



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ESTRUTURAL E CONSTRUÇÃO CIVIL
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA CIVIL:
ESTRUTURAS E CONSTRUÇÃO CIVIL

ELIAS SARAIVA BARROSO

ANÁLISE E OTIMIZAÇÃO DE ESTRUTURAS LAMINADAS UTILIZANDO A
FORMULAÇÃO ISOGOMÉTRICA

FORTALEZA

2015

ELIAS SARAIVA BARROSO

ANÁLISE E OTIMIZAÇÃO DE ESTRUTURAS LAMINADAS UTILIZANDO A
FORMULAÇÃO ISOGOMÉTRICA

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Civil: Estruturas e Construção Civil, da Universidade Federal do Ceará, como requisito parcial para obtenção do Título de Mestre em Engenharia Civil. Área de Concentração: Estruturas.

Orientador: Evandro Parente Junior, D.Sc.

Coorientador: Antônio Macário Cartaxo de Melo, D.Sc.

FORTALEZA

2015

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca de Ciências e Tecnologia

-
- B285a Barroso, Elias Saraiva.
Análise e otimização de estruturas laminadas utilizando a formulação isogeométrica. / Elias Saraiva Barroso .– 2015.
161 f.: il., enc.; color.
- Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Tecnologia, Departamento de Engenharia Estrutural e Construção Civil, Programa de Pós-Graduação em Engenharia Estrutural e Construção Civil, Fortaleza, 2015.
Área de Concentração: Estruturas.
Orientação: Prof. Dr. Evandro Parente Junior.
Coorientação: Prof. Dr. Antônio Macário Cartaxo de Melo.
1. Engenharia de estruturas. 2 Método dos Elementos Finitos. 3. Programação Orientada a Objetos (Computação). I. Título.

ELIAS SARAIVA BARROSO

ANÁLISE E OTIMIZAÇÃO DE ESTRUTURAS LAMINADAS UTILIZANDO A
FORMULAÇÃO ISOGEOMÉTRICA

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Civil: Estruturas e Construção Civil, da Universidade Federal do Ceará, como requisito parcial para obtenção do Título de Mestre em Engenharia Civil. Área de Concentração: Estruturas.

Aprovada em 29 /09 /2015.

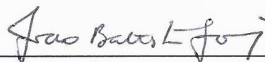
BANCA EXAMINADORA



Prof. Evandro Parente Junior, Dr. (Orientador)
Universidade Federal do Ceará – UFC



Prof. Antônio Macário Cartaxo de Melo, Dr. (Coorientador)
Universidade Federal do Ceará – UFC



Prof. João Batista Marques de Sousa Junior, Dr.
Universidade Federal do Ceará – UFC



Prof. Luiz Fernando Campos Ramos Martha, Dr.
Pontifícia Universidade Católica do Rio de Janeiro – PUC-RIO

A Deus.

A meus pais, Eliseu e Maria José, por todo o amor e dedicação que me deram durante toda minha vida.

AGRADECIMENTOS

A Deus, por ter me concedido a oportunidade de realizar este trabalho.

A meus irmãos Eliseu Neto e Lígia, pela grande amizade.

À minha namorada Renata, pela motivação, carinho e companherismo que sempre me deu.

A meu orientador Evandro Parente Junior, pela amizade e exemplo de dedicação e honestidade que testemunhei ao longo destes 5 anos que trabalhamos juntos.

A meu coorientador Antônio Macário Cartaxo de Melo, pelas contribuições realizadas em todos os trabalhos que fizemos em conjunto.

Aos professores Luiz Fernando Martha e João Batista, por fazerem parte de minha banca examinadora.

Aos colegas Carlos David, Rafael Wandson e Pedro Sanderson. Pela amizade, incentivo e ajuda dados nos momentos mais difíceis que passei ao longo deste mestrado.

A UFC, DEECC, LMCV e CAPES, que me deram suporte financeiro para realização deste trabalho.

”Efetivamente, em qualquer lugar onde olhares, a sabedoria te fala pelos vestígios que imprimiu em todas as suas obras”

Santo Agostinho

RESUMO

As estruturas laminadas são fabricadas utilizando um conjunto de camadas de material compósito empilhadas em uma sequência determinada de forma a se obter um desempenho estrutural adequado. Atualmente, a análise de estruturas laminadas é realizada principalmente utilizando o Método dos Elementos Finitos (MEF). Contudo, este método não é capaz de representar exatamente geometrias complexas. Uma alternativa ao MEF é a Análise Isogeométrica (AIG). A AIG utiliza na análise numérica as mesmas funções utilizadas pelo sistemas CAD para Modelagem Geométrica, como as B-Splines e NURBS, permitindo que a geometria dos modelos seja representada de forma exata para qualquer nível de discretização adotado. O presente trabalho utilizou a formulação isogeométrica baseada em NURBS para realizar a análise não linear geométrica de estruturas laminadas. Esta formulação foi implementada em um software acadêmico de análise por elementos finitos. Utilizando uma formulação apropriada do método e o paradigma de Programação Orientada a Objetos (POO), foi possível minimizar as alterações realizadas na estrutura deste programa para a implementação da Análise Isogeométrica, inclusive em problemas de estruturas laminadas. A verificação da implementação foi realizada com base em exemplos disponíveis na literatura. Exemplos de análises lineares e não-lineares de estruturas com material isotrópico e compósito laminado foram realizados, tendo obtido excelentes resultados. No projeto de estruturas laminadas é necessário determinar o número de camadas de material compósito e as características de cada camada (material, espessura e orientação das fibras). Pelo fato de existirem um grande número de combinações possíveis, o procedimento padrão de tentativa e erro não é apropriado, sendo necessário a utilização de técnicas de otimização. Algoritmos de otimização bio-inspirados, como Algoritmos Genéticos e Nuvem de Partículas, apresentam bom desempenho em problemas de otimização combinatória. Considerando estes aspectos, no presente trabalho foi desenvolvido um algoritmo híbrido, baseado nos métodos da Nuvem de Partículas e Algoritmo Genético, para otimização de estruturas laminadas. Algumas variantes do algoritmo proposto foram comparadas considerando vários exemplos de otimização. Um processo de calibração dos parâmetros numéricos do algoritmo de otimização foi realizado, de modo a permitir uma comparação isenta entre as variantes. Estas variantes foram utilizadas na otimização de placas e cascas laminadas. No caso de cascas, a análise isogeométrica foi utilizada como ferramenta de análise estrutural. Os resultados obtidos mostraram que o método de otimização proposto apresentou desempenho comparável com Algoritmos Genéticos na otimização de laminados tradicionais, onde a orientação das fibras é limitada a poucos ângulos. Por outro lado, o método proposto obteve desempenho superior ao Algoritmo Genético na otimização de laminados dispersos.

Palavras-chave: Análise Isogeométrica, Estruturas laminadas, Otimização estrutural.

ABSTRACT

The laminate structures are made using a set of layers of a composite material stacked in a particular sequence in order to obtain a good structural performance. Currently, the analysis of laminated structures is mainly performed using the Finite Element Method (FEM). However, this method is not able to accurately represent complex geometries. An alternative to the FEM is the Isogeometric Analysis (IGA). IGA uses in the numerical analysis the same functions used by Geometric Modeling in CAD systems, as B-splines and NURBS, allowing an exact representation of the geometry regardless of model discretization level. This study used the isogeometric formulation based on NURBS for performing geometric nonlinear analysis of laminated structures. This formulation was implemented in an academic finite element software. Using an appropriate formulation of the method and the Object Oriented Programming (OOP), it was possible to minimize the changes made in the structure of the program for implementing the Isogeometric Analysis, including in laminated structures problems. The verification of the implementation is carried out based on available examples in literature. Several examples of linear and non-linear analyzes of structures with isotropic and laminated composite material were performed and they obtained excellent results. In laminated structures project, it is necessary to determine the number of layers of composite material and the characteristics of each layer (material, thickness, and fiber orientation). Because there are numerous possible combinations, the standard procedure based on trial and error is not appropriate, requiring the use of optimization techniques. Bio-inspired optimization algorithms, such as Genetic Algorithms and Particle Swarm Optimization, perform well in combinatorial optimization problems. Considering these aspects, the present study was developed a hybrid algorithm, based on the Particle Swarm Optimization and Genetic Algorithm methods for optimization of laminated structures. Some variants of the proposed algorithm were compared considering several optimization examples. A calibration process of the algorithm parameters was conducted in order to avoid biased results. These variants were used in the optimization of laminated plates and shells. In the case of shells, the isogeometric analysis was used as a structural analysis tool. The results showed that the proposed optimization method presents comparable performance with the genetic algorithms in traditional laminates optimization, where the orientation of the fibers is limited to a few angles. Moreover, the proposed method outperforms genetic algorithm in the optimization of dispersed laminates.

Keywords: Isogeometric Analysis, Laminated structures, Structural optimization.

LISTA DE ILUSTRAÇÕES

Figura 1 – Compósito laminado.	15
Figura 2 – Sistemas de coordenadas global (x, y, z) , sistema global do laminado $(\hat{x}, \hat{y}, \hat{z})$, e o sistema local da lâmina (x_1, x_2, x_3) , assim como seus vetores unitários $(\hat{e}_1, \hat{e}_2, \hat{e}_3)$	16
Figura 3 – Esquema geral de laminação.	20
Figura 4 – Etapas de uma análise estrutural.	24
Figura 5 – Exemplo de otimização de forma.	25
Figura 6 – Exemplo de curvas de Bézier de grau 3.	26
Figura 7 – Definição de contorno de fontes utilizando representação de Bézier.	27
Figura 8 – Polinômios de Bernstein para vários graus.	29
Figura 9 – Algoritmo de <i>de Casteljau</i> aplicado a curva de Bézier cúbica.	30
Figura 10 – Efeito da variação do peso.	31
Figura 11 – Representação de um arco de círculo utilizando Bézier Racional.	31
Figura 12 – Exemplo de superfície Bézier.	32
Figura 13 – Exemplo de sólido de Bézier.	33
Figura 14 – Funções base B-spline quadráticas.	35
Figura 15 – Funções de base B-spline cúbica.	35
Figura 16 – Efeito da multiplicidade em uma B-spline quadrática.	37
Figura 17 – Funções de base B-spline quadráticas com $\Xi = [0, 0, 0, 0, 0.5, 0.5, 1, 1, 1, 1]$	37
Figura 18 – Exemplo de inserção de knot em curva B-spline ($\bar{\xi} = 2$).	39
Figura 19 – Controle local obtido pelo inserção de <i>knots</i>	40
Figura 20 – Elevação de grau aplicada a uma curva B-spline quadrática.	41
Figura 21 – Curva NURBS $C(\xi)$ em \mathbb{R}^2 sendo obtida da projeção de uma curva B-spline $C^w(\xi)$ em \mathbb{R}^3	42
Figura 22 – Bases NURBS e B-spline quadráticas com $\Xi = [0, 0, 0, 0.5, 1, 1, 1]$	43
Figura 23 – Semi circunferência construída com NURBS e um arco de 180° construído com B-spline.	44
Figura 24 – Exemplo de superfície NURBS.	45
Figura 25 – Exemplo de um sólido NURBS com seus pontos de controle.	47
Figura 26 – Exemplo de modelo considerando múltiplos <i>patches</i>	48
Figura 27 – Superfície NURBS modelada com quatro <i>patches</i>	48
Figura 28 – Malhas da AIG e MEF para um problema bidimensional.	50
Figura 29 – Descrição de uma estrutura por NURBS.	51
Figura 30 – Corpo de volume V submetido a forças de corpo \mathbf{b} , cargas de superfícies \mathbf{q} , cargas pontuais \mathbf{F}_i	56
Figura 31 – Elementos isogeométricos do <i>patch</i>	59

Figura 32 – Geometria, graus de liberdade e funções de base para um modelo de cabo.	60
Figura 33 – Geometria modelada com múltiplos <i>patches</i> . Os pontos de controle azul resultam em graus de liberdade compartilhados.	61
Figura 34 – Mapeamentos realizados na AIG para integração numérica.	62
Figura 35 – Refinamento p realizado na NURBS construída pela Figura 13.	62
Figura 36 – Refinamento h realizado na NURBS construída pela Figura 13.	63
Figura 37 – Refinamento k realizado na NURBS definida pela Figura 13.	63
Figura 38 – Exemplo de uma curva B-Spline cúbica com $\Xi = [0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4]$.	68
Figura 39 – Sequência de modificações nos pontos de controle e funções base obtidas pela inserção dos <i>knots</i> $\bar{\Xi} = [1, 1, 2, 2, 3, 3]$, na curva ilustrada pela Figura 38. .	69
Figura 40 – Bases B-Spline e Bernstein do primeiro <i>knot span</i> da curva dada pela Figura 38.	71
Figura 41 – Obtenção das bases B-Spline por combinação linear das bases Bernstein.	72
Figura 42 – Algoritmo de cálculo das matrizes de extração \bar{C}^e univariante de cada elemento.	74
Figura 43 – Rotina para cálculo das funções de forma e suas derivadas para o elemento sólido isogeométrico.	76
Figura 44 – Diagrama de classes UML ⁵⁷ descrevendo a arquitetura geral do programa FAST.	77
Figura 45 – Diagrama classes UML da classe <i>cControl</i>	78
Figura 46 – Diagrama classes UML da classe <i>cShape</i>	79
Figura 47 – Trecho referente descrição da geometria de um arquivo de entrada do FAST AIG.	80
Figura 48 – Diagrama classes UML da classe <i>cPatch</i>	81
Figura 49 – Exemplo de resultados da AIG visualizado através de um pós-processador de elementos finitos.	82
Figura 50 – Código em linguagem C++ da etapa de montagem da matriz de rigidez global paralelizada no programa FAST.	83
Figura 51 – Descrição do Exemplo 5.3.1.	84
Figura 52 – Malha de pontos de controle da geometria básica de um quarto de cilindro.	85
Figura 53 – Malhas de 1-4 utilizadas no Exemplo 5.3.1.	86
Figura 54 – Resultados utilizando refinamento h	87
Figura 55 – Resultados referentes à tensão radial utilizando refinamento p	88
Figura 56 – Resultados referentes à tensão radial utilizando refinamento k	88
Figura 57 – Descrição do Exemplo 5.3.2.	89
Figura 58 – Malhas de 1-5 utilizadas no Exemplo 5.3.2.	90
Figura 59 – Deslocamento radial considerando diferentes graus.	91
Figura 60 – Deformada do Exemplo 5.3.2.	92
Figura 61 – Descrição do Exemplo 5.3.3.	92

Figura 62 – Malhas 1-5 utilizadas no Exemplo 5.3.3.	93
Figura 63 – Convergência de w_p considerando NURBS de vários graus.	93
Figura 64 – Deformada com fator de escala = $2e6$	94
Figura 65 – Descrição do Exemplo 5.3.4	96
Figura 66 – Malhas 1-5 utilizadas no Exemplo 5.3.4.	96
Figura 67 – Resultados obtidos do Exemplo 5.3.4.	97
Figura 68 – Deformada da casca Scordelis-Lo (Malha 5 com $p = 4$), com fator de escala = 40.	97
Figura 69 – Descrição do Exemplo 5.3.5.	99
Figura 70 – Curvas carga x deslocamento obtidas da análise da viga curva.	100
Figura 71 – Deformada Curved Beam.	101
Figura 72 – Descrição do Exemplo 5.3.6.	102
Figura 73 – Curva carga \times deslocamento considerando material isotrópico.	103
Figura 74 – Curva carga \times deslocamento considerando material compósito.	103
Figura 75 – Deformada semicilindro engastado $[0^\circ 90^\circ 0^\circ]$	104
Figura 76 – Tempos de execução das análises.	105
Figura 77 – Exemplo de saída do programa <i>gprof</i>	106
Figura 78 – Uso percentual de cada processador durante a execução do FAST em paralelo.	107
Figura 79 – Processo de Codificação e Decodificação.	111
Figura 80 – Mapeamento da função objetivo.	114
Figura 81 – Probabilidades de seleção por ranking.	115
Figura 82 – Processo de cruzamento sendo aplicado a uma lâmina.	115
Figura 83 – Processo de mutação sendo aplicado.	116
Figura 84 – Exemplo de aplicação do operador troca de lâmina.	117
Figura 85 – Exemplo de aplicação dos operadores de adição e eliminação de lâmina.	118
Figura 86 – Pseudo-código do Algoritmo Genético.	119
Figura 87 – Topologias da nuvem.	121
Figura 88 – Manipulação das restrições laterais.	122
Figura 89 – Pseudo-código do algoritmo PSO híbrido proposto.	123
Figura 90 – Placa laminada sujeita a carregamento biaxial.	127
Figura 91 – Placa Laminada com carregamento biaxial e de cisalhamento.	136
Figura 92 – Casos de carga biaxial considerados.	137
Figura 93 – Resultados obtidos do Exemplo 6.7.1, com $\Delta\theta = 45^\circ$	140
Figura 94 – Resultados obtidos do Exemplo 6.7.1, com $\Delta\theta = 15^\circ$	141
Figura 95 – Resultados obtidos do Exemplo 1, com $\Delta\theta = 5^\circ$	142
Figura 96 – Placa Laminada com carregamento biaxial e de cisalhamento.	143
Figura 97 – Descrição do Exemplo 6.7.3.	144
Figura 98 – Resultados de $\overline{f_b}$ por número de avaliações para o caso $\Delta\theta = 45^\circ$	146

- Figura 99 – Resultados de $\overline{f_b}$ por número de avaliações para o caso $\Delta\theta = 15^\circ$ 147
- Figura 100– Resultados de $\overline{f_b}$ por número de avaliações para o caso $\Delta\theta = 5^\circ$ 147

LISTA DE TABELAS

Tabela 1 – Pontos de controle do Sólido de Bézier da Figura 13.	33
Tabela 2 – Intervalo de contribuição das bases B-spline para diferentes vetores Ξ . . .	36
Tabela 3 – Pontos de controle da geometria básica de um quarto de cilindro.	85
Tabela 4 – Propriedades do material compósito.	94
Tabela 5 – Resultados considerando pares de lâminas $[0^\circ 90^\circ]$	95
Tabela 6 – Resultados considerando pares de lâminas $[-45^\circ 45^\circ]$	95
Tabela 7 – Resultados do Exemplo 5.3.4 considerando pares de lâminas $[0^\circ 90^\circ]$. . .	98
Tabela 8 – Resultados do Exemplo 5.3.4 considerando pares de lâminas $[-45^\circ 45^\circ]$. .	98
Tabela 9 – Pontos de controle da geometria básica da viga curva.	99
Tabela 10 – Propriedades elásticas do compósito considerado no Exemplo 5.3.6. . . .	102
Tabela 11 – Estudo de discretização realizado no Exemplo 5.3.7.	104
Tabela 12 – Tempo de execução (s) e <i>speedup</i> obtidos.	105
Tabela 13 – Percentual do tempo de execução total (%).	107
Tabela 14 – Variantes dos algoritmos.	124
Tabela 15 – Propriedades do material utilizado na meta-otimização.	128
Tabela 16 – Valores de cada variável-parâmetro da meta-otimização.	129
Tabela 17 – Funções objetivos ótimas nos problemas do <i>Caso 1</i> e <i>Caso 2</i>	130
Tabela 18 – Funções objetivos ótimas nos problemas do <i>Caso 3</i> ($\Delta\theta = 05^\circ$).	130
Tabela 19 – Resultados das meta-funções objetivos obtidas	131
Tabela 20 – Acréscimo percentual em relação a busca aleatória.	131
Tabela 21 – Parâmetros ótimos para o método <i>GA</i>	132
Tabela 22 – Parâmetros ótimos para o método <i>GA - S</i>	132
Tabela 23 – Parâmetros ótimos para o método <i>PSO - R</i>	132
Tabela 24 – Parâmetros ótimos para o método <i>PSO - GMS</i>	133
Tabela 25 – Novos limites das variável-parâmetro das meta-otimização.	134
Tabela 26 – Parâmetros ótimos dos métodos PSO para caso de problemas de maximização ($\Delta\theta = 45^\circ$).	134
Tabela 27 – Parâmetros ótimos dos métodos AG para caso de problemas de maximização ($\Delta\theta = 45^\circ$).	134
Tabela 28 – Meta-função objetivo ótimas para o caso de problemas de maximização ($\Delta\theta = 45^\circ$).	135
Tabela 29 – Parâmetros ótimos dos métodos PSO para caso de problemas de maximização ($\Delta\theta = 15^\circ$).	135
Tabela 30 – Parâmetros ótimos dos métodos AG para caso de problemas de maximização ($\Delta\theta = 15^\circ$).	135

Tabela 31 – Meta-função objetivo ótimas para o caso de problemas de maximização. ($\Delta\theta = 15^\circ$).	136
Tabela 32 – Propriedades de resistência do compósito considerado nas meta-otimizações de minimização.	137
Tabela 33 – f_{best} para os problemas de minimização de peso.	138
Tabela 34 – Parâmetros ótimos dos métodos PSO para caso de problemas de minimização.	138
Tabela 35 – Parâmetros ótimos dos métodos AG para caso de problemas de minimização.	138
Tabela 36 – Meta-função o objetivo ótimas para o caso de problemas de minimização.	139
Tabela 37 – Esquemas de laminação ótimos encontrados no Exemplo 6.7.1. ($\Delta\theta = 45^\circ$)	140
Tabela 38 – Esquemas de laminação ótimos encontrados no Exemplo 6.7.1. ($\Delta\theta = 15^\circ$)	141
Tabela 39 – Esquema de laminação ótimo encontrado no Exemplo 6.7.1. ($\Delta\theta = 5^\circ$) .	141
Tabela 40 – Número de camadas ótimo do Exemplo 02.	143
Tabela 41 – Propriedades elásticas do compósito carbono-epoxi ²²	145
Tabela 42 – Resistências do compósito carbono-epoxi ²²	145
Tabela 43 – Resultados de projeto ótimo, menor deslocamento w e critério de falha obtidos para cada caso $\Delta\theta$ considerado.	145

