria e parametrização. Entretanto, foi acrescentado ao carregamento a carga de peso próprio; tornando-o mais real.

A análise inicial da placa apresenta um deslocamento de $92,4 \text{ cm}$ e $53,3 \text{ cm}$, no centro da placa e no meio do vão das bordas, respectivamente. Esta é apresentada na figura 4.27.

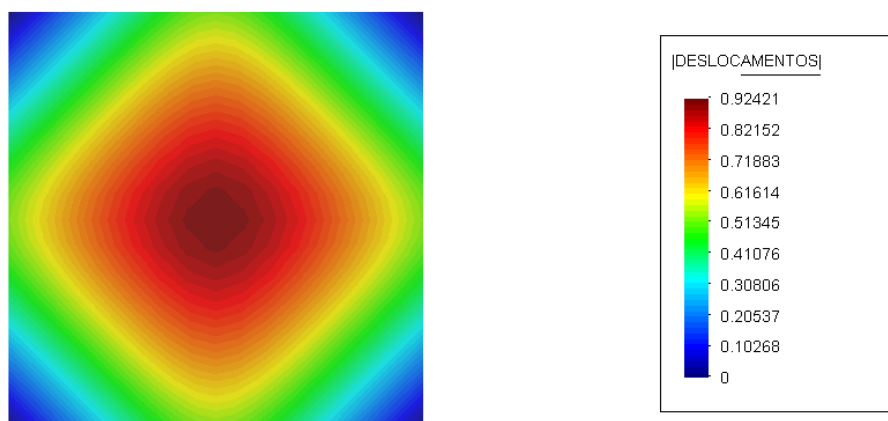


Figura 4.27: Deslocamentos da geometria inicial (9).

Após duzentas e oitenta e duas iterações foi dada a convergência para a otimização da casca, com função objetivo $f_0(\sigma) = 0$ passando para $f_{282}(\sigma) = 2178,3$. O deslocamento no centro da casca é $0,041 \text{ cm}$ e no meio do vão das bordas $0,041 \text{ cm}$. A espessura da casca resultou em $10,0 \text{ cm}$ e $30,2 \text{ cm}$, no centro da casca e nos apoios, respectivamente. Este caso conclui com uma série de exemplos que apresentaram geometrias otimizadas com alto desempenho e ainda, no que concerne à estética, obteve-se um resultado agradável (figura 4.28).

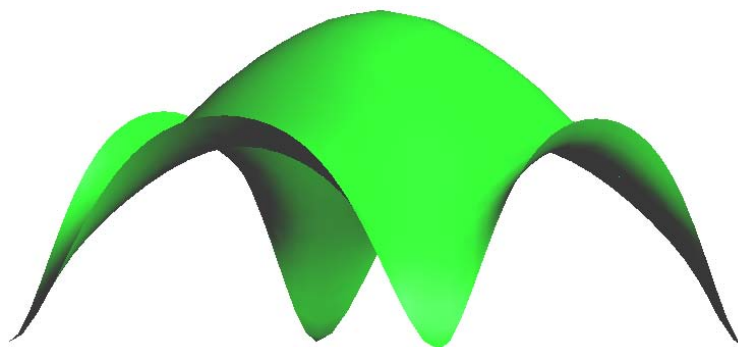


Figura 4.28: Geometria final, agradável resultado estético (9).

Nas figuras 4.29 é apresentada a sequência de iterações, a geometria e malha final do problema de otimização; e a análise da geometria final nas figuras 4.30 e 4.31.

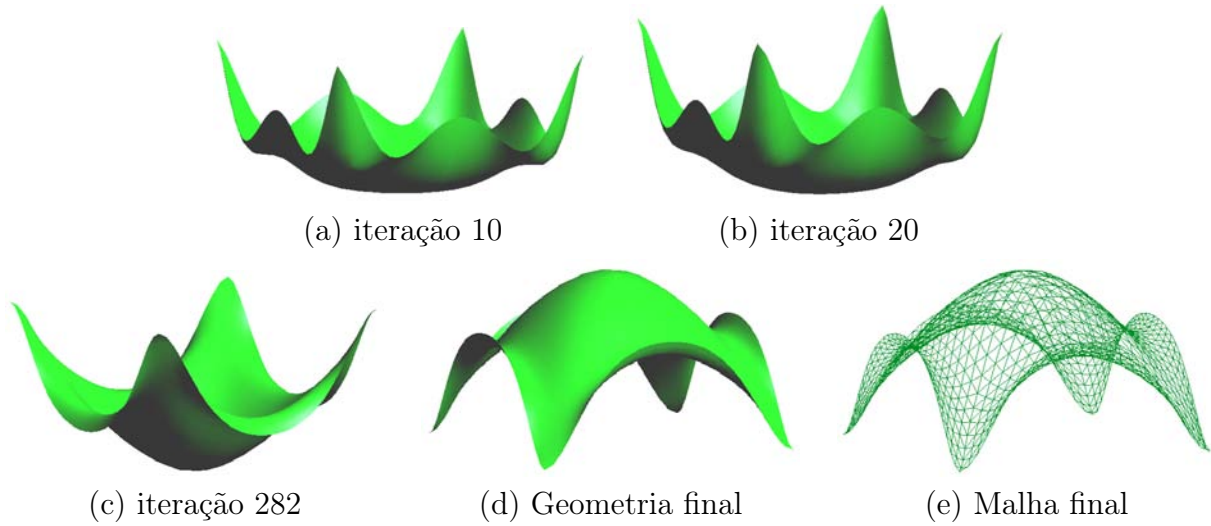


Figura 4.29: Sequência de iterações e geometria final (9).

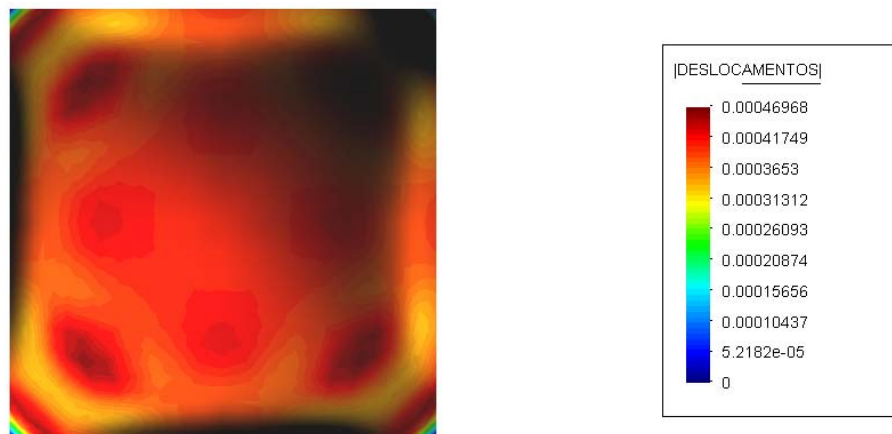


Figura 4.30: Deslocamentos da geometria final 1 (9).

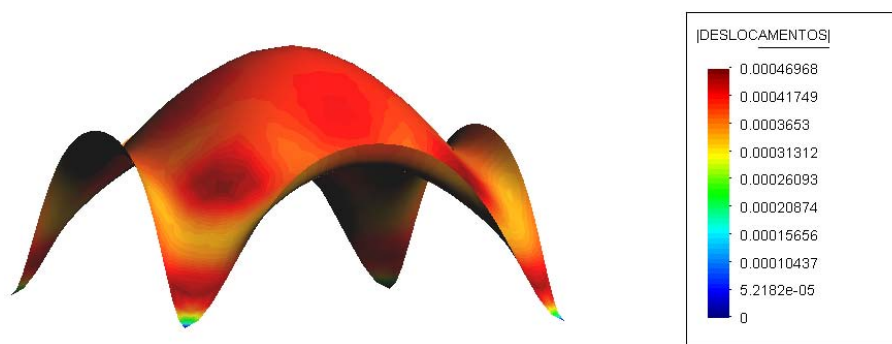


Figura 4.31: Deslocamentos da geometria final 2 (9).

Capítulo 5

Conclusões Finais

Neste trabalho foi desenvolvido um código computacional para a modificação de forma, deformação livre de forma, baseado em NURBS. Este código juntamente com o código de F.E.M., elementos finitos de casca, foram acoplados ao código de SQP, programação quadrática sequencial. Deste modo obteve-se um código de otimização de forma de cascas, onde cada uma destas etapas foram discutidas anteriormente e alguns aspectos merecem ser destacados.

Aspectos da parametrização NURBS e da deformação livre de forma:

- Com base nos exemplos analisados no capítulo 2, onde foram visto exemplos de modificação de forma, pode-se dizer que a parametrização NURBS permite modelar qualquer geometria com qualquer grau de complexidade; ainda, devido à alta eficiência dos algoritmos NURBS e à baixa memória de armazenamento utilizada na modificação de forma levaram a resultados “instantâneos”, i.e., nenhum dos exemplos apresentados marcou tempo diferente de zero, mesmo os que possuíam uma densa malha;
 - Talvez o aspecto mais importante da deformação livre de forma é que em nenhum momento perde-se a parametrização NURBS ao se deformar a geometria. No que concerne à construção da geometria deformada (ou otimizada) este fato permite ter uma descrição completa da geometria no projeto de engenharia;
 - O código desenvolvido, sem nenhuma alteração, pode ser usado para outros fins, como otimização de formas aerodinâmicas;
-

- O código desenvolvido independe do gerador de malha e do CAGD.

Aspectos da otimização de forma:

- Os algoritmos de otimização talvez sejam os menos robustos na engenharia; quase nunca se sabe se o problema será resolvido. Em nenhum dos casos apresentados de otimização de forma obteve-se a solução na primeira tentativa. O fator de escala foi o principal culpado, já que é extremamente difícil prever o comportamento do gradiente e quase impossível prever o comportamento quando tem-se variáveis de diferentes espécies, i.e., espessura, pontos de controle. No primeiro exemplo apresentado, com as variáveis escaladas obteve-se o resultado após seis iterações. Entretanto, sem escalar as variáveis após trinta e sete iterações o programa era interrompido por não obter um decréscimo de f na busca linear;
- Utilizar um método que desloque a malha para a posição deformada, e não ter que refazer a malha após a modificação da forma, diminui muito o tempo de processo, já que gerar a malha é muito lento comparado com a análise estrutural;
- Uma das grandes vantagens do SQP pode ser visto nos últimos quatro exemplos, onde em todos os casos partiu-se de um ponto inviável;
- Mesmo com tantas dificuldades para otimizar uma casca, os resultados foram excelentes, alto desempenho e esteticamente agradáveis;
- Em regiões onde há ocorrência de neve, cascas que aparentemente são ótimas, na verdade podem ser grandes acumuladores de neve, algo bastante indesejável, como é o caso da casca do MIT;
- Na engenharia civil o cuidado com a forma das cascas deve ser criteriosamente analisada, a infra-estrutura para construir uma casca de concreto armado é enorme, i.e., até a construção finalizar, esta é totalmente instável, logo necessita-se de um projeto de escoramento cauteloso. Ainda, a área de formas de madeira necessária é demasiada;
- Claramente percebe-se que o senso comum, na hora de definir a geometria de uma casca, não é confiável; caso contrário o processo de otimização seria dispensável.

Sugestões: Como sugestões são deixadas as seguintes:

- Estudar detalhadamente e demonstrar a não convexidade da função de energia de deformação em relação aos pontos de controle;
- Implementar a continuidade entre superfícies, *surface patch*, para obter uma continuidade prescrita entre duas ou mais superfícies na modificação de forma;
- Investigar, de forma detalhada, a influência da formulação dos elementos sobre os erros dos gradientes;
- Implementar a diferenciação automática para o computo do gradiente e Hessiano (*automatic differentiation*);
- Investigar, de forma detalhada, como utilizar um fator de escala dinâmico, de forma que a escala seja modificada em função da forma do gradiente da iteração anterior, i.e., que possa ser modificado a cada iteração. Esta proposta impede o uso de aproximações do Hessiano do tipo BFGS, já que estes baseiam-se em iterações anteriores;
- Acoplar o código de fluidos ao código de otimização de formas de cascas para a otimização aerodinâmica e aero-elástica;
- Acoplar o problema de otimização topológica;
- Paralelizar o computo dos gradientes.