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Objetivos e Contribuições	13
1.2	Organização da Dissertação	13
2	COMPÓSITOS LAMINADOS	15
2.1	Mecânica de uma lâmina	15
2.1.1	<i>Critérios de Falha</i>	18
2.2	Teorias de Laminação	20
2.2.1	<i>Teoria Clássica da Laminação</i>	21
3	MODELAGEM GEOMÉTRICA	24
3.1	Representações de Curvas e Superfícies	25
3.2	Curvas de Bézier	26
3.2.1	<i>Bézier Racional</i>	30
3.2.2	<i>Superfícies e Sólidos de Bézier</i>	31
3.3	B-Splines	33
3.3.1	<i>Inserção de knot e elevação de grau</i>	38
3.4	NURBS	41
3.4.1	<i>Superfícies NURBS</i>	44
3.4.2	<i>Sólidos NURBS</i>	46
3.4.3	<i>Múltiplos Patches</i>	47
4	ANÁLISE ISOGEOMÉTRICA	50
4.1	Geometria e Deslocamentos	51
4.2	Deformações e tensões	53
4.2.1	<i>Equações de Equilíbrio</i>	55
4.3	Matriz de Rigidez Tangente	58
4.4	Elemento Isogeométrico	59
4.5	Refinamentos	62
4.6	Avaliação das Tensões	64
4.7	Análise de Sólidos Laminados	64
4.7.1	<i>Integração numérica</i>	65
5	IMPLEMENTAÇÃO COMPUTACIONAL DA AIG	67
5.1	Extração de Bézier	67
5.1.1	<i>Operador de Extração Bézier</i>	69
5.2	Implementação da AIG no programa FAST	77

5.2.1	<i>Arquitetura do programa FAST</i>	77
5.2.2	<i>Implementação da classe cShapeIGA</i>	79
5.2.3	<i>Implementação da classe cPatch</i>	80
5.2.4	<i>Implementação do Pós-processamento</i>	81
5.2.5	<i>Paralelização na montagem da matriz de rigidez</i>	82
5.3	Exemplos Numéricos	83
5.3.1	<i>Cilindro espesso submetido à pressão interna</i>	84
5.3.2	<i>Cilindro fino submetido à pressão interna</i>	89
5.3.3	<i>Cilindro fino com carga concentrada no centro</i>	92
5.3.4	<i>Casca cilíndrica com carga distribuída (Scordelis-Lo)</i>	95
5.3.5	<i>Viga curva</i>	98
5.3.6	<i>Semicilindro engastado</i>	101
5.3.7	<i>Estudo do efeito da paralelização</i>	104
6	OTIMIZAÇÃO DE LAMINADOS	108
6.1	Codificação das variáveis	110
6.2	Tratamento de restrições	111
6.3	Busca Aleatória	112
6.4	Algoritmos Genéticos	113
6.4.1	<i>Função Aptidão e Seleção</i>	114
6.4.2	<i>Cruzamento e Elitismo</i>	115
6.4.3	<i>Mutação</i>	116
6.4.4	<i>Operadores Adicionais para Laminados</i>	116
6.4.5	<i>Pseudo-código</i>	118
6.5	Otimização por Nuvem de Partículas	120
6.5.1	<i>Tratamento das Restrições</i>	122
6.5.2	<i>PSO Híbrido para Otimização de Laminados</i>	123
6.5.3	<i>Variantes</i>	124
6.6	Meta-otimização	124
6.6.1	<i>Relação entre tamanho da população x número de gerações</i>	125
6.6.2	<i>Algoritmo de meta-otimização</i>	126
6.6.3	<i>Resultados</i>	127
6.7	Exemplos de Aplicação	139
6.7.1	<i>Maximização de resistência de uma placa</i>	139
6.7.2	<i>Minimização de peso de uma placa</i>	142
6.7.3	<i>Maximização de rigidez de uma casca cilíndrica</i>	144
7	CONCLUSÃO	148
	REFERÊNCIAS	149

1 INTRODUÇÃO

Materiais compósitos são obtidos pela união de um ou mais materiais em escala macroscópica, com o intuito de produzir um material com características superiores aos de seus componentes individuais. Este trabalho trata dos materiais compósitos reforçados por fibras, que são compostos por fibras de alta resistência e rigidez imersos em uma matriz polimérica.

Os compósitos reforçados por fibras apresentam elevadas relações de rigidez/peso e resistência peso, além de outras propriedades importantes como alta resistência à fadiga e à corrosão e elevado amortecimento estrutural. Devido a estas características, estes compósitos vem sendo cada vez mais utilizados nas indústrias aeronáutica, naval e offshore, automobilística, mecânica, química e na construção civil.

Neste trabalho serão considerados compósitos produzidos com fibras unidirecionais, que são os mais utilizados em aplicações que requerem alto desempenho estrutural. Como as fibras estão alinhadas em uma direção, estes compósitos são tratados em nível macroscópico como materiais homogêneos e ortotrópicos.

É importante notar que o comportamento ortotrópico dos compósitos é responsável por grande parte de suas vantagens, ao permitir colocar as fibras nas direções que requerem mais resistência e rigidez, o que não é possível no caso de materiais isotrópicos. Por outro lado, esta característica também é responsável pela maior complexidade tanto da análise quanto do projeto de estruturas de material compósito, quando comparado às estruturas de material isotrópico.

Os compósitos reforçados por fibras são bastante utilizados na fabricação de componentes laminados, que são produzidos pelo empilhamento de diversas camadas (lâminas) com características distintas. No projeto de estruturas laminadas, é necessário determinar o número de camadas, bem como o material, espessura e orientação das fibra para cada camada, definindo o esquema de laminação utilizado. O esquema de laminação deve ser definido de maneira a que o laminado satisfaça aos requisitos de desempenho impostos pelo projeto, bem como às limitações relacionadas ao processo de fabricação adotado.

Devido ao grande número de variáveis que caracterizam uma estrutura laminada, estas oferecem grande liberdade aos projetistas para obtenção de soluções adaptadas a situações específicas em termos de geometria, apoios e carregamentos. Por outro lado, o método de projeto convencional, baseado em tentativa e erro, não é apropriado para laminados, sendo necessário o uso de técnicas de otimização para explorar de forma completa as vantagens propiciadas pelo uso de compósitos.

Os processos de fabricação utilizados e a disponibilidade de resultados experimentais muitas vezes limitam a espessura da lâmina e a orientação das fibras a valores discretos. A presença de variáveis discretas impede a utilização de métodos de otimização baseados no uso de gradientes (Programação Matemática), pois estes requerem que as variáveis, função objetivo e restrições sejam contínuas.

Dessa forma, a otimização de estruturas laminadas é um problema de otimização combinatória e requer o uso de algoritmos de solução apropriados. Em particular, métodos de otimização bio-inspirados, como os Algoritmos Genéticos¹⁻⁵, Otimização por Colônia de Formigas⁶⁻⁸ e Otimização por Nuvem de Partículas⁹⁻¹², tem apresentado bons resultados na solução desses problemas.

Devido às suas características e processo de fabricação, os laminados são normalmente utilizados na forma de estruturas de parede fina, como placas, cascas e vigas. As cascas são estruturas curvas cuja espessura é muito menor que as outras dimensões. A análise estrutural de cascas laminadas é normalmente realizada através de métodos numéricos, pois as hipóteses cinemáticas utilizadas e equações de equilíbrio resultantes tornam o problema de difícil solução analítica, principalmente no caso de geometrias, carregamentos e condições de contorno complexas. Esse problema é ainda complexo no caso de compósitos laminados, pois existem descontinuidades nos campos de deformação e tensão de uma lâmina para outra.

Atualmente o Método dos Elementos Finitos (MEF) é a opção mais utilizada para realização da análise estrutural tanto de estruturas de materiais homogêneos e isotrópico quanto de compósitos laminados. Na formulação isoparamétrica do MEF, tanto os deslocamentos quanto a geometria da estrutura são aproximados com o uso de funções polinomiais¹³. Assim, salvo no caso de geometria simples, a resposta do MEF contém tanto erros devido a aproximação do campo de deslocamentos, quanto devido a aproximação da geometria do modelo. Ambos os erros são reduzidos a medida que a malha de elementos finitos é refinada, o que pode ser feito aumentando o número de elementos (refinamento- h) ou aumentando o grau dos polinômios utilizados (refinamento- p).

Uma alternativa ao MEF é a Análise Isogeométrica (AIG)^{14,15}. A principal vantagem da AIG é utilizar na análise numérica as mesmas funções utilizadas pelos sistemas CAD para Modelagem Geométrica, como as B-splines e NURBS. Dessa forma, a geometria do problema é representada exatamente independente do nível de discretização considerado, eliminando o erro na representação da geometria que existe no MEF. Adicionalmente, as B-splines e NURBS possuem um grau de continuidade maior que os normalmente utilizados no MEF, fazendo com que a AIG possua uma convergência mais rápida que o MEF.

A discretização do modelo na AIG é obtida pela aplicação de algoritmos clássicos de Modelagem Geométrica, eliminando a etapa de geração de malha no processo de análise estrutural pelo MEF. Assim, o próprio programa de análise trabalha diretamente com a geometria exata do modelo e pode gerenciar os procedimentos de discretização, possibilitando que análises adaptativas sejam realizadas de maneira automática ou semi-automática, de maneira mais simples do que é feito no MEF¹⁴. Outra vantagem da AIG é que ela possui um novo tipo de refinamento do modelo, denominado de refinamento- k , onde tanto o grau do polinômio quanto o número de elementos aumentam simultaneamente. Tal efeito não é observado nos refinamentos tradicionais do MEF (refinamento- p e refinamento- h)¹⁴.

Apesar do grande interesse na extensão da AIG para a solução de diferentes proble-

mas de engenharia, ainda existem poucos trabalhos tratando da análise de estruturas de material compósito utilizando a abordagem isogeométrica^{16,17}.

1.1 Objetivos e Contribuições

O presente trabalho tem como objetivo principal o desenvolvimento de métodos computacionais para a otimização e análise isogeométrica de estruturas laminadas.

Assim, neste trabalho foi desenvolvido um novo método para otimização de compósitos laminados baseado na combinação das heurísticas de Otimização por Nuvem de Partículas (*Particle Swarm Optimization* - PSO) e Algoritmos Genéticos (AG). Diversas variantes deste método foram implementados em um sistema de otimização por algoritmos bio-inspirados desenvolvido utilizando os recursos de Programação Orientado a Objetos¹⁸.

Uma das dificuldades de comparar algoritmos de otimização é a grande influência que os parâmetros numéricos utilizados tem sobre comportamento destes algoritmos. A fim de permitir uma comparação isenta, os parâmetros numéricos utilizados por cada variante foram determinados por um procedimento de meta-otimização¹⁹. Após a calibração dos parâmetros, as diferentes variantes foram aplicadas na otimização de laminados, com objetivo de determinar as mais adequadas à solução destes problemas. De forma geral, o método proposto apresentou excelentes resultados, mas algumas variantes tem desempenho superior às outras.

Com relação à análise de estruturas, inicialmente foi desenvolvida uma formulação isogeométrica para análise não linear geométrica de sólidos baseadas na abordagem Lagrangeana Total. Esta formulação foi depois estendida para a análise de sólidos laminados, permitindo a análise de cascas laminadas. A formulação isogeométrica foi implementada em programa orientado a objetos para análise de estruturas pelo Método dos Elementos Finitos^{18,20}. Diversas estruturas de material isotrópico e compósito foram analisadas e excelentes resultados foram obtidos em todos os casos. Finalmente, a análise isogeométrica foi utilizada na otimização de uma casca cilíndrica laminada.

1.2 Organização da Dissertação

A presente dissertação está dividida em 7 capítulos. O Capítulo 2 apresenta os conceitos básicos de compósitos laminados, incluindo os principais aspectos utilizados na formulação dos elementos sólidos isogeométricos laminados e na otimização de placas laminadas.

O Capítulo 3 apresenta uma introdução à Modelagem Geométrica, onde são abordadas as representações de Bézier, B-Spline e NURBS, mostrando sua importância para Modelagem Geométrica e abordando os conceitos e algoritmos necessários para formulação da análise isogeométrica.

O Capítulo 4 aborda a formulação isogeométrica desenvolvida neste trabalho, com foco na análise não linear geométrica de sólidos laminados, utilizando abordagem Lagrangeana

Total. O conceito de elemento isogeométrico é apresentado e os refinamentos utilizados na AIG são discutidos. Finalmente, apresenta-se a extensão da formulação para sólidos laminados.

O Capítulo 5 mostra toda a implementação computacional realizada para incorporar a AIG no programa FAST, mostrando diagramas UML para ilustrar melhor as modificações realizadas. Posteriormente, são mostrados exemplos numéricos de análise lineares e não lineares geométrica utilizando a AIG em estruturas com material isotrópico e compósito laminado.

O Capítulo 6 trata dos métodos de otimização utilizados neste trabalho. Uma revisão de literatura é realizada sobre Algoritmos Genéticos e Otimização por Nuvem de Partículas, bem como sobre o procedimento de meta-otimização. O algoritmo híbrido proposto neste trabalho e suas variantes são descritos de forma detalhada. Em seguida são apresentados os resultados da determinação dos parâmetros via meta-otimização e os resultados obtidos na solução de diversos problemas de otimização de laminados.

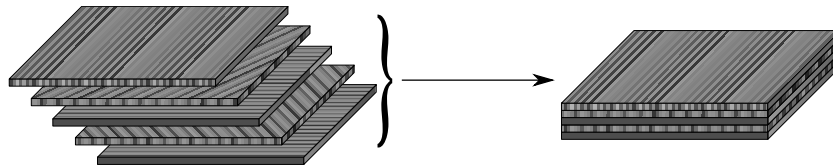
O Capítulo 7 mostra as conclusões e comentários finais do trabalho, bem como as sugestões para trabalhos futuros.

2 COMPÓSITOS LAMINADOS

O objetivo deste capítulo é discutir alguns conceitos que são importantes para este trabalho. Um estudo mais aprofundado sobre materiais compósitos pode ser encontrado na literatura²¹⁻²³.

Os materiais compósitos são materiais formados pela combinação em escala macroscópica de dois ou mais materiais de modo que juntos apresentem melhores propriedades do que seus constituintes isolados²⁴. Os compósitos reforçados por fibras são materiais formados por um conjunto de fibras de alta resistência (e.g. fibras de carbono ou vidro) envolvidas por uma matriz polimérica (e.g. resina epoxi). Várias camadas de compósitos com diferentes orientações de fibras são empilhadas de forma a se obter características estruturais mais eficientes, resultando em uma estrutura laminada, como apresentado na Figura 1.

Figura 1 – Compósito laminado.



Fonte: Elaborada pelo autor.

Os compósitos reforçados por fibras apresentam boa relação de resistência/peso e rigidez/peso e também apresentam boas propriedades como isolamento térmico, amortecimento estrutural e resistência à fadiga²². Possuem aplicações importantes na Engenharia Aeroespacial, Engenharia Naval, Engenharia Mecânica e Engenharia Civil.

Neste trabalho serão considerados compósitos produzidos com fibras unidirecionais, que são os mais utilizados em aplicações que requerem alto desempenho estrutural. Como as fibras estão alinhadas em uma direção, estes compósitos são tratados em nível macroscópico como materiais homogêneos e ortotrópicos.

O comportamento mecânico da estrutura laminada é obtida analisando o comportamento de cada lâmina, tal comportamento será abordado na próxima seção.

2.1 Mecânica de uma lâmina

O comportamento mecânico de uma lâmina pode ser previsto de acordo com duas abordagens distintas. A *micromecânica* procura analisar os materiais de fibras e matriz isoladamente, considerando a interação entre eles. Outra abordagem é a *macromecânica*, nesta o material de cada lâmina é analisado de forma conjunta, considerando que o conjunto fibra e matriz da lâmina constitua um material compósito e homogêneo. Neste trabalho foi considerado apenas a abordagem *macromecânica*.

No caso de laminados com fibras unidirecionais, cada lâmina apresenta um comportamento ortotrópico no sistema de coordenadas do material (x_1, x_2, x_3) , onde x_1 é a direção das fibras, x_2 é a direção perpendicular às fibras no plano da lâmina e x_3 é a direção perpendicular à lâmina. É importante notar que no contexto de análise de estruturas laminadas existirão três sistemas de coordenadas distintos, o sistema global da estrutura (x, y, z) , o sistema local do laminado $(\hat{x}, \hat{y}, \hat{z})$ e o sistema local de cada lâmina (x_1, x_2, x_3) , também conhecido como sistema do material. Tais sistemas estão apresentados na Figura 2.

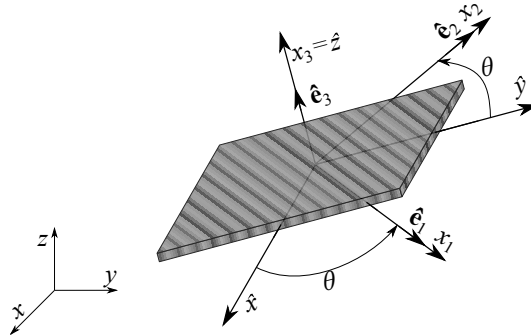


Figura 2 – Sistemas de coordenadas global (x, y, z) , sistema global do laminado $(\hat{x}, \hat{y}, \hat{z})$, e o sistema local da lâmina (x_1, x_2, x_3) , assim como seus vetores unitários $(\hat{e}_1, \hat{e}_2, \hat{e}_3)$.

O comportamento mecânico de materiais compósitos antes da falha pode ser representado de forma adequada utilizando a Lei de Hooke generalizada²⁴:

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \tau_{12} \\ \tau_{13} \\ \tau_{23} \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} & 0 & 0 & 0 \\ Q_{12} & Q_{22} & Q_{23} & 0 & 0 & 0 \\ Q_{13} & Q_{23} & Q_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & Q_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & Q_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & Q_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{bmatrix} \Rightarrow \sigma_1 = \mathbf{Q}\varepsilon_1 \quad (1)$$

onde σ_1 são as tensões na lâmina, \mathbf{Q} é a matriz constitutiva da lâmina no sistema local, e ε_1 são as deformações na lâmina. Os coeficientes da matriz \mathbf{Q} são definidos a partir das propriedades mecânicas do compósito:

$$\begin{aligned} Q_{11} &= E_1 \frac{1 - \nu_{23}\nu_{32}}{\Lambda}; & Q_{12} &= E_1 \frac{\nu_{21} + \nu_{31}\nu_{32}}{\Lambda}; & Q_{13} &= E_3 \frac{\nu_{31} + \nu_{21}\nu_{32}}{\Lambda} \\ Q_{22} &= E_2 \frac{1 - \nu_{13}\nu_{31}}{\Lambda}; & Q_{23} &= E_2 \frac{\nu_{32} + \nu_{12}\nu_{31}}{\Lambda}; & Q_{33} &= E_3 \frac{1 - \nu_{12}\nu_{21}}{\Lambda} \end{aligned} \quad (2)$$

$$Q_{66} = G_{12}; \quad Q_{44} = G_{13}; \quad Q_{66} = G_{23}$$

$$\Lambda = 1 - \nu_{12}\nu_{21} - \nu_{23}\nu_{32} - \nu_{31}\nu_{13} - 2\nu_{21}\nu_{32}\nu_{13}$$

onde E_1 , E_2 e E_3 são os módulos de elasticidade nas direções principais, ν_{12} , ν_{21} , ν_{13} , ν_{31} , ν_{23} e ν_{32} são os coeficientes de Poisson e G_{12} , G_{13} e G_{23} são os módulos de elasticidade ao

cisalhamento. É importante notar que materiais elásticos ortotrópicos possuem apenas nove constantes independentes, pois os coeficientes de Poisson (ν_{ij}) devem satisfazer à relação:

$$\frac{\nu_{ij}}{E_i} = \frac{\nu_{ji}}{E_j} \quad (i, j = 1, 2, 3 \quad \text{e} \quad i \neq j) \quad (3)$$

uma vez que a matriz \mathbf{Q} é simétrica.

Um aspecto importante na análise de estruturas laminadas é que as relações cinemáticas e as equações de equilíbrio são escritas no sistema global da estrutura (x, y, z), enquanto que as relações constitutivas são definidas no sistema lâmina (x_1, x_2, x_3). Assim, é necessário transformar as tensões e deformações entre os 2 sistemas. A transformação das deformações pode ser escrita de forma matricial¹³ como:

$$\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \gamma_{12} \\ \gamma_{13} \\ \gamma_{23} \end{bmatrix} = \begin{bmatrix} l_1^2 & m_1^2 & n_1^2 & l_1 m_1 & n_1 l_1 & m_1 n_1 \\ l_2^2 & m_2^2 & n_2^2 & l_2 m_2 & n_2 l_2 & m_2 n_2 \\ l_3^2 & m_3^2 & n_3^2 & l_3 m_3 & n_3 l_3 & m_3 n_3 \\ 2l_1 l_2 & 2m_1 m_2 & 2n_1 n_2 & l_1 m_2 + l_2 m_1 & n_1 l_2 + n_2 l_1 & m_1 n_2 + m_2 n_1 \\ 2l_3 l_1 & 2m_3 m_1 & 2n_3 n_1 & l_3 m_1 + l_1 m_3 & n_3 l_1 + n_1 l_3 & m_3 n_1 + m_1 n_3 \\ 2l_2 l_3 & 2m_2 m_3 & 2n_2 n_3 & l_2 m_3 + l_3 m_2 & n_2 l_3 + n_3 l_2 & m_2 n_3 + m_3 n_2 \end{bmatrix} \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{xz} \\ \gamma_{yz} \end{bmatrix} \quad (4)$$

onde l , m e n são os cossenos diretores dos vetores unitários locais ($\hat{e}_1, \hat{e}_2, \hat{e}_3$) em relação ao sistema global:

$$\begin{aligned} l_1 &= \cos(x, \hat{e}_1); & m_1 &= \cos(y, \hat{e}_1); & n_1 &= \cos(z, \hat{e}_1) \\ l_2 &= \cos(x, \hat{e}_2); & m_2 &= \cos(y, \hat{e}_2); & n_2 &= \cos(z, \hat{e}_2) \\ l_3 &= \cos(x, \hat{e}_3); & m_3 &= \cos(y, \hat{e}_3); & n_3 &= \cos(z, \hat{e}_3) \end{aligned} \quad (5)$$

A transformação global-local pode ser escrita de forma compacta como:

$$\boldsymbol{\varepsilon}_1 = \mathbf{T} \boldsymbol{\varepsilon} \quad (6)$$

onde \mathbf{T} é a matriz de transformação definida na Equação (4) e $\boldsymbol{\varepsilon}$ é o vetor de deformações no sistema global. Utilizando o Princípio dos Trabalhos Virtuais, pode-se mostrar¹³ que:

$$\boldsymbol{\sigma} = \mathbf{T}^T \boldsymbol{\sigma}_1 \quad (7)$$

onde $\boldsymbol{\sigma}$ é o vetor de tensões no sistema global. Finalmente, a relação constitutiva no sistema global do laminado pode então ser encontrada utilizando as Equações (6) e (7):

$$\boldsymbol{\sigma} = \mathbf{C} \boldsymbol{\varepsilon} \quad \Rightarrow \quad \boldsymbol{\sigma} = \mathbf{T}^T \mathbf{Q} \mathbf{T} \boldsymbol{\varepsilon} \quad (8)$$

onde \mathbf{C} é a matriz constitutiva no sistema global.