Referências Bibliográficas

- Almeida, F. S. (2006). Otimização de estruturas de materiais compósitos laminados utilizando algoritmos genéticos. Master's thesis, Programa de Pós-Graduação em Engenharia Civil, Universidade Federal do Rio Grande do Sul.
- Bathe, K.-J. e Ho, L.-W. (1981). A simple and effective element for analysis of general shell structures. *Computers & Structures*, Vol. 13, No. 5–6, pp. 673–681.
- Batoz, J.-L., Bathe, K.-J., e Ho, L.-W. (1980). A study of three-node triangular plate bending elements. *International Journal for Numerical Methods in Engineering*, Vol. 15, No. 1, pp. 1771–1812.
- Bernstein, S. N. (1912). Démonstration du théorème de weierstrass fondée sûr le calcul dès probabilités. *Commun. Soc. Math*, Vol. 12, No. 2, pp. 1–2.
- Butterfield, K. R. (1976). The computation of all the derivatives of a b-spline basis. *Inst. Math. Applic.*, Vol. 17, No. 1, pp. 15–25.
- Bézier, P. E. (1972). *Numerical Control: Mathematics and applications*. John Wiley, New York.
- Cox, M. G. (1972). The numerical evaluation of b-spline. *Inst. Math. Applic.*, Vol. 10, No. 1, pp. 134–149.
- Daniel, I. M. e Ishai, O. (1994). *Engineering Mechanics of Composite Materials*. Oxford Press.
- deBoor, C. (1972). On calculating with b-spline. *Approx. theory*, Vol. 6, No. 1, pp. 50–72.
- Hiriart-Urruty, J. B. e Lemaréchal, C. (1993). *Convex Analysis and Minimization Algorithms*. Springer-Verlag, Berlin, New York, 6th edition.
- Luo, Z. e Hutton, S. G. (2002). Formulation of a three-node traveling triangular plate element subjected to gyroscopic and in-plane forces. *Computers & Structures*, Vol. 80, No. 26, pp. 1935–1944.
- López, S. D. C. (2009). Otimização de placas e cascas de materiais compósitos, utilizando algoritmos genéticos, redes neurais e elementos finitos. Master's thesis, Programa de Pós-Graduação em Engenharia Civil, Universidade Federal do Rio Grande do Sul.
- Nocedal, J. e Wright, S. J. (1999). *Numerical Optimization*. Springer, 2th edition.
-

-
- Patterson, R. R. (1985). Projective transformations of the parameter of a bernstein-bézier curve. *ACM TOG*, Vol. 4, No. 4, pp. 276–290.
- Piegl, L. A. (1989a). Modifying the shape of rational b-splines. part 1: curves. *Computer-Aided Design*, Vol. 21, No. 8, pp. 509–518.
- Piegl, L. A. (1989b). Modifying the shape of rational b-splines. part 1: surfaces. *Computer-Aided Design*, Vol. 21, No. 9, pp. 538–546.
- Piegl, L. A. e Tiller, W. (1989). Key developments in computer-aided geometry and design. *CAD*, Vol. 21, No. 5, pp. 262–273.
- Piegl, L. A. e Tiller, W. (1997). *The Nurbs Book*. Springer, 2th edition.
- Ramm, E., Bletzinger, K. U., e Reitering, R. (1993). Shape optimization of shell structures. *IASS Bulletin of the international association for shelland spatial structures*, Vol. 34, No. 112, pp. 103–121.
- Riesenfeld, R. F. (1973). *Applications of B-spline approximation to geometric problems of Computer-Aided Design*. PhD thesis, Syracuse.
- Riesenfeld, R. F. (1981). Homogeneous coordinates and projective planes in computer graphics. *IEEE Comput. Graph. and Appl.*, Vol. 1, No. 1, pp. 50–55.
- Roberts, D. F. (1965). Homogeneous matrix representation and manipulation of n -dimensional constructs. Technical report ms-1405, Lincoln Laboratory, MIT, Cambridge, MA.
- Sederberg, T. e Parry, S. (1986). Free-form deformation of solid geometric objects. *Comput. Graph.*, Vol. 20, No. 4, pp. 151–160.
- Sederberg, T., Zheng, J., Bakenov, A., e Nasri, A. (2003). T-splines and t-nurccs. *ACM Transactions on Graphics*, Vol. 22, No. 3, pp. 477–484.
- Terzopoulos, D. e Qin, H. (1994). Dynamic nurbs with geometric constraints for interactive sculpting. *ACM Transactions on Graphics*, Vol. 13, No. 2, pp. 103–136.
- To, C. W. S. e Wang, B. (1998). Hybrid strain-based three-node flat triangular laminated composite shell elements for vibration analysis. *Journal of Sound and Vibration*, Vol. 211, No. 2, pp. 277–291.

Apêndice A

Análise de cascas por elementos finitos

Neste apêndice é apresentada a formulação do elemento finito de casca. O código do elemento foi desenvolvido por Almeida (2006), e este apêndice é baseado no trabalho López (2009).

A.1 Formulação do Elemento Finito

Pela superposição do elemento triangular de estado plano de tensões, *Constant Stress Triangular element* (CST) e do elemento triangular para a análise de flexão de placas finas *Discret Kirchhoff Triangular element* (DKT), obtém-se o elemento de casca (Batoz et al., 1980). As considerações adotadas para a análise das estruturas de materiais compósitos laminados são as seguintes (Daniel e Ishai, 1994)

- Cada lâmina é quase-homogênea e ortotrópica;
 - O laminado é delgado se comparado às suas dimensões laterais e suas lâminas estão sob estado plano de tensões;
 - Os deslocamentos são pequenos se comparados à espessura do laminado, consideração essa válida para teoria com linearidade geométrica;
 - Os deslocamentos são contínuos no laminado;
 - Deslocamentos no plano variam linearmente ao longo da espessura do laminado;
-

- Deformações transversais por cisalhamento são ignoradas. Essa consideração e a anterior implicam em que linhas retas normais ao plano médio continuam retas e normais a este após a deformação;
- As relações deformação-deslocamento e tensão-deformação são lineares;
- Distâncias normais ao plano médio permanecem constantes, i.e., a deformação normal na direção perpendicular ao plano médio ϵ_z é ignorada.

O elemento tem em total seis graus de liberdade por nó; totalizando dezoito graus de liberdade, os quais são indicados na figura A.1.

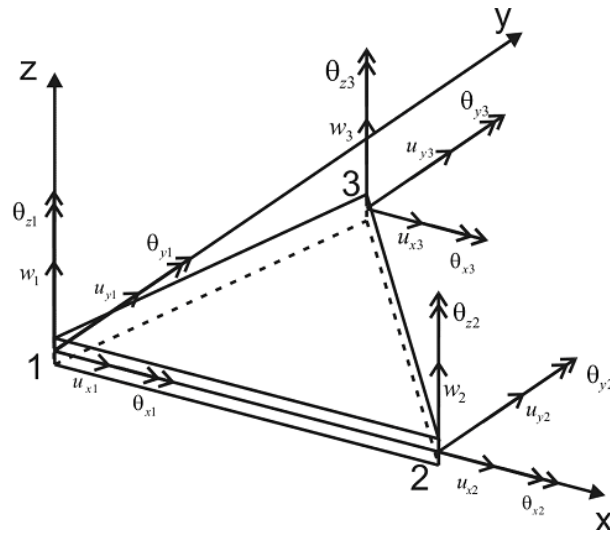


Figura A.1: Graus de liberdade do elemento triangular plano para placa e casca (Almeida, 2006)

A.1.1 Matriz de Rigidez

A matriz de rigidez linear total é obtida pela sobreposição das matrizes de rigidez de membrana $[K_m]$, de flexão $[K_b]$, de acoplamento membrana-flexão $[K_{mb}]$ e um termo, o qual é adicionado na diagonal da matriz correspondentes ao grau de liberdade de rotação perpendicular ao plano, que é 10^{-4} vezes menor que o maior termo encontrado na matriz $[K_b]$ (Bathe e Ho, 1981). As matrizes descritas são apresentadas a seguir.

$$[K_m] = [B_m]^T [D_m] [B_m] \quad (\text{A.1})$$

$$[K_b] = 2A \int_0^1 \int_0^{1-\eta} [B_b]^T [D_b] [B_b] d\xi d\eta \quad (\text{A.2})$$

$$[K_{mb}] = \int_A [B_m]^T [D_{mb}] [B_b] dA \quad (\text{A.3})$$

As matrizes constitutivas dos laminados vêm dadas por

$$[D_m] = \sum_{k=1}^{nlam} [\bar{Q}]_{(k)} \int_{z_{k-1}}^{z_k} dz, \text{ sendo } D_{m \ ij} = \sum_{k=1}^{nlam} \bar{Q}_{(k)ij} (z_k - z_{k-1}) \quad (\text{A.4})$$

$$[D_{mb}] = \sum_{k=1}^{nlam} [\bar{Q}]_{(k)} \int_{z_{k-1}}^{z_k} z dz, \text{ sendo } D_{mb \ ij} = \frac{1}{2} \sum_{k=1}^{nlam} \bar{Q}_{(k)ij} (z_k^2 - z_{k-1}^2) \quad (\text{A.5})$$

$$[D_b] = \sum_{k=1}^{nlam} [\bar{Q}]_{(k)} \int_{z_{k-1}}^{z_k} z^2 dz, \text{ sendo } D_{b \ ij} = \frac{1}{3} \sum_{k=1}^{nlam} \bar{Q}_{(k)ij} (z_k^3 - z_{k-1}^3) \quad (\text{A.6})$$

sendo

$$[\bar{Q}] = [T]^{-1} [Q] [T_e] \quad (\text{A.7})$$

As matrizes da equação (A.7) vêm dadas por

$$[Q] = \begin{bmatrix} \frac{E_1}{1-\nu_{12}\nu_{21}} & \frac{\nu_{12}E_2}{1-\nu_{12}\nu_{21}} & 0 \\ \frac{\nu_{12}E_2}{1-\nu_{12}\nu_{21}} & \frac{E_2}{1-\nu_{12}\nu_{21}} & 0 \\ 0 & 0 & G_{12} \end{bmatrix} \quad (\text{A.8})$$

$$[T] = \begin{bmatrix} m^2 & n^2 & 2nm \\ n^2 & m^2 & -2nm \\ -nm & nm & m^2 - n^2 \end{bmatrix} \quad (\text{A.9})$$

$$[T_e] = \begin{bmatrix} m^2 & n^2 & nm \\ n^2 & m^2 & -nm \\ -2nm & 2nm & m^2 - n^2 \end{bmatrix}, \text{ com } m = \cos \theta \text{ e } n = \sin \theta \quad (\text{A.10})$$

onde θ corresponde ao ângulo esquematizado na figura A.2. Na equação (A.8) E_1 e E_2 são os módulos de elasticidade nas direções das fibras e transversal a elas, G_{12} é o módulo de cisalhamento, ν_{12} e ν_{21} os coeficientes de Poisson. A matriz $[B_{m_i}]$, para $i = 1, 2, 3$ que relaciona componentes de deformações específicas e componentes de deslocamentos correspondente a graus de liberdade de membrana, forma parte de $[B_m]$ na equação (A.1) e (A.3), vem dada por

$$[B_{m_i}] = \frac{1}{4A} \begin{bmatrix} 2b_i & 0 \\ 0 & 2c_i \\ 2c_i & 2b_i \end{bmatrix}, (i = 1, 2, 3) \quad (\text{A.11})$$