2.1.1 Critérios de Falha

Os critérios de falha buscam avaliar se houve falha de alguma lâmina, em função do estado de deformação ou tensão desta. Existem muitos critérios de falha na literatura, como o Critério de Máxima Tensão. Neste trabalho serão utilizados o Critério de Máxima Deformação e o Critério de Tsai-Wu²⁵.

O critério de máxima tensão avalia se alguma tensão principal excedeu o limite de resistência do material. As tensões normais últimas ($\sigma_1^u, \sigma_2^u, \sigma_3^u$) são dadas em função das resistências à tração (sub-escrito t) e compressão (sub-escrito c):

$$\sigma_1 = \begin{cases} \sigma_{1,t}^u & \text{para } \sigma_1 > 0 \\ \sigma_{1,c}^u & \text{para } \sigma_1 < 0 \end{cases} \quad \sigma_2 = \begin{cases} \sigma_{2,t}^u & \text{para } \sigma_2 > 0 \\ \sigma_{2,c}^u & \text{para } \sigma_2 < 0 \end{cases} \quad \sigma_3 = \begin{cases} \sigma_{3,t}^u & \text{para } \sigma_3 > 0 \\ \sigma_{3,c}^u & \text{para } \sigma_3 < 0 \end{cases} \quad (9)$$

$$|\tau_{12}| = |\tau_{12}^u| \quad |\tau_{13}| = |\tau_{13}^u| \quad |\tau_{23}| = |\tau_{23}^u|$$

onde os termos σ_i^u e τ_{ij}^u são respectivamente as tensões últimas normais e as tensões de últimas de cisalhamento do material compósito.

O fator de segurança S_f avalia a relação entre a tensão última e a tensão de atuante no material:

$$S_f = \frac{\sigma_i^u}{\sigma_i} \quad (10)$$

A falha ocorre quando $S_f \geq 1$. No caso do Critério de Máxima Tensão, o fator de segurança é definido como sendo a menor razão calculada considerando todas as tensões principais.

No Critério de Máxima Deformação são considerados os limites de deformação presentes no material de cada lâmina. As deformações são avaliadas nas direções principais da lâmina, como apresentado por:

$$\epsilon_1 = \begin{cases} \epsilon_{1,t}^u & \text{para } \epsilon_1 > 0 \\ \epsilon_{1,c}^u & \text{para } \epsilon_1 < 0 \end{cases} \quad \epsilon_2 = \begin{cases} \epsilon_{2,t}^u & \text{para } \epsilon_2 > 0 \\ \epsilon_{2,c}^u & \text{para } \epsilon_2 < 0 \end{cases} \quad \epsilon_3 = \begin{cases} \epsilon_{3,t}^u & \text{para } \epsilon_3 > 0 \\ \epsilon_{3,c}^u & \text{para } \epsilon_3 < 0 \end{cases} \quad (11)$$

$$|\gamma_{12}| = |\gamma_{12}^u| \quad |\gamma_{13}| = |\gamma_{13}^u| \quad |\gamma_{23}| = |\gamma_{23}^u|$$

onde os termos ϵ_i^u e γ_{ij}^u são respectivamente as deformações últimas normais e as deformações de últimas de cisalhamento do material compósito. O fator de segurança é avaliado considerando a menor razão entre as deformação última e atuante em todas as direções:

$$S_f = \frac{\epsilon_i^u}{\epsilon_i} \quad (12)$$

O Critério de Tsai-Wu²² é derivado do critério de von Mises, sendo aplicado para materiais ortotrópicos. O estado de tensão da lâmina é avaliado de modo que a tensão em todas

as direções do compósito contribuem interativamente para verificação da falha. O critério de falha de Tsai-Wu é dado por²⁴:

$$\sum_{i=1}^6 F_i \sigma_i + \sum_{i=1}^6 \sum_{j=1}^6 F_{ij} \sigma_i \sigma_j \geq 1 \quad (13)$$

onde σ_i são as tensões locais das lâminas (σ_1). Os termos F_i e F_{ij} são obtidos por:

$$\begin{aligned} F_1 &= \frac{1}{\sigma_{1,t}^u} - \frac{1}{\sigma_{1,c}^u}, & F_2 &= \frac{1}{\sigma_{2,t}^u} - \frac{1}{\sigma_{2,c}^u}, & F_3 &= \frac{1}{\sigma_{3,t}^u} - \frac{1}{\sigma_{3,c}^u}, \\ F_{11} &= \frac{1}{\sigma_{1,t}^u \sigma_{1,c}^u}, & F_{22} &= \frac{1}{\sigma_{2,t}^u \sigma_{2,c}^u}, & F_{33} &= \frac{1}{\sigma_{3,t}^u \sigma_{3,c}^u}, \\ F_{44} &= \frac{1}{(\tau_{23}^u)^2}, & F_{55} &= \frac{1}{(\tau_{13}^u)^2}, & F_{66} &= \frac{1}{(\tau_{12}^u)^2} \end{aligned} \quad (14)$$

os termos F_{12} , F_{13} e F_{23} são obtidos em função de ensaios experimentais complicados, porém tais termos podem ser avaliados de forma aproximada utilizando as resistências longitudinais por²²:

$$F_{12} = -\frac{1}{2} \frac{1}{\sqrt{\sigma_{1,t}^u \sigma_{1,c}^u \sigma_{2,t}^u \sigma_{2,c}^u}}, \quad F_{13} = -\frac{1}{2} \frac{1}{\sqrt{\sigma_{1,t}^u \sigma_{1,c}^u \sigma_{3,t}^u \sigma_{3,c}^u}}, \quad F_{23} = -\frac{1}{2} \frac{1}{\sqrt{\sigma_{2,t}^u \sigma_{2,c}^u \sigma_{3,t}^u \sigma_{3,c}^u}} \quad (15)$$

Os outros termos F_{ij} que não estão presentes nas Equações (14) e (15) são nulos.

O fator de segurança relacionado a este método é obtido utilizando a Equação (13). Considerando um fator de segurança S_f tal que multiplique as tensões σ de forma que o ponto material esteja no limiar da falha, temos que:

$$\sum_{i=1}^6 F_i (\sigma_i S_f) + \sum_{i=1}^6 \sum_{j=1}^6 F_{ij} (\sigma_i S_f) (\sigma_j S_f) = 1 \Rightarrow S_f \sum_{i=1}^6 F_i \sigma_i + S_f^2 \sum_{i=1}^6 \sum_{j=1}^6 F_{ij} \sigma_i \sigma_j = 1$$

O fator de segurança é obtido pelo cálculo da raiz positiva da expressão quadrática:

$$S_f^2 \underbrace{\sum_{i=1}^6 \sum_{j=1}^6 F_{ij} \sigma_i \sigma_j}_a + S_f \underbrace{\sum_{i=1}^6 F_i \sigma_i}_b - 1 = 0 \quad (16)$$

$$S_f = \frac{-b \pm \sqrt{b^2 + 4a}}{2a} \quad (17)$$

2.2 Teorias de Laminação

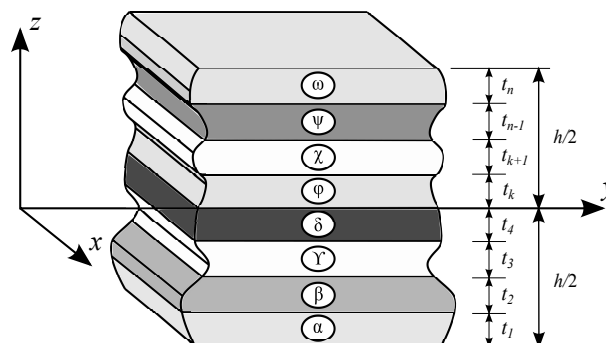
Existem diversos modelos que combinam os efeitos de cada lâmina a fim de descrever o comportamento mecânico do laminado como um todo. Estas teorias podem ser divididas em dois grupos, as teorias de camadas equivalentes (*equivalent single layer*) e as teorias de lâminas discretas (*layerwise*)^{26,27}.

Nas teorias de camadas equivalentes, todas as camadas do laminado são tratadas como uma lâmina equivalente, onde cada lâmina contribui para as propriedades mecânicas da camada resultante. Hipóteses cinemáticas são consideradas de forma a simplificar o campo de deslocamentos e de deformações no interior do laminado, resultando em uma análise estrutural com custo computacional reduzido. Um exemplo de teoria de camada equivalente é a Teoria Clássica da Laminação, que utiliza as hipóteses cinemáticas da teoria de placa de Kirchhoff, transformando o problema tridimensional em um problema bidimensional. A desvantagem destas teorias reside nos erros ocasionados pelas hipóteses cinemáticas consideradas.

Nas teorias de lâminas discretas, cada lâmina é tratada separadamente. Estas teorias conseguem captar de forma mais precisa o campo de deslocamentos e de deformações no laminado. Também são capazes de captar fenômenos como a delaminação. A desvantagem deste tipo de abordagem está no elevado custo computacional.

Os esquemas de laminação utilizados neste trabalho são representados por uma sequência dos valores de orientação das fibras $[\alpha/\beta/\gamma/\dots/\omega]$, enumerados de baixo para cima, como ilustrado na Figura 3. Estes valores devem estar definidos no intervalo $[-90^\circ, +90^\circ]$.

Figura 3 – Esquema geral de laminação.



Fonte: Rocha (2013).

Os esquemas de laminação podem ser classificados de acordo com a orientação das fibras em *cross-ply* e *angly-ply*. Os laminados *cross-ply* possuem apenas lâminas com valor de 0° e 90° graus, por exemplo o laminado $[0^\circ/90^\circ/0^\circ/90^\circ]$ é *cross-ply*. Os laminados *angly-ply* são laminados que possuem qualquer valor de ângulo, como o laminado $[-45^\circ/0^\circ/45^\circ/90^\circ]$. Os esquemas de laminação também podem ser classificados quanto à simetria em:

- Laminados Simétricos: São laminados que possuem simetria de espessura, material e orientação das fibras em relação à superfície média, como $[90^\circ/0^\circ/0^\circ/90^\circ]$.

A letra s pode ser utilizada para indicar a simetria, por exemplo um laminado simétrico $[45^\circ/-45^\circ/0^\circ/0^\circ/-45^\circ/45^\circ]$ pode ser representado por $[45^\circ/-45^\circ/0^\circ]_s$. Em caso de esquemas de laminação com um número impar de lâminas, a última camada é representada por um traço, como no caso do laminado $[45^\circ/90^\circ/0^\circ/90^\circ/45^\circ]$ que pode ser representado por $[45^\circ/90^\circ/\bar{0}^\circ]_s$, em que a superfície média divide a camada do meio em duas.

- b) Laminados Balanceados: São laminados que para cada lâmina com orientação α diferente de 0° e 90° , exista outra camada com orientação $-\alpha$, de mesma espessura e material constituinte, podendo ou não ser adjacente. O laminado $[-45^\circ/90^\circ/+45^\circ/0^\circ]$ é balanceado.
- c) Laminados Balanceados e Simétricos: São laminados que apresentam simetria e camadas com ângulos invertidos, seguindo o que foi descrito nos itens a) e b). O laminado $[90^\circ/-45^\circ/0^\circ/45^\circ]_s$ é balanceado e simétrico.
- d) Laminados Anti-simétricos: São laminados que possuem simetria de espessura, material mas anti-simetria em relação as orientações das fibras, em relação à superfície média. Por exemplo o laminado $[45^\circ/-45^\circ]_5$, que é equivalente a $[45^\circ/-45^\circ/45^\circ/-45^\circ/45^\circ/-45^\circ/45^\circ/-45^\circ/45^\circ/-45^\circ]$, é anti-simétrico.
- e) Laminados Assimétricos: Representam o caso geral de laminados, em que não existe simetria em relação a superfície média considerando espessura, material e orientação das fibras.

2.2.1 Teoria Clássica da Laminação

A Teoria Clássica da Laminação (TCL) utiliza as premissas da Teoria de Placa de Kirchhoff. Nesta teoria, as deformações ε_z , γ_{xz} e γ_{yz} são desprezadas, restringindo sua aplicação à placas finas. Utilizando estas considerações, a relação tensão-deformação passa a ser:

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{bmatrix} = \begin{bmatrix} Q_{11} & Q_{12} & 0 \\ Q_{12} & Q_{22} & 0 \\ 0 & 0 & Q_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{bmatrix} \Rightarrow \boldsymbol{\sigma}_1 = \mathbf{Q}\boldsymbol{\varepsilon}_1 \quad (18)$$

onde os termos \mathbf{Q} são dados por:

$$Q_{11} = \frac{E_1}{1 - \nu_{12}\nu_{21}}; \quad Q_{12} = \frac{\nu_{12}E_2}{1 - \nu_{12}\nu_{21}}; \quad Q_{22} = \frac{E_2}{1 - \nu_{12}\nu_{21}}; \quad Q_{66} = G_{12} \quad (19)$$

Uma transformação de eixos é realizada de forma análoga ao que foi mostrado nas Equações (7) e (8). A matriz de transformação \mathbf{T} considerando uma lâmina com orientação θ

em relações aos eixos globais:

$$\mathbf{T} = \begin{bmatrix} \cos^2\theta & \sin^2\theta & \sin\theta \cos\theta \\ \sin^2\theta & \cos^2\theta & -\sin\theta \cos\theta \\ -2\sin\theta \cos\theta & 2\sin\theta \cos\theta & \cos^2\theta - \sin^2\theta \end{bmatrix} \quad (20)$$

A matriz constitutiva de cada lâmina no sistema global é obtida aplicando a Equação (8):

$$\boldsymbol{\sigma} = \bar{\mathbf{Q}}\boldsymbol{\varepsilon} \Rightarrow \bar{\mathbf{Q}} = \mathbf{T}^T \mathbf{Q} \mathbf{T} \quad (21)$$

onde os coeficientes da matriz constitutiva transformada $\bar{\mathbf{Q}}$ são dados por:

$$\begin{aligned} \bar{Q}_{11} &= Q_{11} \cos^4\theta + (Q_{12} + 2Q_{66}) 2\sin^2\theta \cos^2\theta + Q_{22} \sin^4\theta \\ \bar{Q}_{12} &= (Q_{11} + Q_{22} - 4Q_{66}) \sin^2\theta \cos^2\theta + Q_{12} (\sin^4\theta + \cos^4\theta) \\ \bar{Q}_{16} &= (Q_{11} - Q_{12} - 2Q_{66}) \sin\theta \cos^3\theta + (Q_{12} - Q_{22} + 2Q_{66}) \sin^3\theta \cos\theta \\ \bar{Q}_{22} &= Q_{11} \sin^4\theta + 2(Q_{12} + 2Q_{66}) \sin^2\theta \cos^2\theta + Q_{22} \cos^4\theta \\ \bar{Q}_{26} &= (Q_{11} - Q_{12} - 2Q_{66}) \sin^3\theta \cos\theta + (Q_{12} - Q_{22} + 2Q_{66}) \sin\theta \cos^3\theta \\ \bar{Q}_{66} &= (Q_{11} + Q_{22} - 2Q_{12} - 2Q_{66}) \sin^2\theta \cos^2\theta + Q_{66} (\sin^4\theta + \cos^4\theta) \end{aligned} \quad (22)$$

Considerando as premissas da TCL²², as deformações em planos paralelos ao da superfície média da lâmina são dados por:

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_m + z\boldsymbol{\kappa} \quad (23)$$

onde $\boldsymbol{\varepsilon}_m$ representa as deformações de membrana e $\boldsymbol{\kappa}$ representa as curvaturas dos laminados. Em caso da análise de placas e cascas, é mais conveniente trabalhar com os esforços resultantes (forças e momentos), estes esforços podem ser obtidos integrando as tensões ao longo da espessura t :

$$\mathbf{n} = \begin{bmatrix} N_x \\ N_y \\ N_{xy} \end{bmatrix} = \int_{-t/2}^{t/2} \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} dz \quad \text{e} \quad \mathbf{m} = \begin{bmatrix} M_x \\ M_y \\ M_{xy} \end{bmatrix} = \int_{-t/2}^{t/2} \begin{bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{bmatrix} z dz \quad (24)$$

Utilizando as Equações (24), (22) e (21), a relação entre os esforços internos e deformações do laminado podem ser escritas como:

$$\begin{bmatrix} \mathbf{n} \\ \mathbf{m} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \boldsymbol{\varepsilon}_m \\ \boldsymbol{\kappa} \end{bmatrix} \quad (25)$$

onde \mathbf{A} é a matriz de rigidez de membrana, \mathbf{D} é a matriz de rigidez à flexão e \mathbf{B} é a matriz

de acoplamento membrana-flexão. Os termos destas matrizes podem ser calculados pelas expressões:

$$A_{ij} = \sum_{k=1}^n \bar{Q}_{ij}^k (z_{k+1} + z_k) ; B_{ij} = \sum_{k=1}^n \frac{\bar{Q}_{ij} (z_{k+1}^2 - z_k^2)}{2} ; D_{ij} = \sum_{k=1}^n \frac{\bar{Q}_{ij} (z_{k+1}^3 - z_k^3)}{3} \quad (26)$$

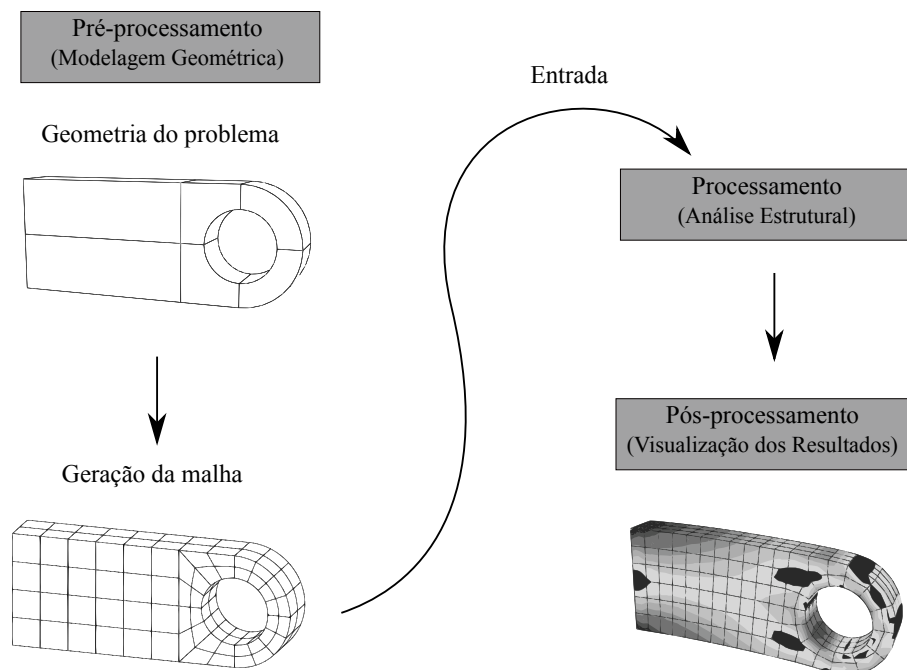
É importante notar que a matriz \mathbf{B} de laminados simétricos é nula. Adicionalmente, no caso de laminados balanceados, os termos de acoplamento de cisalhamento no plano do laminado são nulos ($A_{13} = A_{23} = A_{31} = A_{32} = 0$).

3 MODELAGEM GEOMÉTRICA

A Modelagem Geométrica é uma área da matemática aplicada que estuda descrição de formas geométricas. Possui importante aplicação em sistemas de computação gráfica, como em sistemas CAD (*Computer Aided Design*). Estes são sistemas de computador que auxiliam no desenho, modificação e otimização de projetos. São utilizados em várias áreas, como em projetos estruturais, em projetos de circuitos eletrônicos interligados, na criação de filmes de animação, dentre outras aplicações. Estes sistemas utilizam a Modelagem Geométrica como base das representações envolvidas no desenho gráfico. Tais desenhos podem ser planos ou modelos tridimensionais.

Os sistemas CAD tem grande importância para análise estrutural na atualidade. Os sistemas de análise estrutural utilizam as ferramentas CAD como ferramentas de pré-processamento, como ilustrado pela Figura 4. Além de prover um sistema interativo para modelagem da geometria do problema, também são utilizados algoritmos de geração de malha para que possa ser fornecida uma malha para um analisador estrutural. Uma correta geração da malha é fundamental para que a análise estrutural obtenha resultados corretos.

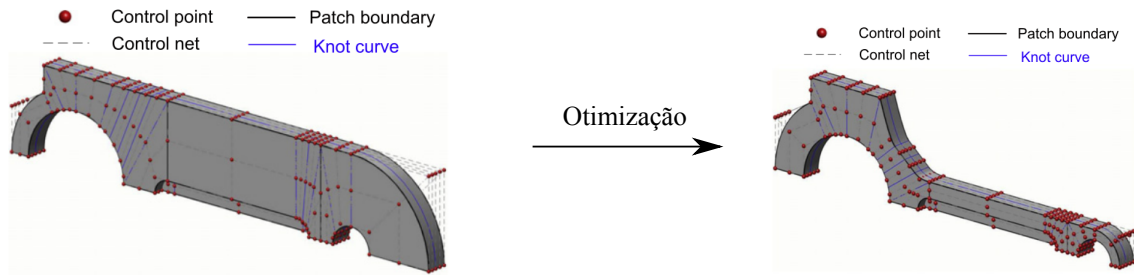
Figura 4 – Etapas de uma análise estrutural.



Fonte: Elaborada pelo autor.

Outra importante aplicação da Modelagem Geométrica é na otimização estrutural, como apresentado na Figura 5. Em tal otimização, modifica-se a geometria do modelo afim de minimizar a quantidade de material utilizada para concepção da peça estrutural, respeitando limites de desempenho mecânico.

Figura 5 – Exemplo de otimização de forma.



Fonte: Modificado de Li e Qian (2011).

Este capítulo apresenta uma revisão bibliográfica sobre os conceitos de Modelagem Geométrica utilizados neste trabalho. Uma abordagem mais abrangente do tema pode ser encontrada na literatura²⁹⁻³¹.

3.1 Representações de Curvas e Superfícies

A maneira mais comum de representação de curvas é através de equações explícitas. Um exemplo simples é o da equação de uma parábola sobre o plano xy :

$$y = ax^2 + bx + c \quad (27)$$

Alternativamente, esta curva pode ser escrita como:

$$C(x,y) = ax^2 + bx + c - y = 0 \quad (28)$$

onde $C(x,y)$ é a equação implícita da parábola. Existem expressões que não podem ser escritas de forma explícita, isto ocorre quando a variável explícita não pode ser isolada, como em:

$$C(x,y) = y \cos(x/y) - 3x^2 = 0 \quad (29)$$

A representação paramétrica é outra forma de representar uma curva. Esta representação descreve os pontos (x, y, z) de um modelo através de uma função paramétrica, como é o caso de do primeiro quadrante de uma circunferência de raio unitário centrada na origem:

$$C(t) = (x(t), y(t)) = (\cos(t), \sin(t)), \quad \text{com } 0 \leq t \leq \frac{\pi}{2} \quad (30)$$

onde t é a variável paramétrica. A equação implícita equivalente é definida pela expressão:

$$C(x,y) = (x, \sqrt{1-x^2}), \quad \text{com } 0 \leq x \leq 1 \quad (31)$$

Uma superfície pode ser descrita implicitamente por $S(x,y,z) = 0$. Por exemplo, uma esfera de raio unitário centrada na origem pode ser descrita por:

$$S(x,y,z) = x^2 + y^2 + z^2 - 1 = 0 \quad (32)$$

Descrevendo esta equação de forma paramétrica, temos:

$$S(t,u) = (x(t,u), y(t,u), z(t,u)) = (\sin t \cos u, \sin t \sin u, \cos t) \quad 0 \leq t \leq 1, \quad 0 \leq u \leq 1 \quad (33)$$

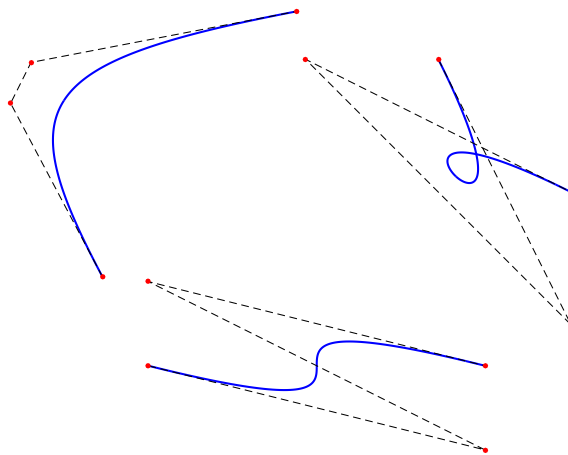
Podemos observar que é necessário usar uma variável paramétrica para descrever uma curva e 2 variáveis para descrever uma superfície. No caso de um sólido serão necessárias três variáveis.

A representação paramétrica é mais adequada para representação de formas no computador³⁰. Esta representação será utilizada nas formulações de Bézier, B-spline e NURBS apresentadas nas próximas seções.

3.2 Curvas de Bézier

A curva de Bézier foi criada pelo engenheiro francês Paul De Casteljaou, quando ainda era funcionário da Citroën, uma fabricante de automóveis. A curva foi popularizada pelo engenheiro francês Pierre Bézier, que a utilizou como uma ferramenta de projeto de fácil utilização para projetistas e desenhistas de automóveis, sem requerer um grande conhecimento matemático destes para sua utilização. As primeiras publicações sobre esta representação foram apresentadas entre o final da década de 1950 e início da década de 1960^{32,33}. Embora tenha sido criada para projeto de automóveis, existem diversas aplicações da curva de Bézier em áreas relacionadas à computação gráfica, como em programas de manipulação de imagens (GIMP³⁴ e Photoshop³⁵) e programas de desenho vetorial (Inkscape³⁶ e CorelDraw³⁷).

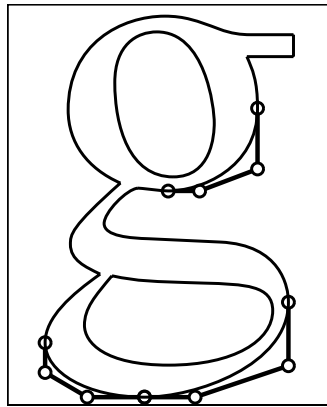
Figura 6 – Exemplo de curvas de Bézier de grau 3.



Fonte: Elaborada pelo autor.

A curva é definida em função de um conjunto de pontos discretos, denominados pontos de controle, e sua geometria pode ser facilmente alterada movendo os pontos de controle de posição. A Figura 6 mostra exemplos de curvas de Bézier cúbicas (linha espessa) com o polígono formado pelos seus pontos de controle (linha tracejada). É importante notar que apenas os pontos de controle extremos pertencem à curva. Portanto, os pontos de controle não são pontos de interpolação. Pode-se notar que a curva é tangente ao polígono nas extremidades, isto possibilita que geometrias complexas possam ser obtidas pela união de várias curvas mantendo a tangente contínua (continuidade C^1). Uma aplicação interessante é na criação de fontes de contorno como é o caso das fontes PostScript, a Figure 7 mostra o contorno da letra "g" definido pela fonte PostScript com detalhes dos pontos de controle das curvas de Bézier utilizadas.

Figura 7 – Definição de contorno de fontes utilizando representação de Bézier.



Fonte: Sederberg (2014).

Uma curva de Bézier de grau p é construída por uma combinação linear dos pontos de controle \mathbf{p}_i :

$$C(\xi) = \sum_{i=1}^{p+1} B_{i,p}(\xi) \mathbf{p}_i \quad (34)$$

onde $B_{i,p}$ são os polinômios de Bernstein de grau p e ξ é a coordenada paramétrica. Esta expressão mostra que cada curva de grau p possui $p + 1$ pontos de controle. Os polinômios de Bernstein³⁸ foram estudados por Bernstein para provar o Teorema de Stone-Weierstrass.

Considerando um intervalo paramétrico $[0, 1]$, as funções de base Bernstein podem ser calculadas pela expressão³⁹:

$$B_{i,p}(\xi) = \binom{p}{i} (1 - \xi)^{p-i} \xi^i, \quad i = 0, 1, \dots, p \quad (35)$$

$$\text{com } \binom{p}{i} = \frac{p!}{i!(p-i)!} \quad (36)$$

Outra forma de cálculo dos polinômios de Bernstein é através da fórmula recursiva³⁰:

$$B_{i,p}(\xi) = (1 - \xi)B_{i,n-1}(\xi) + \xi B_{i-1,n-1}(\xi) \quad (37)$$

$$B_{i,p}(\xi) = 0 \quad \text{para } \xi < 0 \quad \text{e } \xi > 1, \quad B_{0,0}(\xi) = 1$$

A primeira derivada pode ser calculadas por:

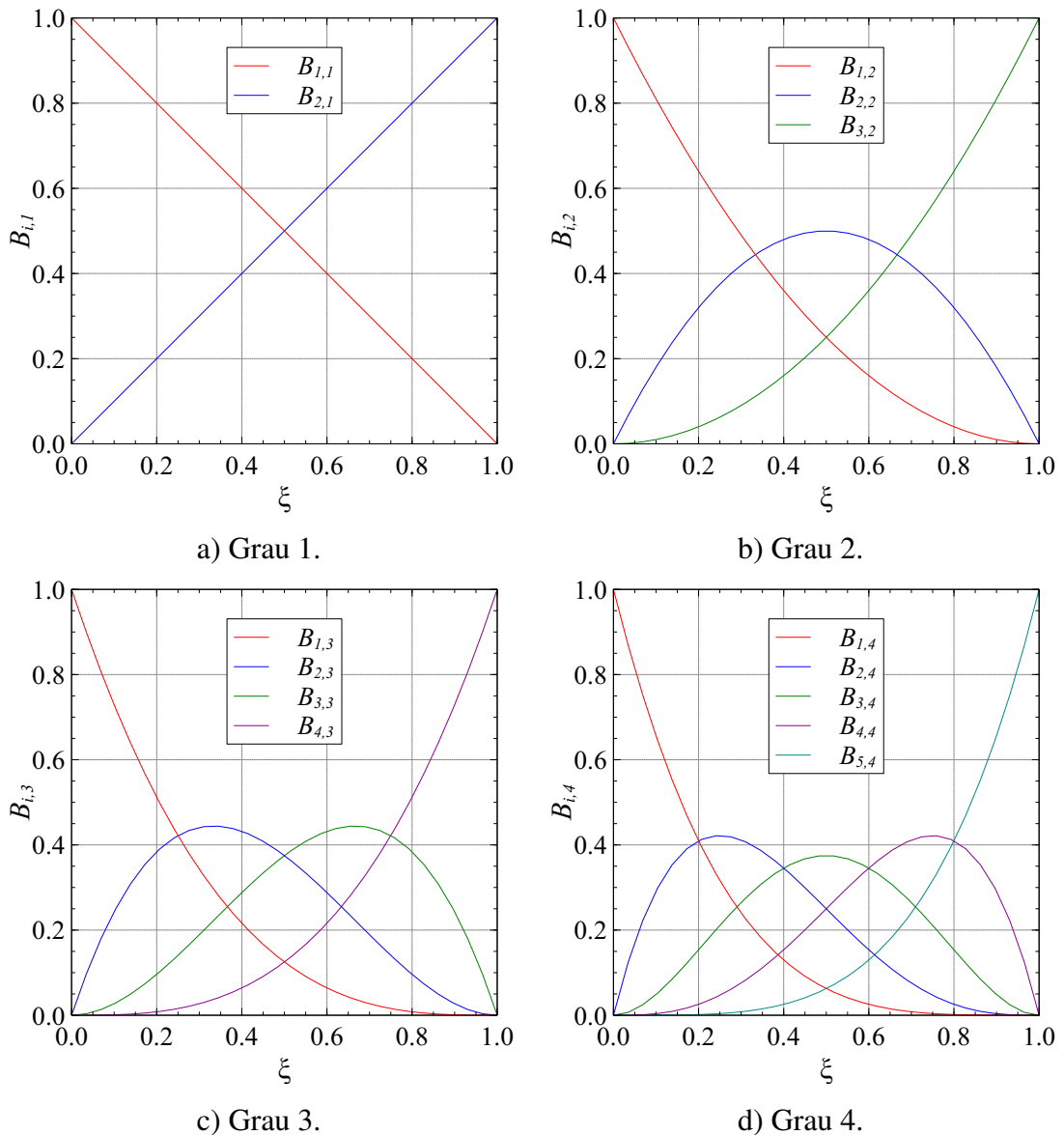
$$\frac{d}{d\xi} B_{i,p}(\xi) = p(B_{i-1,p-1}(\xi) - B_{i,p-1}(\xi)) \quad (38)$$

Os polinômios de Bernstein $B_{i,p}$ possuem as importantes propriedades:

- a) Independência linear;
- b) Não-negatividade: $B_{i,p}(\xi) \geq 0$;
- c) Partição da unidade: $\sum_{i=1}^n B_{i,p}(\xi) = 1$
- d) Simetria: $B_{i,p}(\xi) = B_{p-i,p}(1 - \xi)$;
- e) Controle pseudo-local: Se um ponto de controle de uma representação de Bézier for movido, então as mudanças da forma da representação ocorrem de forma dominante na vizinhança do ponto movido.

A Figura 8 mostra polinômios de Bernstein de diferentes graus.

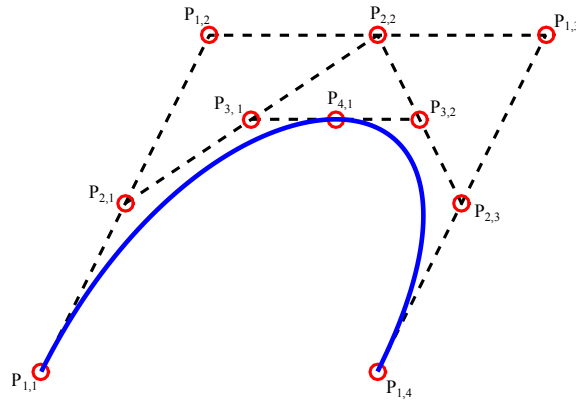
Figura 8 – Polinômios de Bernstein para vários graus.



Fonte: Elaborada pelo autor.

Uma maneira mais eficiente e estável numericamente de calcular os pontos da curva de Bézier do que pela Equação (34) é através do algoritmo de *de Casteljau*³⁰. Neste algoritmo, cada segmento do polígono de controle é dividido de acordo com o ponto paramétrico ξ calculado, sendo traçados novas retas com os pontos obtidos, reduzindo em um o número de retas. O processo é repetido até que apenas uma reta seja desenhando, e o ponto paramétrico ξ desta reta resultante é o valor da curva de Bézier. O processo é ilustrado na Figura 9. Uma abordagem mais detalhada do algoritmo pode ser encontrada em outras fontes^{30,39}.

Figura 9 – Algoritmo de *de Casteljau* aplicado a curva de Bézier cúbica.



Fonte: Elaborada pelo autor.

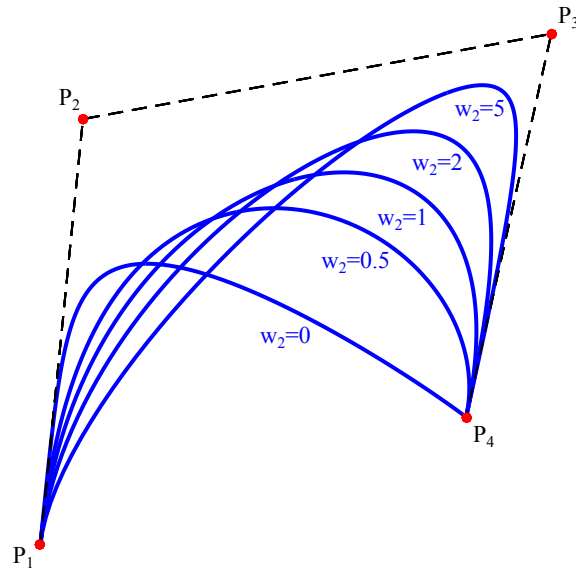
3.2.1 Bézier Racional

Uma das desvantagens da curva de Bézier apresentada anteriormente é que esta não é capaz de representar curvas cônicas, como circunferências e elipses de forma exata, pois estas curvas não são descritas por funções polinomiais. Contudo, esta representação é possível utilizando funções racionais. Para este fim são acrescentados pesos aos pontos de controle da curva. Desta forma, a curva de Bézier passa a ser calculada pela expressão:

$$C(\xi) = \frac{\sum_{i=1}^{p+1} w_i B_{i,p}(\xi) \mathbf{p}_i}{\sum_{\hat{i}=1}^{p+1} w_{\hat{i}} B_{\hat{i},p}(\xi)} \quad (39)$$

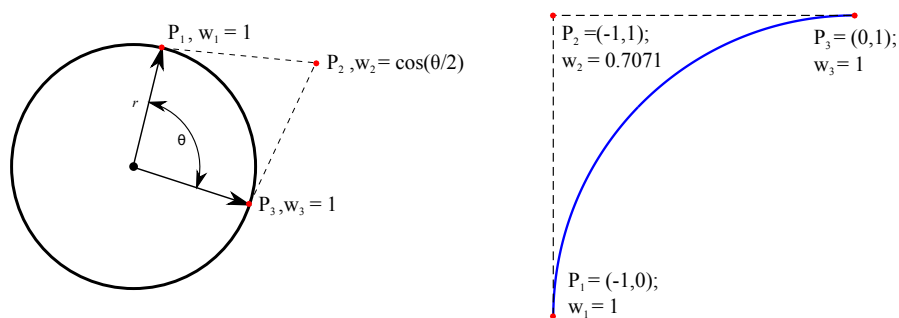
onde w_i é o peso associado a cada ponto de controle \mathbf{p}_i . O efeito da variação do peso pode ser observado na Figure 10, onde o peso do ponto \mathbf{p}_2 é modificado entre os valores de 0 e 5. Observe que caso todos os pesos sejam iguais a 1, a equação racional se torna igual a equação tradicional. Um caso particular interessante de uma curva de Bézier racional é o arco de circunferência. Este pode ser obtido utilizando os pontos de controle e pesos ilustrados na Figura 11.

Figura 10 – Efeito da variação do peso.



Fonte: Elaborada pelo autor.

Figura 11 – Representação de um arco de círculo utilizando Bézier Racional.



Fonte: Elaborada pelo autor.

3.2.2 Superfícies e Sólidos de Bézier

Uma superfície de Bézier construída por produto tensorial é obtida a partir do produto de dois polinômios de Bernstein univariantes. Considerando uma superfície de Bézier racional de grau p na direção ξ e q na direção η , esta pode ser calculada por:

$$S(\xi, \eta) = \frac{\sum_{i=1}^{p+1} \sum_{j=1}^{q+1} w_{ij} B_{i,p}(\xi) B_{j,q}(\eta) \mathbf{P}_{ij}}{\sum_{\hat{i}=1}^{p+1} \sum_{\hat{j}=1}^{q+1} w_{\hat{i}\hat{j}} B_{\hat{i},p}(\xi) B_{\hat{j},q}(\eta)} \quad (40)$$

onde $B_{i,p}(\xi)$ e $B_{j,q}(\eta)$ são os polinômios de Bernstein univariantes e \mathbf{P} é uma matriz de pontos de controle de tamanho $(q+1) \times (p+1)$. Analogamente, um sólido de Bézier racional de grau

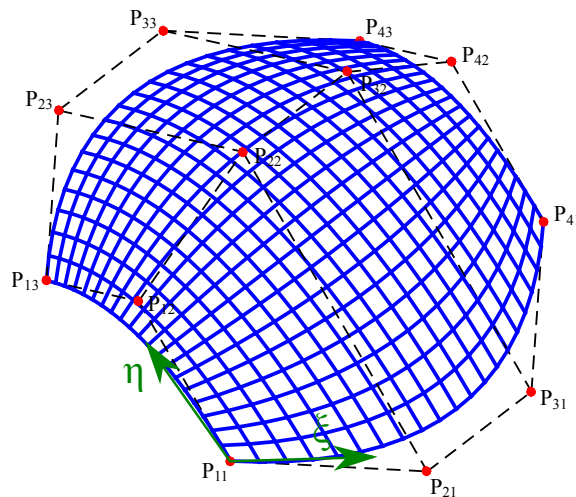
p na direção ξ , q na direção η e l na direção ζ , pode ser calculado por:

$$V(\xi, \eta, \zeta) = \frac{\sum_{i=1}^{p+1} \sum_{j=1}^{q+1} \sum_{k=1}^{l+1} w_{ijk} B_{i,p}(\xi) B_{j,q}(\eta) B_{k,l}(\zeta) \mathbf{P}_{ijk}}{\sum_{\hat{i}=1}^{p+1} \sum_{\hat{j}=1}^{q+1} \sum_{\hat{k}=1}^{l+1} w_{\hat{i}\hat{j}\hat{k}} B_{\hat{i},p}(\xi) B_{\hat{j},q}(\eta) B_{\hat{k},l}(\zeta)} \quad (41)$$

onde $B_{i,l}$, $B_{j,q}$ e $B_{k,p}$ são os polinômios de Bernstein univariantes e \mathbf{P} é um tensor de terceira ordem de pontos de controle de tamanho $(l+1) \times (q+1) \times (p+1)$. É importante notar que a Equação (40) e Equação (41) consideram o caso geral de superfícies e sólidos racionais, portanto consideram os pesos. Vale lembrar que caso todos os pesos sejam iguais a 1, resultará no caso particular das superfícies e sólidos não-rationais.

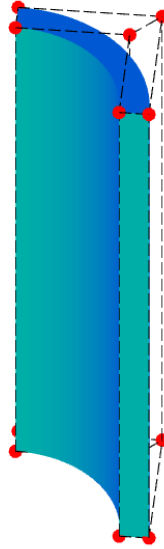
A Figura 12 mostra uma de superfície de Bézier de grau 3×2 . A Figura 13 mostra um quarto de cilindro modelado através de sólido de Bézier de grau $2 \times 1 \times 1$. Os pontos de controle do sólido de Bézier são apresentados na Tabela 1.