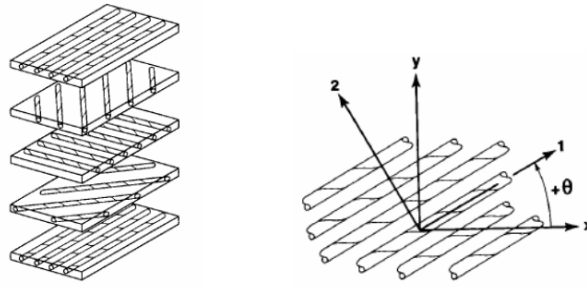


Figura A.2: Lâminas formando o laminado e ângulo θ entre eixo principal do material 1-2 e sistema x-y do laminado (Almeida, 2006)

sendo que A , a área do elemento, e os termos b_i e c_i são dados por

$$A = \frac{1}{2} (b_1 c_2 - b_2 c_1) \quad (\text{A.12})$$

com $b_i = y_j - y_k$ e $c_i = x_k - x_j$ e $i, j, k = 1, 2, 3$ (com permutação cíclica)

Empregando os procedimentos padrão do FEM para obtenção da matriz de rigidez chega-se à relação deformação-deslocamento de flexão dada por (Batoz et al., 1980)

$$[B_b(\xi, \eta)] = \frac{1}{2A} \begin{bmatrix} y_{31} \{H_{x,\xi}\}^T + y_{12} \{H_{x,\eta}\}^T \\ -x_{31} \{H_{y,\xi}\}^T - x_{12} \{H_{y,\eta}\}^T \\ -x_{31} \{H_{x,\xi}\}^T - x_{12} \{H_{x,\eta}\}^T + y_{31} \{H_{y,\xi}\}^T + y_{12} \{H_{y,\eta}\}^T \end{bmatrix} \quad (\text{A.13})$$

com $x_{ij} = x_i - x_j$, $y_{ij} = y_i - y_j$ e onde $\{H_{x,\xi}\}$, $\{H_{x,\eta}\}$, $\{H_{y,\xi}\}$ e $\{H_{y,\eta}\}$ são as derivadas de $\{H_x\}$ e $\{H_y\}$ em relação a ξ e η , respetivamente, sendo que ξ , η e $1 - \xi - \eta$ são as coordenadas naturais (ou triangulares).

Sendo as funções de forma N_i

$$\begin{aligned} N_1 &= 2(1 - \xi - \eta) \left(\frac{1}{2} - \xi - \eta\right) \\ N_2 &= \xi(2\xi - 1) \\ N_3 &= \eta(2\eta - 1) \\ N_4 &= 4\xi\eta \\ N_5 &= 4\eta(1 - \xi - \eta) \\ N_6 &= 4\xi(1 - \xi - \eta) \end{aligned} \quad (\text{A.14})$$

os vetores $\{H_x\}$, $\{H_y\}$ são vetores cujas nove componentes são as novas funções de interpolação. Essas componentes são escritas em função de N_i , $i = 1$ a 6 e das coordenadas dos nós, como segue

$$\begin{aligned} H_{x1} &= 1,5(a_6 N_6 - a_5 N_5) \\ H_{x2} &= b_5 N_5 - b_6 N_6 \\ H_{x3} &= N_1 - c_5 N_5 - c_6 N_6 \\ H_{y1} &= 1,5(d_6 N_6 - d_5 N_5) \\ H_{y2} &= -N_1 + e_5 N_5 + e_6 N_6 \\ H_{y3} &= -H_{x2} \end{aligned} \quad (\text{A.15})$$

As funções H_{x4} , H_{x5} , H_{x6} , H_{y4} , H_{y5} e H_{y6} são obtidas através das expressões acima, substituindo N_1 por N_2 e os índices 6 e 5 por 4 e 6, respectivamente. As funções H_{x7} , H_{x8} , H_{x9} , H_{y7} , H_{y8} e H_{y9} são obtidas substituindo N_1 por N_3 e os índices 6 e 5 por 5 e 4, respectivamente. As outras variáveis que compõem as equações são

$$\begin{aligned}
 a_k &= \frac{-x_{ij}}{l_{ij}^2} \\
 b_k &= \frac{3x_{ij}y_{ij}}{4l_{ij}^2} \\
 c_k &= \frac{\left(\frac{1}{4}x_{ij}^2 - \frac{1}{2}y_{ij}^2\right)}{l_{ij}^2} \\
 d_k &= \frac{-y_{ij}}{l_{ij}^2} \\
 e_k &= \frac{\left(\frac{1}{4}y_{ij}^2 - \frac{1}{2}x_{ij}^2\right)}{l_{ij}^2}
 \end{aligned} \tag{A.16}$$

onde $k = 4, 5, 6$ são os nós intermediários dos lados $ij = 23, 31$ e 12 , respectivamente. Também são utilizadas as constantes geométricas $x_{ij} = x_i - x_j$ e $y_{ij} = y_i - y_j$, e l_{ij} é o comprimento do lado ij .

As derivadas de $\{H_x\}$ e $\{H_y\}$ em relação a ξ e η vem dadas por

$$\{H_{x,\xi}\} = \left\{ \begin{array}{c} P_6 (1 - 2\xi) + (P_5 - P_6) \eta \\ q_6 (1 - 2\xi) - (q_5 + q_6) \eta \\ -4 + 6 (\xi + \eta) + r_6 (1 - 2\xi) - \eta (r_5 + r_6) \\ -P_6 (1 - 2\xi) + \eta (P_4 + P_6) \\ q_6 (1 - 2\xi) - \eta (q_6 - q_4) \\ -2 + 6\xi + r_6 (1 - 2\xi) + \eta (r_4 - r_6) \\ -\eta (P_5 + P_4) \\ \eta (q_4 - q_5) \\ -\eta (r_5 - r_4) \end{array} \right\} \tag{A.17}$$

$$\{H_{y,\xi}\} = \left\{ \begin{array}{c} t_6 (1 - 2\xi) + \eta (t_5 - t_6) \\ 1 + r_6 (1 - 2\xi) - \eta (r_5 + r_6) \\ -q_6 (1 - 2\xi) - \eta (q_5 + q_6) \\ -t_6 (1 - 2\xi) + \eta (t_4 + t_6) \\ -1 + r_6 (1 - 2\xi) + \eta (r_4 - r_6) \\ -q_6 (1 - 2\xi) - \eta (q_4 - q_6) \\ -\eta (t_4 + t_5) \\ \eta (r_4 - r_5) \\ -\eta (q_4 - q_5) \end{array} \right\} \tag{A.18}$$

$$\{H_{x,\eta}\} = \left\{ \begin{array}{c} -P_5(1-2\eta) - \xi(P_6 - P_5) \\ q_5(1-2\eta) - \xi(q_5 + q_6) \\ -4 + 6(\xi + \eta) + r_5(1-2\eta) - \xi(r_5 + r_6) \\ \xi(P_4 + P_6) \\ \xi(q_4 - q_6) \\ -\xi(r_6 - r_4) \\ P_5(1-2\eta) - \xi(P_4 + P_5) \\ q_5(1-2\eta) + \xi(q_4 - q_5) \\ -2 + 6\eta + r_5(1-2\eta) + \xi(r_4 - r_5) \end{array} \right\} \quad (\text{A.19})$$

$$\{H_{y,\eta}\} = \left\{ \begin{array}{c} -t_5(1-2\eta) - \xi(t_6 - t_5) \\ 1 + r_5(1-2\eta) - \xi(r_5 + r_6) \\ -q_5(1-2\eta) + \xi(q_5 + q_6) \\ \xi(t_4 + t_6) \\ \xi(r_4 - r_6) \\ -\xi(q_4 - q_6) \\ t_5(1-2\eta) - \xi(t_4 + t_5) \\ -1 + r_5(1-2\eta) + \xi(r_4 - r_5) \\ -q_5(1-2\eta) - \xi(q_4 - q_5) \end{array} \right\} \quad (\text{A.20})$$

onde

$$\begin{aligned} P_k &= -6x_{ij}/l_{ij}^2 = 6a_k & q_k &= -6y_{ij}/l_{ij}^2 = 6d_k \\ t_k &= 3x_{ij}y_{ij}/l_{ij}^2 = 4b_k & r_k &= 3y_{ij}^2/l_{ij}^2 \end{aligned} \quad (\text{A.21})$$

sendo $k = 4, 5$ e 6 para $ij = 23, 31$ e 12 , respectivamente

A.1.2 Análise geométrica não-linear

A análise não-linear geométrica é tratada por um esquema incremental-iterativo, dado por

$$([\mathop{t}K_L] + [\mathop{t}K_{NL}]) \{\Delta U^{(i)}\} = \{^{t+\Delta t}P\} - \{^{t+\Delta t}F^{(i-1)}\} \quad (\text{A.22})$$

onde o superíndice direito indica o número de iteração que é utilizada a formulação Lagrangeana atualizada, sendo que a configuração de referência corresponde ao tempo em que as quantidades são calculadas, tempo representado pelo superíndice esquerdo (Bathe e Ho, 1981).

K_L e K_{NL} são as matrizes de rigidez linear e não linear, respectivamente, e F é o vetor de forças internas.

O vetor de deslocamento para um instante $t + \Delta t$ é obtido por

$$\{^{t+\Delta t}U^{(i)}\} = \{^{t+\Delta t}U\} + \{\Delta U^{(i)}\} \quad (\text{A.23})$$

O sistema global de coordenadas é fixo; adicionalmente considera-se um sistema local fixo ao corpo, do qual são obtidas a matriz de rigidez e o vetor de forças que são transformados logo para o sistema global fixo.

A matriz de rigidez não linear é dada por (Bathe e Ho, 1981)

$$[{}^t K_{NL}] = \int_A [B_{NL}]^T [{}^t \hat{N}] [B_{NL}] dA \quad (\text{A.24})$$

sendo

$$[{}^t \hat{N}] = \begin{bmatrix} [{}^t N^*] & & \\ & [{}^t N^*] & \\ & & [{}^t N^*] \end{bmatrix} \quad (\text{A.25})$$

$$[{}^t N^*] = \begin{bmatrix} {}^t N_{xx} & {}^t N_{xy} \\ {}^t N_{xy} & {}^t N_{yy} \end{bmatrix} \quad (\text{A.26})$$

$$\{{}^t N\} = \begin{Bmatrix} {}^t N_{xx} \\ {}^t N_{yy} \\ {}^t N_{xy} \end{Bmatrix} = [D_m] \{{}^t \varepsilon^0\} + [D_{mb}] \{{}^t \kappa\} \quad (\text{A.27})$$

$$\{{}^t \varepsilon^0\} = [B_m] \{{}^t \bar{U}_m\} \quad (\text{A.28})$$

$$\{{}^t \kappa\} = [B_b] \{\Delta^t U_b\} + \{{}^{t-\Delta} \kappa\} \quad (\text{A.29})$$

$$[B_{NL}] = \frac{1}{2A} \begin{bmatrix} b_1 & 0 & 0 & b_2 & 0 & 0 & 0 & 0 & 0 \\ c_1 & 0 & 0 & c_2 & 0 & 0 & c_3 & 0 & 0 \\ 0 & b_1 & 0 & 0 & b_2 & 0 & 0 & 0 & 0 \\ 0 & c_1 & 0 & 0 & c_2 & 0 & 0 & c_3 & 0 \\ 0 & 0 & b_1 & 0 & 0 & b_2 & 0 & 0 & 0 \\ 0 & 0 & c_1 & 0 & 0 & c_2 & 0 & 0 & c_3 \end{bmatrix} \quad (\text{A.30})$$

com $b_i = y_j - y_k$ e $c_i = x_k - x_j$ sendo $i, j, k = 1, 2, 3$ com permutação cíclica.