Figura 12 – Exemplo de superfície Bézier.



Fonte: Elaborada pelo autor.

Figura 13 – Exemplo de sólido de Bézier.



Fonte: Elaborada pelo autor.

Tabela 1 – Pontos de controle do Sólido de Bézier da Figura 13.

$\mathbf{p}_{i,j,k}$	x	y	z	w
$\mathbf{p}_{1,1,1}$	0.000	0.995	0.000	1.000
$\mathbf{p}_{2,1,1}$	0.995	0.995	0.000	0.707
$\mathbf{p}_{3,1,1}$	0.995	0.000	0.000	1.000
$\mathbf{p}_{1,2,1}$	0.000	1.255	0.000	1.000
$\mathbf{p}_{2,2,1}$	1.255	1.255	0.000	0.707
$\mathbf{p}_{3,2,1}$	1.255	0.000	0.000	1.000
$\mathbf{p}_{1,1,2}$	0.000	0.995	5.000	1.000
$\mathbf{p}_{2,1,2}$	0.995	0.995	5.000	0.707
$\mathbf{p}_{3,1,2}$	0.995	0.000	5.000	1.000
$\mathbf{p}_{1,2,2}$	0.000	1.255	5.000	1.000
$\mathbf{p}_{2,2,2}$	1.255	1.255	5.000	0.707
$\mathbf{p}_{3,2,2}$	1.255	0.000	5.000	1.000

Fonte: Elaborada pelo autor.

3.3 B-Splines

O termo Spline se refere a uma ferramenta de desenho utilizada antigamente no desenhos de peças mecânicas e projeto de embarcações⁴⁰. A ferramenta possibilitava de maneira prática o traçado de curvas suaves sobre a prancha de desenho. As B-splines são curvas capazes de descrever vários segmentos distintos ao longo da mesma representação paramétrica. Esta característica é obtida limitando a atuação das funções base em regiões do espaço paramétrico. Estas regiões são conhecidas na literatura como *knot spans* e são definidas por um

vetor de valores paramétricos, o vetor de *knots* (a tradução de *knot* é nó, termo relacionado aos laços utilizados por marinheiros, porém para não confundir com os nós de elementos finitos, será considerado o termo em inglês). As B-splines podem descrever qualquer representação de Bézier não racional. Existem várias aplicações que utilizam B-splines, como em programas de modelagem tridimensional como Rhinoceros⁴¹, programas de animação como Maya⁴² e sistemas CAD como AutoCAD⁴³.

Uma curva B-spline é construída por uma combinação linear entre os pontos de controle \mathbf{p}_i e funções de base $N_{i,p}(\xi)$, como mostra a expressão:

$$C(\xi) = \sum_{i=1}^n N_{i,p}(\xi) \mathbf{p}_i \quad (42)$$

onde n é o número de funções de base B-splines e p é o grau da curva.

As bases B-splines requerem um vetor de *knots*, que consiste num conjunto de valores paramétricos não-negativos e não-decrescentes delimitados ao longo do intervalo paramétrico $[\xi_1, \xi_{n+p+1}]$ no qual a curva foi definida. Um vetor de *knots* é dito uniforme se os valores paramétricos contidos neste variam segundo um mesmo fator. Por exemplo, os vetores $\Xi = [0, 0, 0, 0.33, 0.67, 0.67, 1, 1, 1]$ e $\Xi = [0, 0, 0, 0, 1, 2, 3, 3, 4, 4, 4, 4]$ são uniformes, já os vetores $\Xi = [0, 0, 0, 0.25, 0.75, 1, 1, 1]$ e $\Xi = [0, 0, 0, 2, 2, 3, 3, 6, 6, 6, 6]$ não são.

Considerando o vetor de *knots* $\Xi = [\xi_1, \xi_2, \dots, \xi_{n+p+1}]$, as funções de base B-spline são definidas pela fórmula recursiva de Cox-de Boor³⁰:

$$N_{i,0}(\xi) = \begin{cases} 1, & \xi_i \leq \xi < \xi_{i+1} \\ 0, & \text{caso contrário} \end{cases}$$

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \quad (43)$$

Assim, cada base $N_{i,p}$ é não nula apenas no intervalo paramétrico $[\xi_i, \xi_{i+n+1}]$.

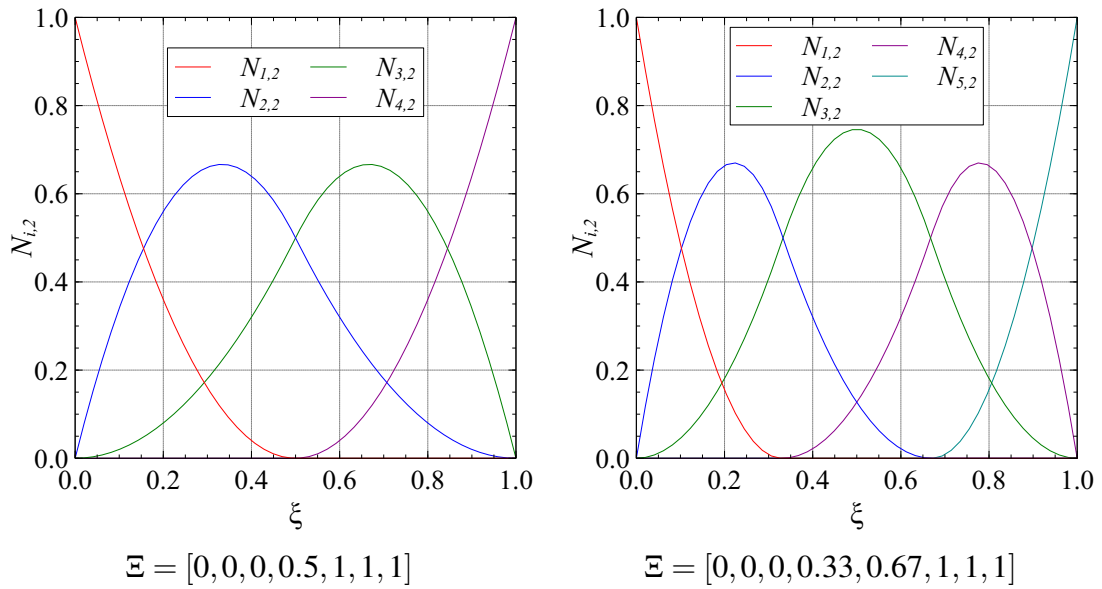
Considerando um exemplo de uma base quadrática com vetor de *knots* $\Xi = [0, 0, 0, 0.5, 1, 1, 1]$, onde as bases estão apresentadas na Figura 14, podemos observar que a função $N_{1,2}$ contribui ao longo do intervalo $[\xi_1, \xi_4] = [0, 0.5]$. A Figura 14 também mostra as bases B-spline quadráticas considerando um vetor de *knots* mais discretizado $\Xi = [0, 0, 0, 0.5, 1, 1, 1]$. Um exemplo similar mostrando as bases de uma B-spline cúbica está apresentado na Figura 15. O intervalo de contribuição de cada base mostrada nas duas figuras estão apresentados na Tabela 2.

Pode-se observar nos exemplos que o número de bases aumenta a medida que são utilizados vetores de *knots* maiores. O número de bases n pode ser calculado em função do tamanho do vetor de *knots* (ks) e do grau das bases (p):

$$n = ks - p - 1$$

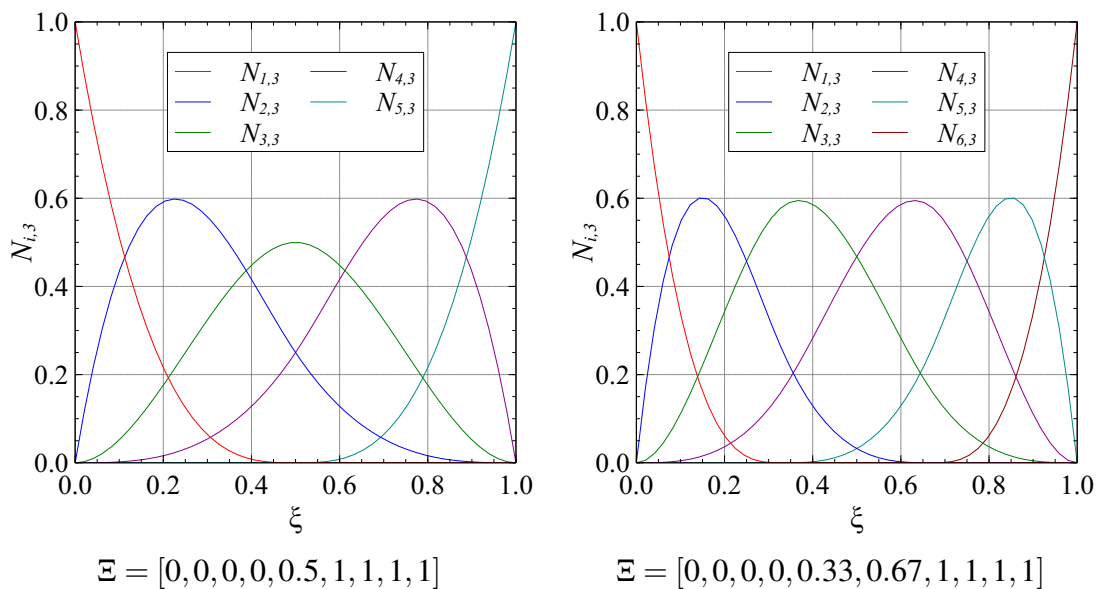
Outra observação importante é que o número de bases não nulas em cada *knot span* é sempre igual a $p+1$. Esta propriedade permite que as B-splines sejam avaliadas de forma bastante eficiente.

Figura 14 – Funções base B-spline quadráticas.



Fonte: Elaborada pelo autor.

Figura 15 – Funções de base B-spline cúbica.



Fonte: Elaborada pelo autor.

Tabela 2 – Intervalo de contribuição das bases B-spline para diferentes vetores Ξ .

B-spline quadrática			B-spline cúbica		
$N_{i,2}$	[0,0,0,0.5,1,1,1]	[0,0,0,0.33,0.67,1,1,1]	$N_{i,3}$	[0,0,0,0,0.5,1,1,1,1]	[0,0,0,0,0.33,0.67,1,1,1,1]
$N_{1,2}$	[0.0,0.5]	[0.00,0.33]	$N_{1,3}$	[0.0,0.5]	[0.00,0.33]
$N_{2,2}$	[0.0,1.0]	[0.00,0.67]	$N_{2,3}$	[0.0,1.0]	[0.00,0.67]
$N_{3,2}$	[0.0,1.0]	[0.00,1.00]	$N_{3,3}$	[0.0,1.0]	[0.00,1.00]
$N_{4,2}$	[0.5,1.0]	[0.33,1.00]	$N_{4,3}$	[0.0,1.0]	[0.00,1.00]
$N_{5,2}$		[0.67,1.00]	$N_{5,3}$	[0.5,1.0]	[0.33,1.00]
			$N_{6,3}$		[0.67,1.00]

Fonte: Elaborada pelo autor.

A primeira derivada das funções base pode ser calculada por:

$$\frac{d}{d\xi} N_{i,p}(\xi) = \frac{p}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) - \frac{p}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi) \quad (44)$$

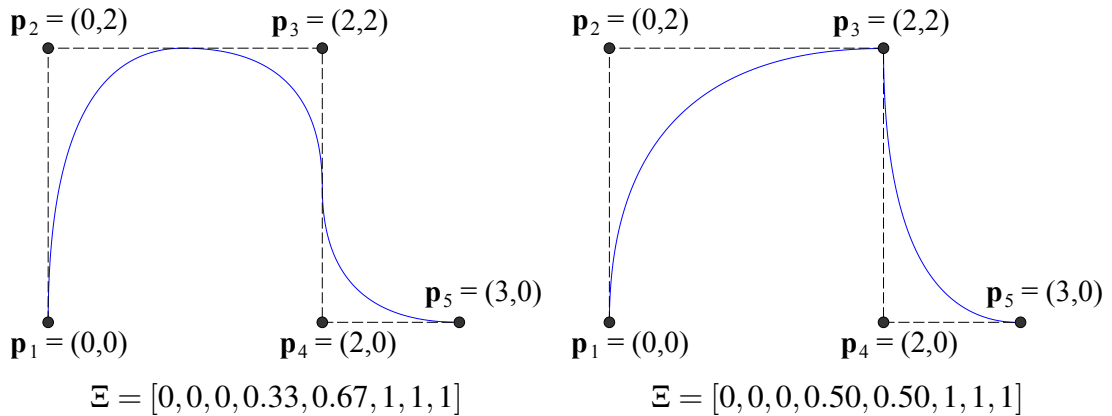
Esta derivada é essencial no contexto de Análise Isogeométrica, pois é utilizada na avaliação da matriz de rigidez \mathbf{K} . A expressão para cálculo de derivadas de ordem superior pode ser encontrada em Piegl e Tiller (1997).

Os valores paramétricos no interior dos vetores de *knots* podem aparecer repetidamente, sendo o número de repetições que um dado valor paramétrico ξ_i possui conhecido como a multiplicidade do *knot*. Uma das propriedades das B-splines é possuir continuidade C^{p-1} . Em caso de algum *knot* ter multiplicidade m , a continuidade da curva naquela coordenada paramétrica será C^{p-m} .

Uma importante observação é que caso um *knot* interno ξ_i possua multiplicidade igual ao grau da B-spline ($m = p$), então a curva interpolará um ponto de controle em ξ_i . Se os *knots* extremos tiverem esta multiplicidade $m = p + 1$, então os pontos de controle extremos serão interpolados. Por este motivo que a maioria das representações B-splines utilizadas em Análise Isogeométrica possuem multiplicidade $p + 1$ nos *knots* extremos, garantindo que os pontos de controle inicial e final sejam interpolados, como ocorre com as curvas de Bézier. Este tipo de vetor de *knots* é conhecido na literatura como *open knot vector*, ou vetor de *knots aberto*.

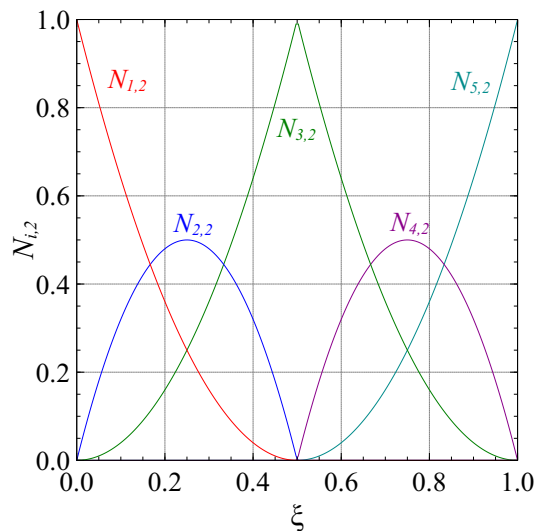
A Figura 17 mostra as funções de base quadrática considerando um *knot* com multiplicidade 2 no valor paramétrico 0.5. Podemos observar o caráter interpolador da função de base na posição 0.5 pelo fato de apenas a base $N_{3,2}$ contribuir para o cálculo da curva definida pela Equação (42), resultando em $C(0.5) = \mathbf{p}_3$. A Figura 16 mostra o efeito da consideração de um vetor de *knots* com multiplicidade $m = p$ em *knots* internos, comparando duas curvas B-splines quadráticas que possuem os mesmos pontos de controle, mas vetores de *knots* diferentes.

Figura 16 – Efeito da multiplicidade em uma B-spline quadrática.



Fonte: Elaborada pelo autor.

Figura 17 – Funções de base B-spline quadráticas com $\mathfrak{E} = [0, 0, 0, 0, 0.5, 0.5, 1, 1, 1, 1]$.



Fonte: Elaborada pelo autor.

É importante notar que as bases são definidas por partes (*piecewise function*), pois uma dada base $N_{i,p}(\xi)$ é nula fora do intervalo $[\xi_i, \xi_{i+p+1}]$, possibilitando que a influência dos pontos de controle se restrinjam a partes da curva. Matematicamente, diz-se que as bases B-spline tem suporte compacto. Esta é uma grande vantagem que a representação B-splines oferece em relação a representação de Bézier, em que todos os pontos de controle influenciam na curva.

As curvas B-splines podem representar curvas de Bézier caso seja considerado um único *knot span* e as multiplicidades dos *knots* deste sejam $p + 1$:

$$\mathfrak{E} = [\underbrace{0, \dots, 0}_{p+1}, \underbrace{1, \dots, 1}_{p+1}]$$

Uma consideração similar pode ser realizada para o caso de superfícies e sólidos³⁰.

As funções de base B-splines $N_{i,p}(\xi)$ possuem as várias propriedades importantes³⁰:

- a) Não-negatividade: $N_{i,p}(\xi) \geq 0$;
- b) Partição da unidade: $\sum_{i=1}^n N_{i,p}(\xi) = 1$
- c) Suporte compacto: $N_{i,p}(\xi) = 0$ se ξ estiver fora do intervalo $[\xi_i, \xi_{i+p+1}]$;
- d) Dado um *knot span* $[\xi_j, \xi_{j+1}]$, exatamente $p + 1$ funções de base são não nulas;
- e) Todas as derivadas de $N_{i,p}(\xi)$ existem no interior dos *knot spans*. Nos *knots* as bases são diferenciáveis $p - m$ vezes, onde m é a multiplicidade do *knot*.

3.3.1 Inserção de knot e elevação de grau

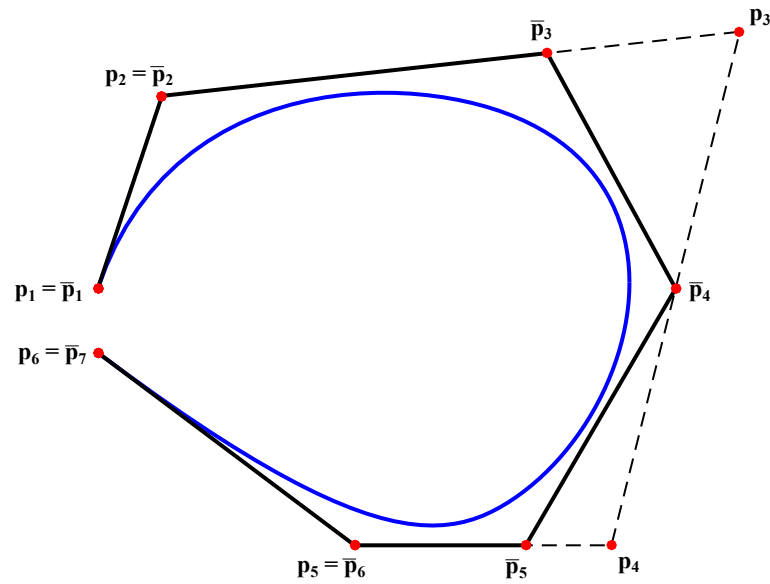
A *Inserção de Knot e Elevação do Grau* são operações realizadas em B-splines que alteram a descrição da curva sem modificar a forma desta. A *Inserção de knot* adiciona um novo valor ξ_i no vetor de *knots* Ξ , uma nova base $N_{i,p}$ e um novo ponto de controle. Para manter a curva idêntica, alguns dos pontos de controle são alterados, como ilustrado na Figura 19. Considerando um vetor de *knots* $\Xi = [\xi_1, \xi_2, \dots, \xi_{n+p+1}]$ e um valor paramétrico a ser inserido $\bar{\xi} \in [\xi_k, \xi_{k+1}]$, os novos pontos de controle $\bar{\mathbf{p}}$ podem ser obtidos em função dos pontos de controle antigos \mathbf{p} pela expressão:

$$\bar{\mathbf{p}}_i = \begin{cases} \mathbf{p}_1, & i = 1, \\ \alpha_i \mathbf{p}_i + (1 - \alpha_i) \mathbf{p}_{i-1}, & 1 < i < h, \\ \mathbf{p}_{h-1} & i = h. \end{cases} \quad (45)$$

onde h é o tamanho do novo vetor de pontos de controle $\bar{\mathbf{p}}$. O termo α é calculado por:

$$\alpha_i = \begin{cases} 1, & 1 \leq i \leq k - p, \\ \frac{\bar{\xi} - \xi_i}{\xi_{i+p} - \xi_i}, & k - p + 1 \leq i \leq k, \\ 0, & i \geq k + 1. \end{cases} \quad (46)$$

Figura 18 – Exemplo de inserção de knot em curva B-spline ($\bar{\xi} = 2$).



Vetor de *knots* antes = [0,0,0,1,3,4,4,4]

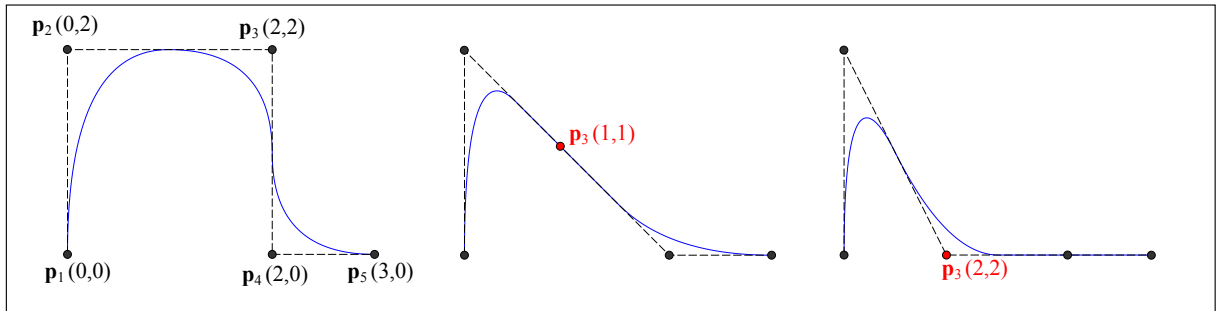
Vetor de *knots* depois = [0,0,0,1,2,3,4,4,4]

Fonte: Elaborada pelo autor.