O vetor de forças equivalentes referente às forças internas de membrana e as forças internas devido ao acoplamento membrana-flexão são dados por

$$\{{}^t F_m^{(i)}\} = [K_m] \{{}^t \bar{U}_m^{(i)}\} + \{{}^t F_{mb}^{(i)}\} \quad (\text{A.31})$$

$$\{{}^t F_{mb}^{(i)}\} = [K_b] \{\Delta^t U_b^{(i)}\} + \{{}^t F_{mb}^{(i-1)}\} \quad (\text{A.32})$$

O vetor de momentos é dado por

$$\{{}^t M\} = \begin{Bmatrix} {}^t M_{xx} \\ {}^t M_{yy} \\ {}^t M_{xy} \end{Bmatrix} = [D_{mb}]^T \{{}^t \varepsilon^0\} + [D_b] \{{}^t \kappa\} \quad (\text{A.33})$$

O vetor de forças nodais equivalentes relativo aos graus de liberdade de flexão é dado por

$$\{{}^t F_b^{(i)}\} = \{{}^t F_{bb}^{(i)}\} + \{{}^t F_{bm}^{(i)}\} \quad (\text{A.34})$$

onde

$$\left\{ {}^t F_{bb}^{(i)} \right\} = [K_b] \left\{ \Delta {}^t U_b^{(i)} \right\} + \left\{ {}^t F_{bb}^{(i-1)} \right\} \quad (\text{A.35})$$

$$\left\{ {}^t F_{bm}^{(i)} \right\} = [K_{mb}]^T \left\{ {}^t \bar{U}_m^{(i)} \right\} \quad (\text{A.36})$$

A.1.3 Matriz de massa consistente

A matriz de massa do elemento também é composta pela superposição das matrizes de massa do CST, e do DKT; também é adicionado um termo de massa para a rotação perpendicular ao plano seguindo o mesmo critério da matriz de rigidez. O termo de rotação perpendicular ao plano é adotado (To e Wang, 1998).

$$J = \frac{\rho h A}{18} (r_2^2 + s_3^2 + r_3(r_3 - r_2)) \quad (\text{A.37})$$

para cada nó, sendo r_i e s_i as coordenadas do nó i no sistema local de coordenadas com centro no nó 1 e o eixo r coincidente com o lado 1 – 2 como mostrado na figura A.3. A

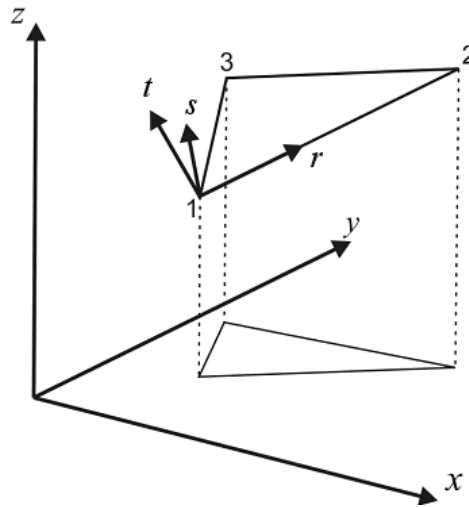


Figura A.3: Eixos locais do elemento (López, 2009)

matriz de massa do elemento CST é a seguinte

$$m_{CST} = \frac{\rho h A}{12} \begin{bmatrix} 2 & 0 & 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 & 0 & 1 \\ 1 & 0 & 2 & 0 & 1 & 0 \\ 0 & 1 & 0 & 2 & 0 & 1 \\ 1 & 0 & 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 1 & 0 & 2 \end{bmatrix} \quad (\text{A.38})$$

A matriz de massa do DKT é apresentada a seguir (Luo e Hutton, 2002).

$$m_{DKT} = \rho h \int_A \left[N_w N_w^T + \frac{h^2}{12} (N_x N_x^T + N_y N_y^T) \right] dA \quad (\text{A.39})$$

que também pode ser escrita da seguinte forma

$$m_{DKT} = \rho h \left[N_{wc} L_M N_{wc}^T + \frac{h^2}{12} (N_{xc} L_N N_{xc}^T + N_{yc} L_N N_{yc}^T) \right] \quad (\text{A.40})$$

onde L_M é obtida da seguinte equação

$$L_M = \int_A L_w L_w^T dA \quad (\text{A.41})$$

sendo

$$L_w^T = \{1 \quad L_2 \quad L_3 \quad L_2^2 \quad L_2 L_3 \quad L_3^2 \quad L_2^3 \quad L_2^2 L_3 \quad L_2 L_3^2 \quad L_3^3\} \quad (\text{A.42})$$

onde L_1 , L_2 e L_3 são as coordenadas triangulares.

As seguintes variáveis são obtidas com a permutação cíclica $i = 1, 2, 3$; $j = 2, 3, 1$; $k = 3, 1, 2$

$$L_i = \frac{1}{2A} (a_i + b_i x_i + c_i y_i) \quad (\text{A.43})$$

$$b_i = y_j - y_k, \quad c_i = x_k - x_j \quad (\text{A.44})$$

$$l_i = b_i^2 + c_i^2 \quad (\text{A.45})$$

$$b_{i+3} = 3b_i c_i / (4l_i) \quad (\text{A.46})$$

$$c_{i+3} = (2b_i^2 - c_i^2) / (4l_i), \quad c_{i+6} = (2c_i^2 - b_i^2) / (4l_i) \quad (\text{A.47})$$

$$a_i = x_j y_k - x_k y_j, \quad a_{i+3} = c_i / l_i, \quad a_{i+6} = b_i / l_i \quad (\text{A.48})$$

$$b_{ij} = b_i b_j, \quad c_{ij} = c_i c_j, \quad e_{ij} = b_i c_j \quad (\text{A.49})$$

$$r_{3i-1} = (c_k - c_j) / 2, \quad r_{3i} = (b_j - b_k) / 2 \quad (\text{A.50})$$