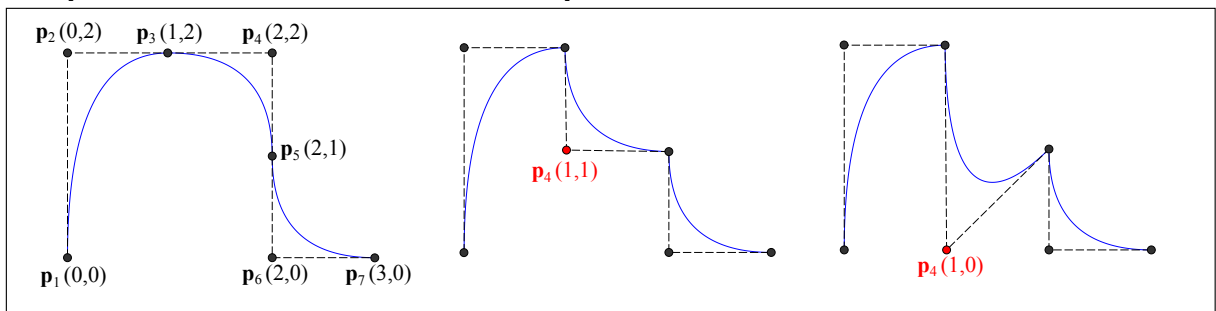
Do ponto de vista de modelagem geométrica, a *inserção de knot* permite um controle local maior da curva. Uma vez que sejam inseridos *knots* com multiplicidade $m = p$, a quebra da continuidade possibilita isolar em intervalos paramétricos as modificações provenientes das alterações de posição de alguns pontos de controle da curva, como está ilustrado na Figura 19. Do ponto de vista da Análise Isogeométrica, o *inserção de knot* possibilita discretizar o modelo inserindo um número maior de graus de liberdade.

Figura 19 – Controle local obtido pelo inserção de *knots*.

$$\Xi = [0, 0, 0, 0, 0.33, 0.67, 1, 1, 1, 1]$$



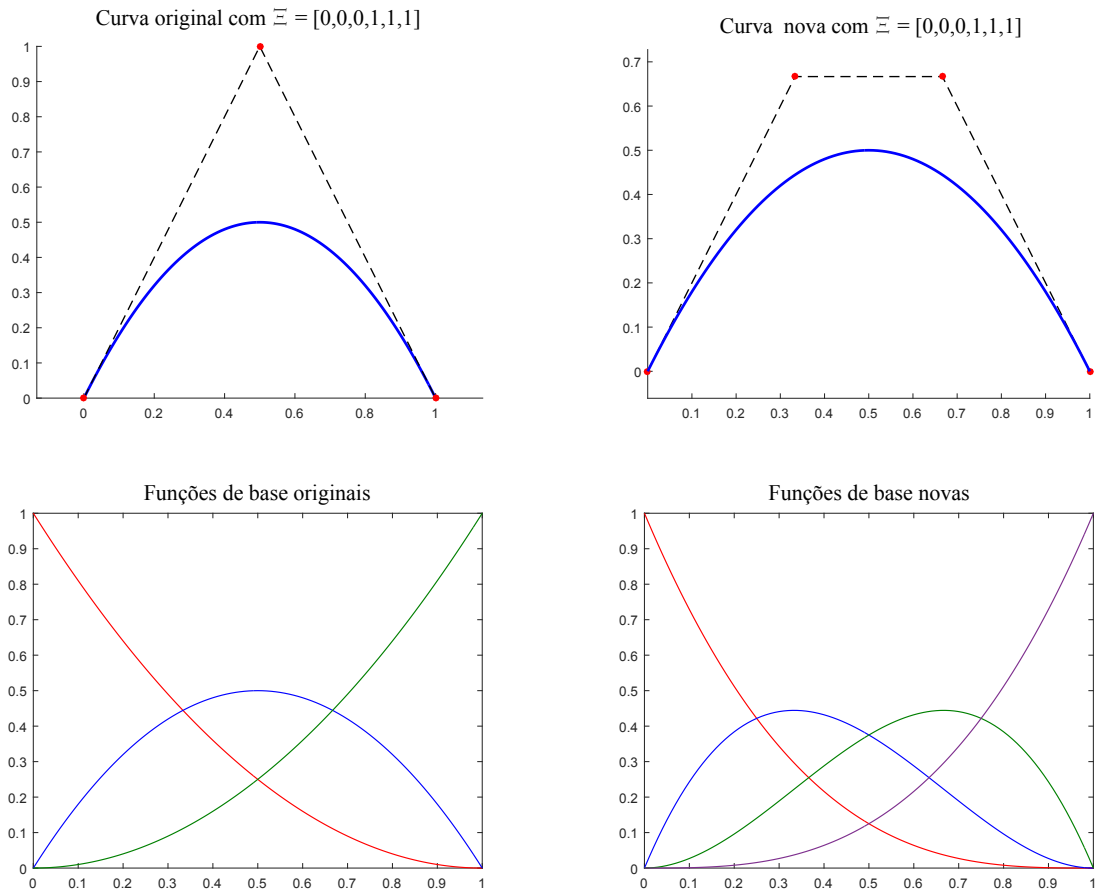
$$\Xi = [0, 0, 0, 0, 0.33, 0.33, 0.67, 0.67, 1, 1, 1, 1]$$



Fonte: Elaborada pelo autor.

A *elevação de grau* atua nas B-splines elevando o grau da curva em cada intervalo paramétrico, mantendo a continuidade original em cada *knot*. A multiplicidade de cada *knot* é aumentada em 1, preservando a continuidade original existente nos *knots*. A Figura 20 mostra o processo de elevação de grau sendo aplicado a uma curva B-spline quadrática. O algoritmo da *elevação de grau* pode ser encontrado em Piegl e Tiller (1997). Para a Análise Isogeométrica, a *elevação de grau* melhora a solução numérica uma vez que se utiliza funções de forma de ordem maior.

Figura 20 – Elevação de grau aplicada a uma curva B-spline quadrática.



Fonte: Elaborada pelo autor.

3.4 NURBS

As B-splines não são capazes de representar exatamente cônicas, como circunferências ou elipses, uma vez que estas curvas não são representadas por polinômios. Esta limitação pode ser eliminada utilizando bases racionais, o mesmo procedimento realizado para curvas de Bézier na Seção 3.2.1. As NURBS (*Non Uniform Rational B-Splines*) são B-splines racionais com vetores de *knots* não uniformes. São amplamente utilizadas na atualidade por modeladores geométricos em geral. A grande aceitação das NURBS na indústria de modelagem computacional se deve as vantagens descritas abaixo⁴⁴:

- a) Oferecem uma forma matemática capaz de representar tanto modelos matemáticos padrão (quádricas, cônicas, superfícies de revolução, ...) quanto modelos de forma livre, utilizando a mesma base de dados para armazenamento de ambos;
- b) Uma grande variedade de formas podem ser modeladas alterando os pontos de controle e pesos do modelo;
- c) Podem ser avaliadas de forma rápida e numericamente estável;
- d) Transformações afins, como translação, rotação, espelhamento e cisalhamento

podem ser aplicadas diretamente nos pontos de controle da NURBS;

- e) Podem representar outras representações paramétricas importantes, como B-splines, Bézier racionais e não-rationais.

As bases racionais requerem que cada ponto de controle tenha um peso, estes alteram a influência de cada ponto de controle sobre NURBS. Uma curva NURBS é expressa por:

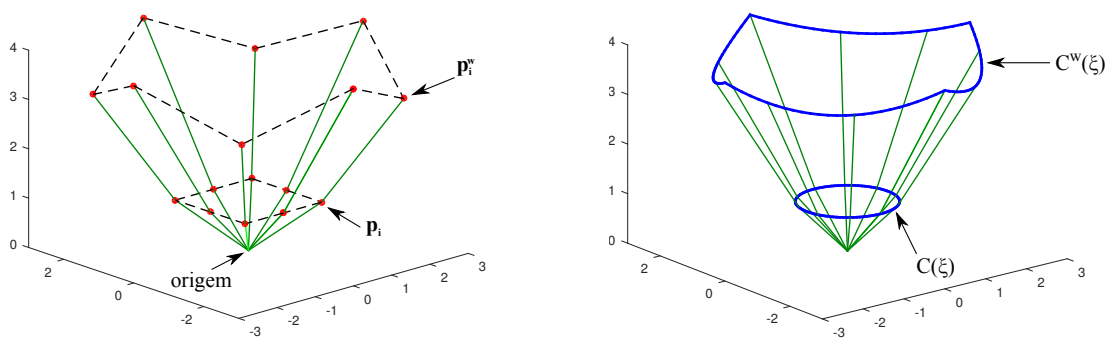
$$C(\xi) = \frac{\sum_{i=1}^n w_i N_{i,p}(\xi) \mathbf{p}_i}{\sum_{i=1}^n w_i N_{i,p}(\xi)} \quad (47)$$

onde w_i é o peso associado a cada ponto de controle \mathbf{p}_i . Uma NURBS descrita no espaço \mathbb{R}^d é obtida da projeção de uma B-spline descrita em \mathbb{R}^{d+1} . Os pontos de projeção \mathbf{p}_i^w podem ser calculados por:

$$(\mathbf{p}_i^w)_j \begin{cases} (\mathbf{p}_i)_j w_i & \text{se } j < d+1 \\ w_i & \text{se } j = d+1 \end{cases} \quad (48)$$

Uma circunferência modelada por uma curva NURBS $C(\xi)$ em $\mathbb{R}^2 (x,y)$ é obtida da projeção de uma B-spline $C(\xi)^w$ em $\mathbb{R}^3 (x,w,y,w)$ sobre o plano $w = 1$, como mostra a Figura 21.

Figura 21 – Curva NURBS $C(\xi)$ em \mathbb{R}^2 sendo obtida da projeção de uma curva B-spline $C^w(\xi)$ em \mathbb{R}^3 .



a) Pontos de controle.

b) Curva.

Fonte: Elaborada pelo autor.

É importante lembrar que os algoritmos de *inserção de knot* e *elevação de grau* podem ser aplicados a NURBS da mesma forma que são aplicados a B-splines se forem aplicados considerando os pontos de controle em \mathbb{R}^{d+1} . A *inserção de knots* pode ser aplicada transformando os pontos de controle em \mathbb{R}^d para o espaço \mathbb{R}^{d+1} ($\mathbf{p} \rightarrow \mathbf{p}^w$) através da Equação (48),

em seguida aplicar a Equação (45) para encontrar novos pontos de controle $\bar{\mathbf{p}}^w$, e finalmente transforma os pontos de volta para o espaço \mathbb{R}^d :

$$(\bar{\mathbf{p}}_i)_j = \frac{(\bar{\mathbf{p}}_i^w)_j}{\bar{w}_i} \quad j = 0, \dots, d \quad (49)$$

onde $\bar{w}_i = (\bar{\mathbf{p}}_i^w)_{d+1}$

As funções de base NURBS $R_{i,p}$ podem ser escritas em função das bases B-spline $N_{i,p}$ e dos pesos w_i como:

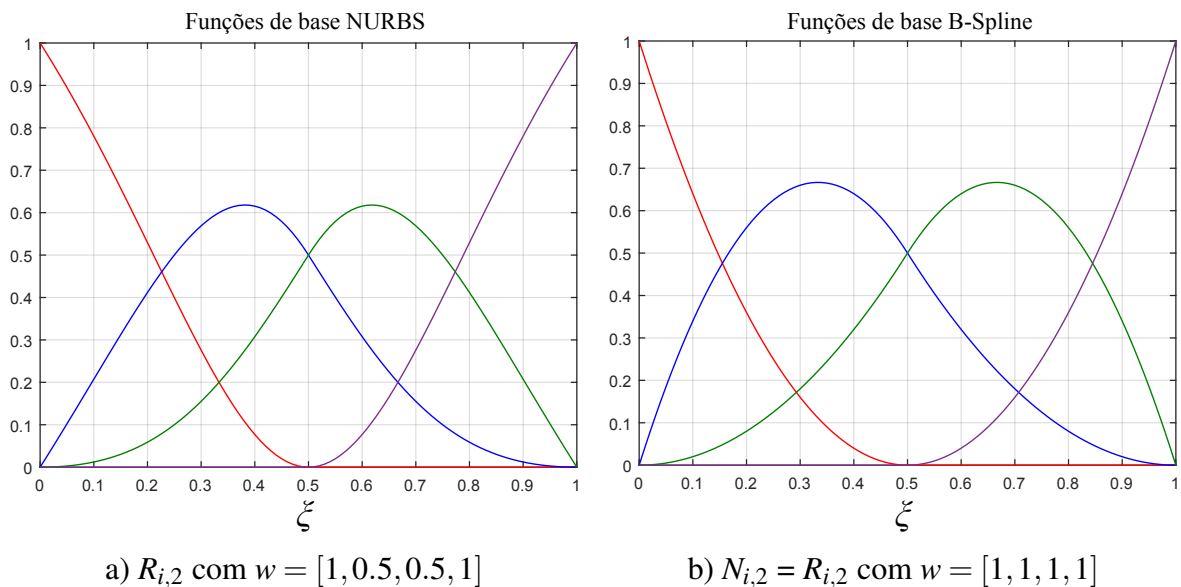
$$R_{i,p}(\xi) = \frac{N_{i,p}(\xi) w_i}{W(\xi)} = \frac{N_{i,p}(\xi) w_i}{\sum_{\hat{i}=1}^n N_{\hat{i},p}(\xi) w_{\hat{i}}} \quad (50)$$

onde $W(\xi)$ é a função peso, dada por:

$$W(\xi) = \sum_{\hat{i}=1}^n N_{\hat{i},p}(\xi) w_{\hat{i}} \quad (51)$$

A Figura 22 mostra as funções de base racionais $R_{i,2}$ e não racionais $N_{i,2}$, considerando um vetor de *knots* $\Xi = [0, 0, 0, 0.5, 1, 1, 1]$ e pesos $w = [1, 0.5, 0.5, 1]$, pode-se observar uma pequena diferença nas curvas de cada base. A Figura 23 mostra as curvas de uma semi-circunferência modelada com NURBS e o arco de 180° resultante de B-spline não racional ($w = 1$).

Figura 22 – Bases NURBS e B-spline quadráticas com $\Xi = [0, 0, 0, 0.5, 1, 1, 1]$.



Fonte: Elaborada pelo autor.

A primeira derivada das bases NURBS racionais podem ser obtidas pela regra de derivação do quociente:

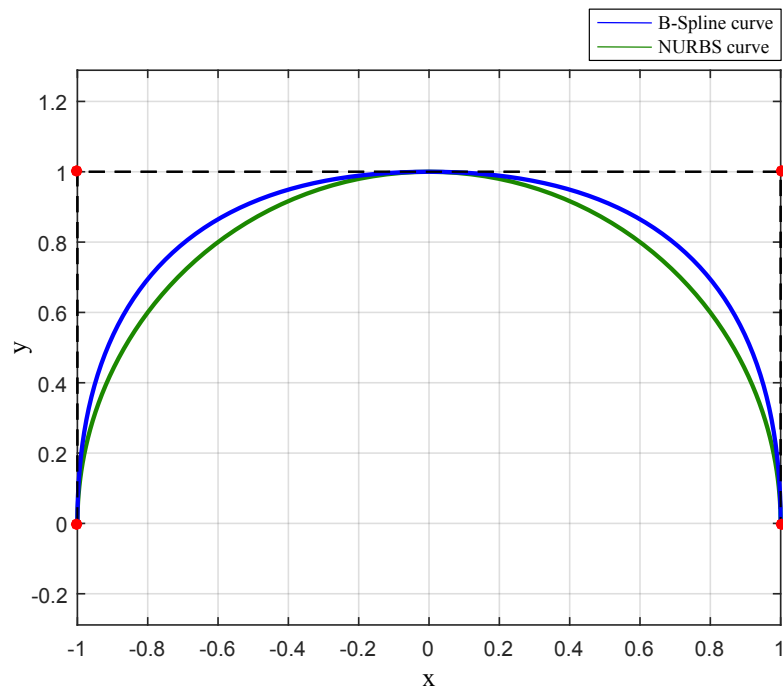
$$\frac{d}{d\xi} R_i = w_i \frac{W(\xi) N'_{i,p}(\xi) - W'(\xi) N_{i,p}(\xi)}{W^2(\xi)} \quad (52)$$

onde a derivada da função peso é dada por:

$$W'(\xi) = \sum_{\hat{i}=1}^n N'_{\hat{i},p}(\xi) w_{\hat{i}} \quad (53)$$

É importante ressaltar que as funções de base racionais $R_{i,p}(\xi)$ herdam as mesmas propriedades das funções de base B-spline $N_{i,p}(\xi)$, descritas na Seção 3.3.

Figura 23 – Semi circunferência construída com NURBS e um arco de 180° construído com B-spline.



Fonte: Elaborada pelo autor.

3.4.1 Superfícies NURBS

Uma superfície NURBS definida por produto tensorial é avaliada pelo produto de duas funções de base univariantes NURBS. Dada uma superfície S definida por uma matriz de pontos de controle \mathbf{P} ($n \times m$), uma NURBS de grau p na direção ξ com vetor de *knots* $\Xi = [\xi_1, \xi_1, \dots, \xi_{n+p+1}]$ e outra NURBS de grau q na direção η com vetor de *knots*

$\mathcal{H} = [\eta_1, \eta_1, \dots, \eta_{m+q+1}]$, a superfície pode ser calculada por:

$$S(\xi, \eta) = \sum_{i=1}^n \sum_{j=1}^m R(\xi, \eta)_{ij} \mathbf{P}_{ij} \quad (54)$$

onde $R(\xi, \eta)$ é a função de base racional bivariante dada por:

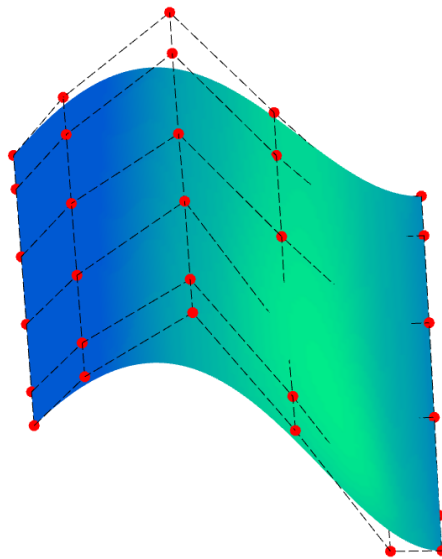
$$R(\xi, \eta)_{ij} = \frac{w_{ij} N_{i,p}(\xi) N_{j,q}(\eta)}{W(\xi, \eta)}, \quad (55)$$

onde $W(\xi, \eta)$ é a função peso bivariante:

$$W(\xi, \eta) = \sum_{\hat{i}=1}^n \sum_{\hat{j}=1}^m w_{\hat{i}\hat{j}} N_{\hat{i},p}(\xi) N_{\hat{j},q}(\eta) \quad (56)$$

A Figura 24 mostra um exemplo de superfície NURBS.

Figura 24 – Exemplo de superfície NURBS.



Fonte: Elaborada pelo autor.

As derivadas parciais de $W(\xi, \eta)$ são expressas por:

$$\begin{aligned} \frac{\partial}{\partial \xi} W(\xi, \eta) &= \sum_{\hat{i}=1}^n \sum_{\hat{j}=1}^m w_{\hat{i}\hat{j}} N'_{\hat{i},p}(\xi) N_{\hat{j},q}(\eta) \\ \frac{\partial}{\partial \eta} W(\xi, \eta) &= \sum_{\hat{i}=1}^n \sum_{\hat{j}=1}^m w_{\hat{i}\hat{j}} N_{\hat{i},p}(\xi) N'_{\hat{j},q}(\eta) \end{aligned} \quad (57)$$

Assim, as derivadas parciais das funções de base bivariantes são calculadas por:

$$\begin{aligned} \frac{\partial}{\partial \xi} R_{ij}(\xi, \eta) &= w_{ij} \frac{W(\xi, \eta) N'_{i,p}(\xi) N_{j,q}(\eta) - \frac{\partial}{\partial \xi} W(\xi, \eta) N_{i,p}(\xi) N_{j,q}(\eta)}{W^2(\xi, \eta)} \\ \frac{\partial}{\partial \eta} R_{ij}(\xi, \eta) &= w_{ij} \frac{W(\xi, \eta) N_{i,p}(\xi) N'_{j,q}(\eta) - \frac{\partial}{\partial \eta} W(\xi, \eta) N_{i,p}(\xi) N_{j,q}(\eta)}{W^2(\xi, \eta)} \end{aligned} \quad (58)$$

As derivadas parciais são necessárias na formulação de elementos isogeométricos bidimensionais.