L_M e L_N são apresentadas a seguir

$$L_M = \frac{A}{840} \begin{bmatrix} 840 & 280 & 280 & 140 & 70 & 140 & 84 & 28 & 28 & 84 \\ 280 & 140 & 70 & 84 & 28 & 28 & 56 & 14 & 28/3 & 14 \\ 280 & 70 & 140 & 28 & 28 & 84 & 14 & 28/3 & 14 & 56 \\ 140 & 84 & 28 & 56 & 14 & 28/3 & 40 & 8 & 4 & 4 \\ 70 & 28 & 28 & 14 & 28/3 & 14 & 8 & 4 & 4 & 8 \\ 140 & 28 & 84 & 28/3 & 40 & 56 & 4 & 4 & 8 & 40 \\ 84 & 56 & 14 & 40 & 8 & 4 & 30 & 5 & 2 & 3/2 \\ 28 & 14 & 28/3 & 8 & 4 & 4 & 5 & 2 & 3/2 & 2 \\ 28 & 28/3 & 14 & 4 & 4 & 8 & 2 & 3/2 & 2 & 5 \\ 84 & 14 & 56 & 4 & 8 & 40 & 3/2 & 2 & 5 & 30 \end{bmatrix} \quad (\text{A.51})$$

$$L_N = \frac{A}{90} \begin{bmatrix} 90 & 30 & 30 & 15 & 15/2 & 15 \\ 30 & 15 & 15/2 & 9 & 3 & 3 \\ 30 & 15/2 & 15 & 3 & 3 & 9 \\ 15 & 9 & 3 & 6 & 3/2 & 1 \\ 15/2 & 3 & 3 & 3/2 & 1 & 3/2 \\ 15 & 3 & 9 & 1 & 3/2 & 6 \end{bmatrix} \quad (\text{A.52})$$

Em função das variáveis definidas nas equações (A.43) até (A.50), apresentam-se as matrizes N_{wc} , N_{xc} e N_{yc} a seguir

$$N_{wc} = \begin{bmatrix} 1 & 0 & 0 & -3 & -4 & -3 & 2 & 4 & 4 & 2 \\ 0 & c_3 & -c_2 & -2c_3 & -3r_2 & 2c_2 & c_3 & c_3 + r_2 & r_2 - c_2 & -c_2 \\ 0 & -b_3 & b_2 & 2b_3 & -3r_3 & -2b_2 & -b_3 & r_3 - b_3 & b_2 + r_3 & b_2 \\ 0 & 0 & 0 & 3 & 2 & 0 & -2 & -2 & -2 & 0 \\ 0 & 0 & 0 & -c_3 & r_5 & 0 & c_3 & c_1 + c_3 - r_5 & -r_5 & 0 \\ 0 & 0 & 0 & b_3 & r_6 & 0 & -b_3 & -b_3 - b_1 - r_6 & -r_6 & 0 \\ 0 & 0 & 0 & 0 & 2 & 3 & 0 & -2 & -2 & -2 \\ 0 & 0 & 0 & 0 & r_8 & c_2 & 0 & -r_8 & -c_1 - c_2 - r_8 & -c_2 \\ 0 & 0 & 0 & 0 & r_9 & -b_2 & 0 & -r_9 & b_1 + b_2 - r_9 & b_2 \end{bmatrix} \quad (\text{A.53})$$

$$N_{xc} = \begin{bmatrix} 0 & -6a_6 & 6a_5 & 6a_6 & 6(a_6 - a_5) & -6a_5 \\ 1 & -3 + 4c_6 & -3 + 4c_5 & 2 - 4c_6 & 4(1 - c_5 - c_6) & 2 - 4c_5 \\ 0 & 4b_6 & 4b_5 & -4b_6 & -4(b_5 + b_6) & -4b_5 \\ 0 & 6a_6 & 0 & -6a_6 & -6(a_4 + a_6) & 0 \\ 0 & -1 + 4c_6 & 0 & 2 - 4c_6 & 4(c_4 - c_6) & 0 \\ 0 & 4b_6 & 0 & -4b_6 & 4(b_4 - b_6) & 0 \\ 0 & 0 & -6a_5 & 0 & 6(a_4 + a_5) & 6a_5 \\ 0 & 0 & -1 + 4c_5 & 0 & 4(c_4 - c_5) & 2 - 4c_5 \\ 0 & 0 & 4b_5 & 0 & 4(b_4 - b_5) & -4b_5 \end{bmatrix} \quad (\text{A.54})$$

$$N_{yc} = \begin{bmatrix} 0 & 6a_9 & -6a_8 & -6a_9 & 6(a_8 - a_9) & 6a_8 \\ 0 & 4b_6 & 4b_5 & -4b_6 & -4(b_5 + b_6) & -4b_5 \\ 1 & -3 + 4c_9 & -3 + 4c_8 & 2 - 4c_9 & 4(1 - c_8 - c_9) & 2 - 4c_8 \\ 0 & -6a_9 & 0 & 6a_9 & 6(a_7 + a_9) & 0 \\ 0 & 4b_6 & 0 & -4b_6 & 4(b_4 - b_6) & 0 \\ 0 & -1 + 4c_9 & 0 & 2 - 4c_9 & 4(c_7 - c_9) & 0 \\ 0 & 0 & 6a_8 & 0 & -6(a_7 + a_8) & -6a_8 \\ 0 & 0 & 4b_5 & 0 & 4(b_4 - b_5) & -4b_5 \\ 0 & 0 & -1 + 4c_8 & 0 & 4(c_7 - c_8) & 2 - 4c_8 \end{bmatrix} \quad (\text{A.55})$$

Assim, com a matriz do DKT (9×9), CST (6×6) e J (3×3) é formada a matriz de massa consistente do sistema, M (18×18), com a qual é montado o sistema generalizado de autovalores

$$[K]\Phi = \omega^2[M]\Phi \quad (\text{A.56})$$

de onde são obtidas as frequências ω e modos Φ de vibração da estrutura.

Foi utilizada uma subrotina em Fortran para o problema de autovalores generalizado, da

biblioteca IMSL, onde a entrada de dados é dada pela matriz quadrada de rigidez e massa, e como saída de dados tem-se o vetor de autovalores e a matriz de autovetores.