3.4.2 Sólidos NURBS

Um sólido NURBS definido por produto tensorial é construído pelo produto de três funções de base univariantes NURBS. Dada um sólido V definido por um tensor de pontos de controle \mathbf{P} ($n \times m \times o$), uma NURBS de grau p na direção ξ com vetor de *knots* $\Xi = [\xi_1, \xi_1, \dots, \xi_{n+p+1}]$, uma NURBS de grau q na direção η com vetor de *knots* $\mathcal{H} = [\eta_1, \eta_1, \dots, \eta_{m+q+1}]$ e uma NURBS de grau l na direção ζ com vetor de *knots* $\mathcal{L} = [\zeta_1, \zeta_1, \dots, \zeta_{o+l+1}]$, o sólido é definido por:

$$V(\xi, \eta, \zeta) = \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^o R(\xi, \eta, \zeta)_{ijk} \mathbf{P}_{ijk} \quad (59)$$

onde $R(\xi, \eta, \zeta)$ é a função de base racional trivariante dada por:

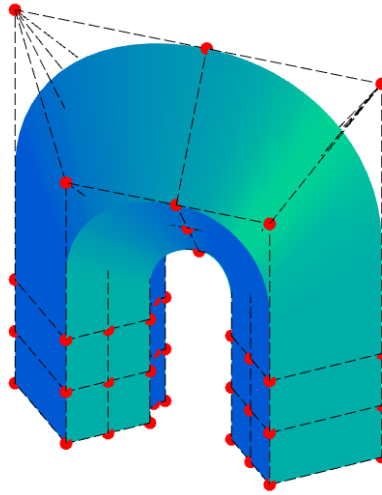
$$R(\xi, \eta, \zeta)_{i,j,k} = \frac{w_{ijk} N_{i,p}(\xi) N_{j,q}(\eta) N_{k,l}(\zeta)}{W(\xi, \eta, \zeta)} \quad (60)$$

e $W(\xi, \eta, \zeta)$ é a função peso trivariante:

$$W(\xi, \eta, \zeta) = \sum_{\hat{i}=1}^n \sum_{\hat{j}=1}^m \sum_{\hat{k}=1}^o w_{\hat{i}\hat{j}\hat{k}} N_{\hat{i},p}(\xi) N_{\hat{j},q}(\eta) N_{\hat{k},l}(\zeta) \quad (61)$$

Uma exemplo de sólido NURBS está apresentado na Figura 25.

Figura 25 – Exemplo de um sólido NURBS com seus pontos de controle.



Fonte: Elaborada pelo autor.

As derivadas parciais de $W(\xi, \eta, \zeta)$ são avaliadas pelas expressões:

$$\begin{aligned} \frac{\partial}{\partial \xi} W(\xi, \eta, \zeta) &= \sum_{\hat{i}=1}^n \sum_{\hat{j}=1}^m \sum_{\hat{k}=1}^o w_{\hat{i}\hat{j}\hat{k}} N'_{\hat{i},p}(\xi) N_{\hat{j},q}(\eta) N_{\hat{k},l}(\zeta) \\ \frac{\partial}{\partial \eta} W(\xi, \eta, \zeta) &= \sum_{\hat{i}=1}^n \sum_{\hat{j}=1}^m \sum_{\hat{k}=1}^o w_{\hat{i}\hat{j}\hat{k}} N_{\hat{i},p}(\xi) N'_{\hat{j},q}(\eta) N_{\hat{k},l}(\zeta) \\ \frac{\partial}{\partial \zeta} W(\xi, \eta, \zeta) &= \sum_{\hat{i}=1}^n \sum_{\hat{j}=1}^m \sum_{\hat{k}=1}^o w_{\hat{i}\hat{j}\hat{k}} N_{\hat{i},p}(\xi) N_{\hat{j},q}(\eta) N'_{\hat{k},l}(\zeta) \end{aligned} \quad (62)$$

As derivadas parciais das funções de base trivariantes podem ser calculadas por:

$$\begin{aligned} \frac{\partial}{\partial \xi} R_{i,j,k}(\xi, \eta, \zeta) &= w_{ijk} N_{j,q}(\eta) N_{k,l}(\zeta) \frac{W(\xi, \eta, \zeta) N'_{i,p}(\xi) - \frac{\partial}{\partial \xi} W(\xi, \eta, \zeta) N_{i,p}(\xi)}{W^2(\xi, \eta, \zeta)} \\ \frac{\partial}{\partial \eta} R_{i,j,k}(\xi, \eta, \zeta) &= w_{ijk} N_{i,p}(\xi) N_{k,l}(\zeta) \frac{W(\xi, \eta, \zeta) N'_{j,q}(\eta) - \frac{\partial}{\partial \eta} W(\xi, \eta, \zeta) N_{j,q}(\eta)}{W^2(\xi, \eta, \zeta)} \\ \frac{\partial}{\partial \zeta} R_{i,j,k}(\xi, \eta, \zeta) &= w_{ijk} N_{i,p}(\xi) N_{j,q}(\eta) \frac{W(\xi, \eta, \zeta) N'_{k,l}(\zeta) - \frac{\partial}{\partial \zeta} W(\xi, \eta, \zeta) N_{k,l}(\zeta)}{W^2(\xi, \eta, \zeta)} \end{aligned} \quad (63)$$

Estas derivadas são utilizadas na formulação de elementos sólidos isogeométricos.

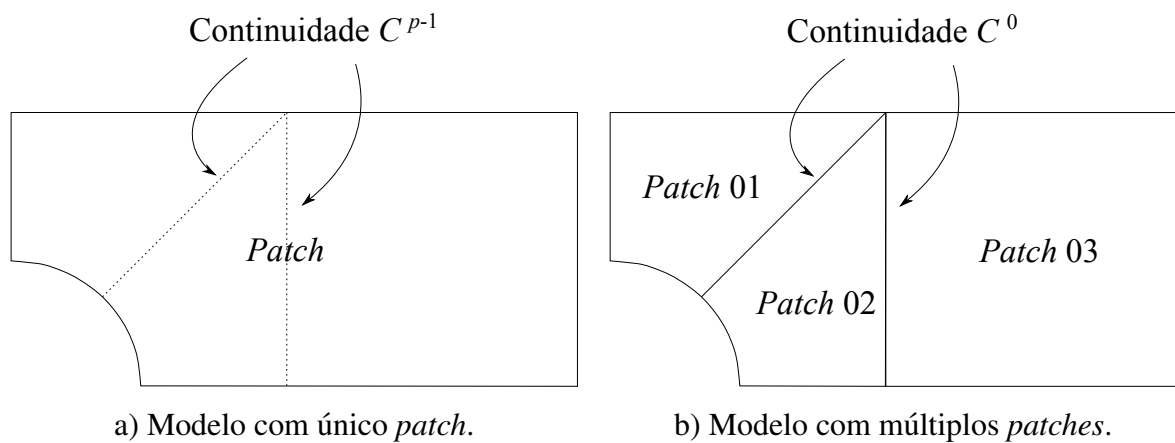
3.4.3 Múltiplos Patches

Ainda que normalmente uma entidade NURBS ser suficiente para modelagem de geometrias complexas, muitas vezes pode ser útil representar o modelo através de múltiplas entidades (curvas, superfícies ou sólidos). Por exemplo, quando várias regiões do modelo apre-

sentam diferentes atributos, como material, carregamento e condições de contorno, é necessário representar tais regiões por entidades distintas.

A Figura 26 mostra uma superfície plana, que pode ser modelada por 1 *patch* ou de maneira mais fácil com 3 *patches* (apesar da termo estar relacionado à superfícies, a palavra *patch* é normalmente utilizada como entidade NURBS, muitas referências da literatura usam o termo mesmo em caso de curvas e sólidos).

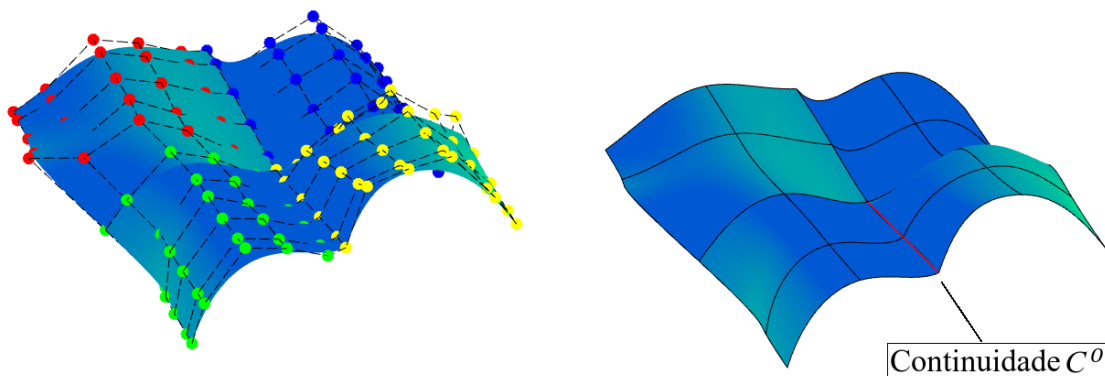
Figura 26 – Exemplo de modelo considerando múltiplos *patches*.



Fonte: Elaborada pelo autor.

A Figura 27 mostra uma superfície modelada com 4 *patches*. Uma utilidade do uso de múltiplos *patches* em caso de problemas de compósitos laminados é permitir que diferentes laminações sejam consideradas para cada *patch*.

Figura 27 – Superfície NURBS modelada com quatro *patches*.



Fonte: Elaborada pelo autor.

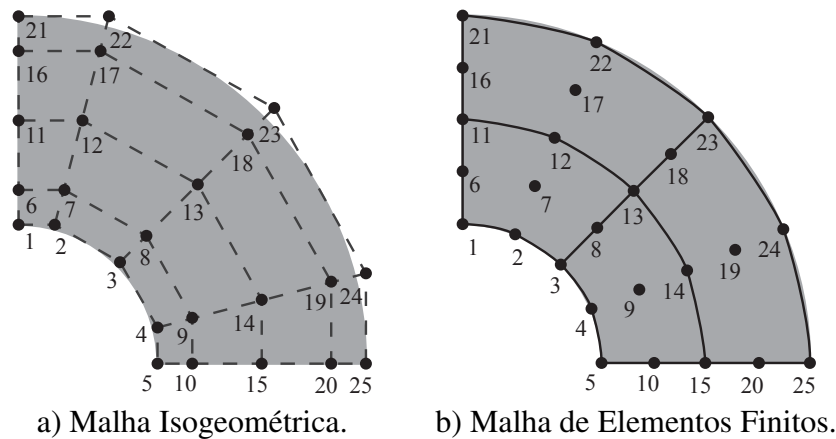
Apesar da utilização de vários *patches* ajudar na modelagem, a continuidade do modelo é reduzida para C^0 nas regiões de contorno dos *patches*, como apresentado nas Figuras

26 e 27. Mesmo que inicialmente exista uma igualdade nos gradientes de ambos os *patches* nas regiões de contorno, não há garantias que durante a análise isométrica esta condição se mantenha.

4 ANÁLISE ISOGEOMÉTRICA

A Análise Isogeométrica é um método de análise estrutural que utiliza na análise numérica as mesmas funções utilizadas pelos sistemas CAD para modelagem da geometria. O método possui muitas semelhanças com o Método dos Elementos Finitos (MEF), que é o método mais utilizado atualmente para análise de sólidos e estruturas. A grande diferença de ambos reside na capacidade que a Análise Isogeométrica tem de tratar a geometria do modelo de forma exata. A Figura 28 mostra as malhas semelhantes da AIG e do MEF, onde são utilizados elementos quadráticos e o mesmo número de nós (pontos de controle no caso da AIG). Pode-se notar que os pontos de controle da malha isogeométrica não interpolam toda a extremidade da estrutura.

Figura 28 – Malhas da AIG e MEF para um problema bidimensional.



Fonte: Borden et al. (2011).

No MEF, a estrutura é dividida em um conjunto de elementos conectados através de nós. Os deslocamentos no interior de cada elemento são interpolados a partir dos deslocamentos de seus nós utilizando polinômios. As equações de equilíbrio da estrutura são obtidas utilizando princípios variacionais (trabalho virtual ou energia potencial) ou Métodos de Resíduos Ponderados (e.g. Galerkin). Na formulação isoparamétrica do MEF a geometria dos elementos é descrita pelas mesmas funções utilizadas para aproximar os deslocamentos. Esta formulação é largamente utilizada, pois garante que as condições necessárias para a convergência do MEF para a solução exata do problema sejam automaticamente satisfeitas¹³.

A Análise Isogeométrica (AIG) utiliza a mesma ideia da formulação isoparamétrica do MEF, mas neste caso a sequência é invertida: os deslocamentos no interior do sólido são aproximados utilizando as mesmas funções utilizadas para definir a geometria do sólido (e.g. B-splines e NURBS). Desta forma, consegue-se representar de forma exata a geometria do modelo e manter a garantia de convergência para a solução do problema.

A AIG foi proposta inicialmente em Hughes e colaboradores^{14,15}, onde foi aplicada para problemas estruturais lineares modelados como sólidos. Outros trabalhos abordando a AIG

foram desenvolvidos, incluindo trabalhos abordando elementos de placas de Kirchhoff-Love⁴⁵, elementos de placa laminada¹⁶, elementos de casca⁴⁶, elementos de casca laminada⁴⁷ e elementos sólido-casca⁴⁸. No presente trabalho foi desenvolvida uma formulação isogeométrica para análise geometricamente não linear de sólidos laminados.

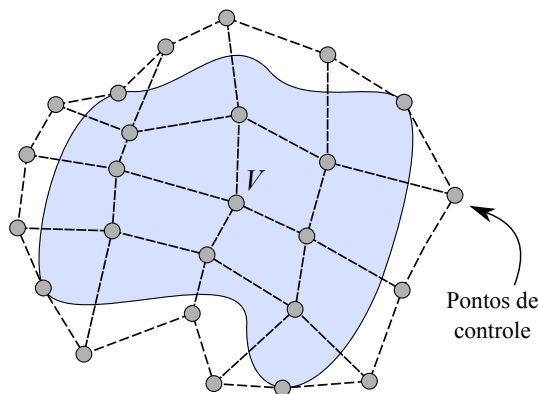
4.1 Geometria e Deslocamentos

A análise estática de um problema estrutural consiste em determinar os campos de deslocamentos, deformações e tensões da estrutura. Dada uma estrutura de volume V , a Análise Isogeométrica considera que o corpo é representado por uma combinação linear de np funções de base R_a e pontos de controle $\mathbf{p}_a = (x_a, y_a, z_a)$, expresso por:

$$x = \sum_{a=1}^{np} R_a x_a; \quad y = \sum_{a=1}^{np} R_a y_a; \quad z = \sum_{a=1}^{np} R_a z_a \quad (64)$$

Considerando que a geometria seja descrita por NURBS, como ilustrado na Figura 29, as funções R_a serão as funções de base racional $R_{ijk}(\xi, \eta, \zeta)$ definidas pela Equação (59).

Figura 29 – Descrição de uma estrutura por NURBS.



Fonte: Elaborada pelo autor.

O conceito isoparamétrico é utilizado, de modo que as mesmas funções de base descritas na geometria do corpo pela Equação (64) são utilizadas para aproximação dos deslocamentos:

$$u = \sum_{a=1}^{np} R_a u_a; \quad v = \sum_{a=1}^{np} R_a v_a; \quad w = \sum_{a=1}^{np} R_a w_a \quad (65)$$

Representando os graus de liberdade por um vetor $\mathbf{u} = [u_1, v_1, w_1, \dots, u_{np}, v_{np}, w_{np}]^T$, esta equa-

ção pode ser escrita em forma matricial:

$$\hat{\mathbf{u}} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \sum_{a=1}^{np} \begin{bmatrix} R_a & 0 & 0 \\ 0 & R_a & 0 \\ 0 & 0 & R_a \end{bmatrix} \begin{bmatrix} u_a \\ v_a \\ w_a \end{bmatrix} = \mathbf{N}\mathbf{u} \quad (66)$$

onde $\hat{\mathbf{u}}$ são os deslocamentos no interior do sólido (o símbolo $\hat{\mathbf{u}}$ é utilizado para diferenciar os deslocamentos dos graus de liberdade dos pontos de controle, que são dados por \mathbf{u}) e \mathbf{N} é a matriz de aproximação dos deslocamentos da estrutura. Tal matriz tem formato definido pelas submatrizes \mathbf{N}_a :

$$\mathbf{N} = \begin{bmatrix} \mathbf{N}_1 & \mathbf{N}_2 & \dots & \mathbf{N}_{np} \end{bmatrix} \Rightarrow \mathbf{N}_a = \begin{bmatrix} R_a & 0 & 0 \\ 0 & R_a & 0 \\ 0 & 0 & R_a \end{bmatrix} \quad (67)$$

A definição das deformações do sólido utiliza as derivadas no espaço físico (x, y, z) das funções base. Tais derivadas pode ser obtidas aplicando a regra da cadeia sobre as derivadas paramétricas. Considerando a derivada paramétrica $R_{a,\xi}$, observando que a notação $R_{a,\xi}$ é uma abreviação de $\frac{\partial R_a}{\partial \xi}$, temos que:

$$R_{a,\xi} = R_{a,x} x_\xi + R_{a,y} y_\xi + R_{a,z} z_\xi \quad (68)$$

As derivadas x_ξ , y_ξ e z_ξ são obtidas pela Equação (64):

$$x_\xi = \sum_{a=1}^{np} R_{a,\xi} x_a; \quad y_\xi = \sum_{a=1}^{np} R_{a,\xi} y_a; \quad z_\xi = \sum_{a=1}^{np} R_{a,\xi} z_a \quad (69)$$

Substituindo na Equação (68), obtemos:

$$R_{a,\xi} = R_{a,x} \sum_{a=1}^{np} R_{a,\xi} x_a + R_{a,y} \sum_{a=1}^{np} R_{a,\xi} y_a + R_{a,z} \sum_{a=1}^{np} R_{a,\xi} z_a \quad (70)$$

Realizando o mesmo procedimento para as derivadas $R_{a,\eta}$ e $R_{a,\zeta}$ e escrevendo as equações em formato matricial, chegamos em:

$$\begin{bmatrix} R_{a,\xi} \\ R_{a,\eta} \\ R_{a,\zeta} \end{bmatrix} = \mathbf{J} \begin{bmatrix} R_{a,x} \\ R_{a,y} \\ R_{a,z} \end{bmatrix} \quad (71)$$

onde \mathbf{J} é a matriz Jacobiana, dada por:

$$\mathbf{J} = \begin{bmatrix} \sum_{a=1}^{np} R_{a,\xi} x_a & \sum_{a=1}^{np} R_{a,\xi} y_a & \sum_{a=1}^{np} R_{a,\xi} z_a \\ \sum_{a=1}^{np} R_{a,\eta} x_a & \sum_{a=1}^{np} R_{a,\eta} y_a & \sum_{a=1}^{np} R_{a,\eta} z_a \\ \sum_{a=1}^{np} R_{a,\zeta} x_a & \sum_{a=1}^{np} R_{a,\zeta} y_a & \sum_{a=1}^{np} R_{a,\zeta} z_a \end{bmatrix} \quad (72)$$

onde $R_{a,\xi}$, $R_{a,\eta}$ e $R_{a,\zeta}$ são respectivamente as derivadas parciais $\frac{\partial R}{\partial \xi}$, $\frac{\partial R}{\partial \eta}$ e $\frac{\partial R}{\partial \zeta}$ da função base tri-variante das NURBS, definidas pela Equação (63). As derivadas no espaço físico são calculadas por:

$$\begin{bmatrix} R_{a,x} \\ R_{a,y} \\ R_{a,z} \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} R_{a,\xi} \\ R_{a,\eta} \\ R_{a,\zeta} \end{bmatrix} \quad (73)$$

4.2 Deformações e tensões

Neste trabalho foi utilizado a Formulação Lagrangeana Total^{49,50}. Nesta formulação são consideradas as deformações de Green-Lagrange e as tensões de Piolla-Kirchhoff II. Tais tensões e deformações são avaliadas sempre na configuração inicial da estrutura.

As deformações de Green-Lagrange $\boldsymbol{\epsilon}$ podem ser escritas em função de uma parcela linear $\boldsymbol{\epsilon}_l$ e uma parcela não-linear $\boldsymbol{\epsilon}_{nl}$, como mostra a expressão matricial:

$$\boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_x \\ \epsilon_y \\ \epsilon_z \\ \gamma_{xy} \\ \gamma_{xz} \\ \gamma_{yz} \end{bmatrix} = \underbrace{\begin{bmatrix} u_x \\ v_y \\ w_z \\ u_y + v_x \\ u_z + w_x \\ u_z + w_y \end{bmatrix}}_{\boldsymbol{\epsilon}_l} + \underbrace{\begin{bmatrix} \frac{1}{2}(u_x^2 + v_x^2 + w_x^2) \\ \frac{1}{2}(u_y^2 + v_y^2 + w_y^2) \\ \frac{1}{2}(u_z^2 + v_z^2 + w_z^2) \\ u_x u_y + v_x v_y + w_x w_y \\ u_x u_z + v_x v_z + w_x w_z \\ u_y u_z + v_y v_z + w_y w_z \end{bmatrix}}_{\boldsymbol{\epsilon}_{nl}} \quad (74)$$

onde as derivadas dos deslocamentos u , v e w podem ser calculadas por:

$$\begin{aligned} u_x &= \sum_{a=1}^{np} R_{a,x} u_a; & u_y &= \sum_{a=1}^{np} R_{a,y} u_a; & u_z &= \sum_{a=1}^{np} R_{a,z} u_a; \\ v_x &= \sum_{a=1}^{np} R_{a,x} v_a; & v_y &= \sum_{a=1}^{np} R_{a,y} v_a; & v_z &= \sum_{a=1}^{np} R_{a,z} v_a; \\ w_x &= \sum_{a=1}^{np} R_{a,x} w_a; & w_y &= \sum_{a=1}^{np} R_{a,y} w_a; & w_z &= \sum_{a=1}^{np} R_{a,z} w_a; \end{aligned} \quad (75)$$

Escrevendo as derivadas em um vetor $\beta = [u_{,x}, u_{,y}, \dots, w_{,z}]^T$, a expressão pode ser escrita em forma matricial:

$$\beta = \begin{bmatrix} u_{,x} \\ u_{,y} \\ u_{,z} \\ v_{,x} \\ v_{,y} \\ v_{,z} \\ w_{,x} \\ w_{,y} \\ w_{,z} \end{bmatrix} = \sum_{a=1}^{np} \begin{bmatrix} R_{a,x} & 0 & 0 \\ R_{a,y} & 0 & 0 \\ R_{a,z} & 0 & 0 \\ 0 & R_{a,x} & 0 \\ 0 & R_{a,y} & 0 \\ 0 & R_{a,z} & 0 \\ 0 & 0 & R_{a,x} \\ 0 & 0 & R_{a,y} \\ 0 & 0 & R_{a,z} \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ w_1 \\ \vdots \\ u_{np} \\ v_{np} \\ w_{np} \end{bmatrix} = \mathbf{G}\mathbf{u} \quad (76)$$

onde a matriz \mathbf{G} tem o mesmo formato da matriz \mathbf{N} , onde as contribuições dos graus de liberdade de cada ponto de controle são dados ao longo de três colunas, de modo análogo como ocorre na Equação (67). Utilizando o vetor β , podemos escrever o vetor de deformações como:

$$\varepsilon = \varepsilon_l + \varepsilon_{nl} = \mathbf{H}\beta + \frac{1}{2}\mathbf{A}\beta \quad (77)$$

onde as matrizes \mathbf{H} e \mathbf{A} são dadas por:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}; \quad \mathbf{A} = \begin{bmatrix} u_{,x} & 0 & 0 & v_{,x} & 0 & 0 & w_{,x} & 0 & 0 \\ 0 & u_{,y} & 0 & 0 & v_{,y} & 0 & 0 & w_{,y} & 0 \\ 0 & 0 & u_{,z} & 0 & 0 & v_{,z} & 0 & 0 & w_{,z} \\ u_{,y} & u_{,x} & 0 & v_{,y} & v_{,x} & 0 & w_{,y} & w_{,x} & 0 \\ u_{,z} & 0 & u_{,x} & v_{,z} & 0 & v_{,x} & w_{,z} & 0 & w_{,x} \\ 0 & u_{,z} & u_{,y} & 0 & v_{,z} & v_{,y} & 0 & w_{,z} & w_{,y} \end{bmatrix} \quad (78)$$

Substituindo $\beta = \mathbf{G}\mathbf{u}$ na Equação (77), obtemos:

$$\varepsilon = \mathbf{H}\mathbf{G}\mathbf{u} + \frac{1}{2}\mathbf{A}\mathbf{G}\mathbf{u} \quad (79)$$

Podemos definir a matriz que descreve a relação deformação-deslocamento \mathbf{B} por:

$$\varepsilon = \mathbf{B}\mathbf{u} \quad \Rightarrow \quad \mathbf{B} = \mathbf{B}_0 + \frac{1}{2}\mathbf{B}_L \quad (80)$$

onde \mathbf{B}_0 e \mathbf{B}_L são respectivamente a parcela linear e não-linear da relação deformação-deslocamento:

$$\mathbf{B}_0 = \mathbf{H}\mathbf{G}; \quad \mathbf{B}_L = \mathbf{A}\mathbf{G} \quad (81)$$

Considerando que o material tenha um comportamento elástico linear, pode-se obter as tensões σ por:

$$\sigma = \mathbf{C}\epsilon \quad (82)$$

onde \mathbf{C} é a matriz constitutiva elástica, dada por:

$$\mathbf{C} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} (1-\nu) & \nu & \nu & 0 & 0 & 0 \\ \nu & (1-\nu) & \nu & 0 & 0 & 0 \\ \nu & \nu & (1-\nu) & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{(1-2\nu)}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2} \end{bmatrix} \quad (83)$$

No caso de materiais compósitos, esta matriz é dada pela Equação (8). É importante notar que Devido ao uso da formulação Lagrangeana Total, as tensões σ são as tensões de Piolla-Kirchhoff II.

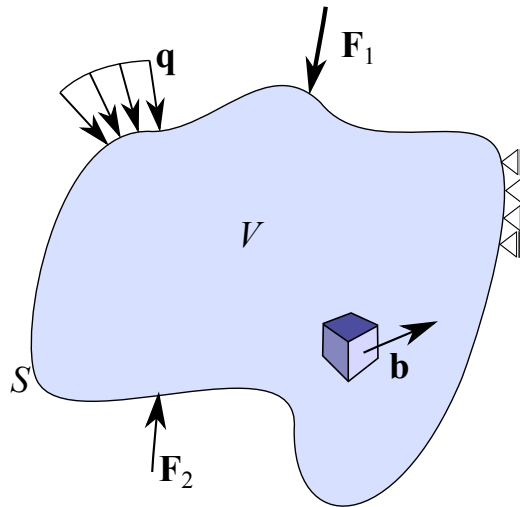
4.2.1 Equações de Equilíbrio

As equações de equilíbrio do sólido na AIG podem ser obtidas a partir do Princípio dos Trabalhos Virtuais (PTV). Dada uma estrutura ilustrada na Figura 30, o equilíbrio ocorre quando o trabalho virtual interno (δU) é igual ao trabalho virtual externo (δW_{ext}), qualquer que seja o campo de deslocamentos virtuais $\delta \hat{\mathbf{u}}$, desde que este seja pequeno e obedeça as condições de contorno essenciais do problema¹³. Matematicamente:

$$\delta U = \delta W_{ext} \Rightarrow \int_V \delta \epsilon^T \sigma dV = \int_V \delta \hat{\mathbf{u}}^T \mathbf{b} dV + \int_S \delta \hat{\mathbf{u}}^T \mathbf{q} dS + \sum \delta \hat{\mathbf{u}}_j^T \mathbf{F}_j \quad (84)$$

onde $\delta \epsilon$ é o vetor de deformações virtuais, σ é vetor das tensões, \mathbf{b} são as forças de corpo, \mathbf{q} são as forças de superfície e \mathbf{F}_j são as cargas concentradas atuantes no corpo.

Figura 30 – Corpo de volume V submetido a forças de corpo \mathbf{b} , cargas de superfícies \mathbf{q} , cargas pontuais \mathbf{F}_i .



Fonte: Elaborada pelo autor.

O trabalho virtual interno pode ser escrito em função do vetor de forças internas da estrutura (\mathbf{g}):

$$\delta U = \int_V \delta \boldsymbol{\epsilon}^T \boldsymbol{\sigma} dV = \delta \mathbf{u}^T \mathbf{g} \quad (85)$$

Partindo das deformações de Green-Lagrange definida pela Equação (79), pode-se mostrar que⁴⁹:

$$\delta \boldsymbol{\epsilon} = \bar{\mathbf{B}} \delta \mathbf{u} \quad (86)$$

com

$$\bar{\mathbf{B}} = \mathbf{B}_0 + \mathbf{B}_L \quad (87)$$

Substituindo a Equação (86) na Equação (85) e considerando deslocamentos virtuais arbitrários, obtemos que o vetor de forças internas da estrutura pode ser calculado por:

$$\mathbf{g} = \int_V \bar{\mathbf{B}}^T \boldsymbol{\sigma} dV \quad (88)$$

É importante lembrar que devido ao uso da formulação Lagrangeana Total, a integral relacionada as forças internas e a integral relacionada aos carregamentos externos são realizadas ao longo do volume inicial da estrutural.

Por outro lado, utilizando as Equações (67) e (84), o trabalho virtual externo pode

ser escrito como:

$$\delta W_{ext} = \delta \mathbf{u}^T \int_V \mathbf{N}^T \mathbf{b} dV + \delta \mathbf{u}^T \int_S \mathbf{N}^T \mathbf{q} dS + \delta \mathbf{u}^T \sum \mathbf{N}_j^T \mathbf{F}_j = \delta \mathbf{u}^T \mathbf{f} \quad (89)$$

Portanto, o vetor das cargas externas (\mathbf{f}) é dado por

$$\mathbf{f} = \int_V \mathbf{N}^T \mathbf{b} dV + \int_S \mathbf{N}^T \mathbf{q} dS + \sum \mathbf{N}_j^T \mathbf{F}_j \quad (90)$$

Nesta expressão, a matriz \mathbf{N}_j corresponde à matriz \mathbf{N} da Equação (67) avaliada no ponto de aplicação da força j .

As equações de equilíbrio do modelo discreto são obtidas substituindo as Equações (85) e (89) na expressão do trabalho virtual:

$$\delta U = \delta W_{ext} \Rightarrow \delta \mathbf{u}^T \mathbf{g} = \delta \mathbf{u}^T \mathbf{f} \quad (91)$$

Como os deslocamentos virtuais $\delta \mathbf{u}$ são arbitrários, as equações de equilíbrio da AIG são dadas por

$$\mathbf{g} = \mathbf{f} \quad (92)$$

É importante notar que esta equação é válida para problemas não lineares físicos e geométricos, independente de quais modelos constitutivos sejam utilizados para descrever o comportamento dos materiais.

Em um problema não-linear tal que os carregamentos sejam independentes dos deslocamentos, as equações de equilíbrio podem ser escritas como:

$$\mathbf{r}(\mathbf{u}, \lambda) = \mathbf{g}(\mathbf{u}) - \lambda \mathbf{q} = 0 \quad (93)$$

onde \mathbf{u} representa o vetor de deslocamentos dos pontos de controle, \mathbf{g} é o vetor de forças internas, \mathbf{q} é o vetor de cargas externas de referência, \mathbf{r} é o vetor dos resíduos (forças desbalanceadas) e λ é o fator de carga, tal fator controla a aplicação do carregamento ($\mathbf{f} = \lambda \mathbf{q}$) na estrutura.

A solução da Equação (93) requer que seja utilizado um método de solução não-linear, como o Método de Newton-Raphson⁴⁹. Neste trabalho foram utilizados diversos métodos de solução não-linear oriundos do Método de Newton-Raphson, como o Método do Controle de Carga e o Método do Comprimento de Arco. Uma explicação mais aprofundada de tais métodos foge do escopo deste trabalho, podendo ser encontradas em outras referências^{18,20,49}.

4.3 Matriz de Rigidez Tangente

A matriz de rigidez tangente (\mathbf{K}_T) da estrutura é obtida através da diferenciação do vetor de forças internas \mathbf{g} , como dado pela expressão:

$$\mathbf{K}_T = \underbrace{\int_V \bar{\mathbf{B}}^T \frac{\partial \sigma}{\partial \mathbf{u}} dV}_{\mathbf{K}_E} + \underbrace{\int_V \frac{\partial \bar{\mathbf{B}}^T}{\partial \mathbf{u}} \sigma dV}_{\mathbf{K}_G} \quad (94)$$

onde \mathbf{K}_E e \mathbf{K}_G são respectivamente as matrizes de rigidez relacionada ao material e a geometria do problema. A derivada parcial $\frac{\partial \sigma}{\partial \mathbf{u}}$ pode ser escrita como:

$$\frac{\partial \sigma}{\partial \mathbf{u}} = \frac{\partial \sigma}{\partial \boldsymbol{\epsilon}} \frac{\partial \boldsymbol{\epsilon}}{\partial \mathbf{u}} = \mathbf{C}_T \bar{\mathbf{B}} \quad (95)$$

onde

$$\mathbf{C}_T = \frac{\partial \sigma}{\partial \boldsymbol{\epsilon}} \quad (96)$$

é a matriz constitutiva tangente. No caso de materiais isotrópicos com comportamento elástico linear, temos que $\mathbf{C}_T = \mathbf{C}$, onde \mathbf{C} é a matriz constitutiva elástica apresentada na Equação (8) ou na Equação (83). Assim, a parcela da matriz de rigidez tangente dependente do material pode ser escrita por

$$\mathbf{K}_E = \int_V \bar{\mathbf{B}}^T \mathbf{C}_T \bar{\mathbf{B}} dV \quad (97)$$

Pode-se mostrar que a matriz \mathbf{K}_G pode ser expressa por⁴⁹:

$$\mathbf{K}_G = \int_V \mathbf{G}^T \mathbf{S} \mathbf{G} dV \quad (98)$$

onde:

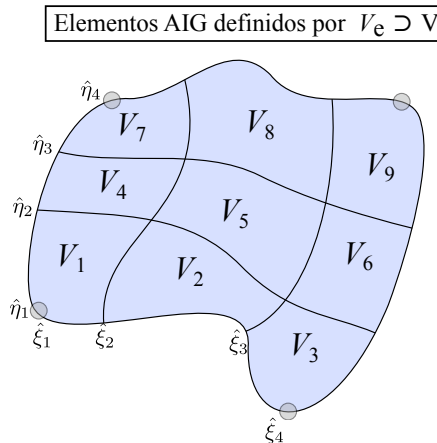
$$\mathbf{S} = \begin{bmatrix} \bar{\mathbf{S}} & 0 & 0 \\ 0 & \bar{\mathbf{S}} & 0 \\ 0 & 0 & \bar{\mathbf{S}} \end{bmatrix} \Rightarrow \bar{\mathbf{S}} = \begin{bmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{xy} & \sigma_{yy} & \tau_{yz} \\ \tau_{xz} & \tau_{yz} & \sigma_{zz} \end{bmatrix} \quad (99)$$

É importante notar que as matrizes \mathbf{K}_E e \mathbf{K}_G são obtidas de modo idêntico ao que é realizado no MEF.

4.4 Elemento Isogeométrico

Apesar das equações formuladas nas seções anteriores terem considerado as integrações ao longo do volume do sólido inteiro, pode-se realizar tais integrações ao longo do volume V_e definido por cada *knot span* do *patch*. Deste modo, o elemento isogeométrico pode ser como cada *knot span*¹⁵ da NURBS, como ilustrado na Figura 31.

Figura 31 – Elementos isogeométricos do *patch*.



Fonte: Elaborada pelo autor.

Com estas considerações, o cálculo do vetor de forças interna (\mathbf{g}) pode ser realizado por:

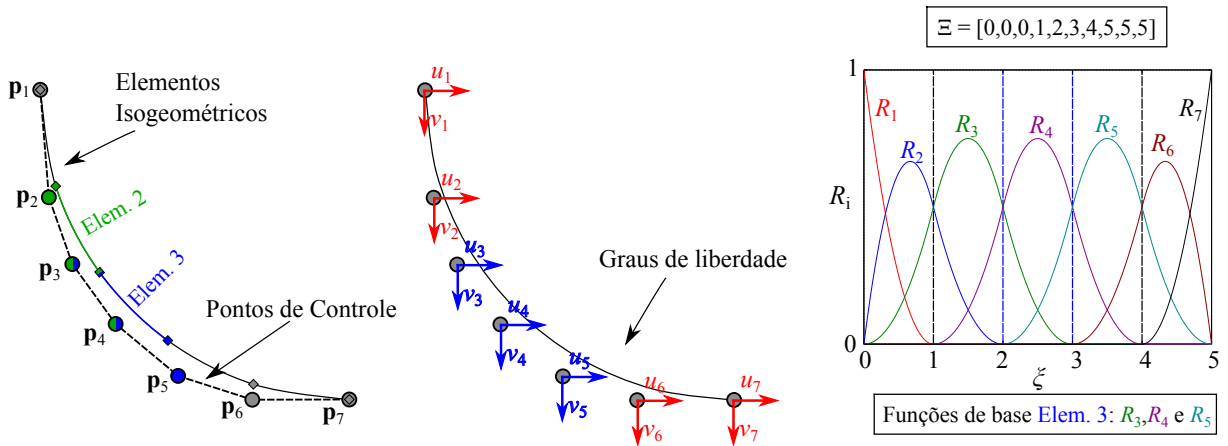
$$\mathbf{g} = \int_V \bar{\mathbf{B}}^T \boldsymbol{\sigma} dV = \sum_{e=1}^{ne} \int_{V_e} \bar{\mathbf{B}}^T \boldsymbol{\sigma} dV \quad (100)$$

onde ne é o número de elementos isogeométricos. Uma vez que as funções NURBS tem suporte compacto, apenas $p + 1$ funções são não nulas em um dado *knot span*. Logo, é mais eficiente realizar a integração considerando apenas as funções base não-nulas.

Como um número fixo de bases serão não nulas e cada base R_a está associada a um ponto de controle distinto, logo os pontos de controle que influenciam cada região V_e não mudam. Cada elemento isogeométrico possui uma incidência de graus liberdade e funções de base de número fixo no interior do *patch*, e que podem ser facilmente determinadas considerando a topologia e os pontos de controle da NURBS.

Uma vez que as incidências dos elementos sejam determinadas, o cálculo da matriz de rigidez tangente (\mathbf{K}_T), do vetor de forças internas (\mathbf{g}) e do vetor de forças externas (\mathbf{f}) são realizados em cada elemento AIG e então somados às respectivas matrizes globais, de maneira idêntica ao que é feito no MEF. Uma notável diferença entre a AIG e o MEF é o fato de que algumas das funções de base e alguns dos graus de liberdade do interior do elemento podem ser compartilhados, como ilustrado pela Figura 32. Isto ocorre porque as bases NURBS não são interpoladoras como os polinômios utilizados no MEF.

Figura 32 – Geometria, graus de liberdade e funções de base para um modelo de cabo.



Fonte: Elaborada pelo autor.

A Figura 32 mostra um cabo modelado com elementos AIG quadráticos, mostrando os pontos de controle, os graus de liberdade e as funções de base. O terceiro elemento é destacado (cor azul), mostrando os seus graus de liberdade e funções de base locais. Também são mostrados os pontos de controle do segundo elemento (cor verde), sendo possível observar um compartilhamento dos pontos de controle internos dos dois elementos.

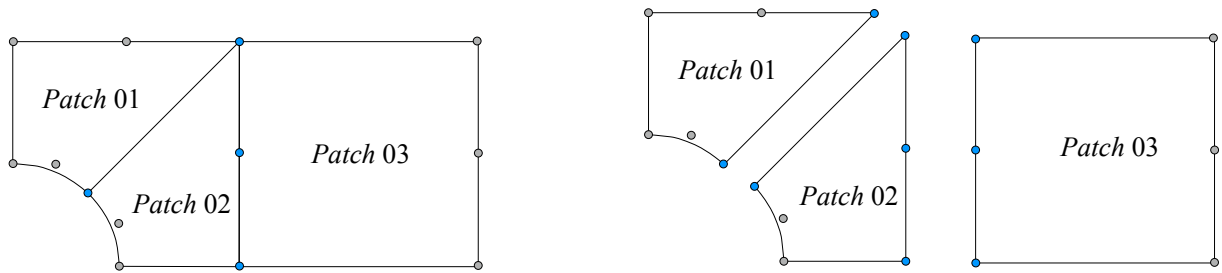
Considerando a estrutura da Figura 32, o vetor de forças internas pode ser calculado considerando apenas as bases não-nulas em cada elemento isogeométrico:

$$\mathbf{g} = \int_V \bar{\mathbf{B}}^T \boldsymbol{\sigma} dV = \underbrace{\int_{V_1} \bar{\mathbf{B}}^T \boldsymbol{\sigma} dV}_{R_1, R_2, R_3} + \underbrace{\int_{V_2} \bar{\mathbf{B}}^T \boldsymbol{\sigma} dV}_{R_2, R_3, R_4} + \underbrace{\int_{V_3} \bar{\mathbf{B}}^T \boldsymbol{\sigma} dV}_{R_3, R_4, R_5} + \underbrace{\int_{V_4} \bar{\mathbf{B}}^T \boldsymbol{\sigma} dV}_{R_4, R_5, R_6} + \underbrace{\int_{V_5} \bar{\mathbf{B}}^T \boldsymbol{\sigma} dV}_{R_5, R_6, R_7} \quad (101)$$

Um procedimento análogo pode ser realizado para o caso de elementos isogeométricos de superfície ou sólido. No caso do sólido, o número de funções não nulas será $(p+1) \times (q+1) \times (l+1)$.

É importante lembrar que sólidos simples podem ser gerados completamente por uma NURBS, mas sólidos complexos podem precisar de várias NURBS diferentes para sua descrição completa, como foi discutido na Seção 3.4.3. As equações do item anterior são independentes do número de *patches* e permanecem válidas, desde que a matriz de rigidez tangente \mathbf{K}_T , o vetor de forças internas \mathbf{g} e o vetor de cargas externas \mathbf{f} sejam calculados de forma adequada. É importante notar que os graus de liberdade dos pontos de controle de contorno dos *patches* devem ser compartilhados. A Figura 33 mostra os pontos de controle que resultam em graus de liberdade compartilhados, considerando a geometria modelada com múltiplos *patches* ilustrada pela Figura 26.

Figura 33 – Geometria modelada com múltiplos *patches*. Os pontos de controle azul resultam em graus de liberdade compartilhados.



Fonte: Elaborada pelo autor.

Assim, apesar das funções NURBS serem definidas em todo o *patch*, a matriz de rigidez será calculada utilizando a quadratura de Gauss em cada *knot span*. Na integração, o elemento é mapeado do espaço físico para o espaço paramétrico da NURBS, e em seguida mapeado para o espaço paramétrico do elemento, definido no intervalo $[-1, 1]$, como ilustrado na Figura 34 para o caso bidimensional. Isto requer que as coordenadas dos pontos de Gauss $(\hat{\xi}, \hat{\eta}, \hat{\zeta})$ sejam mapeadas para o espaço paramétrico da NURBS (ξ, η, ζ) , pela expressão:

$$\xi = \xi_{in} + (\hat{\xi} + 1) \frac{\xi_{fi} - \xi_{in}}{2}; \quad \eta = \eta_{in} + (\hat{\eta} + 1) \frac{\eta_{fi} - \eta_{in}}{2}; \quad \zeta = \zeta_{in} + (\hat{\zeta} + 1) \frac{\zeta_{fi} - \zeta_{in}}{2} \quad (102)$$

onde ξ_{in} , ξ_{fi} , η_{in} , η_{fi} , ζ_{in} e ζ_{fi} são os valores dos *knots* inicial e final do *knot span* do elemento nas direções ξ , η e ζ .

Com as coordenadas dos pontos de Gauss mapeadas, pode-se efetuar o cálculo das funções de base R_i e suas derivadas R'_i do elemento. As derivadas deverão ser mapeadas de volta para o espaço de referência do elemento, sendo necessário o cálculo do Jacobiano da transformação, com termos semelhantes a

$$\frac{\partial \xi}{\partial \hat{\xi}} = \frac{\xi_{fi} - \xi_{in}}{2}. \quad (103)$$

Uma maneira de evitar o segundo mapeamento mencionado é utilizando um procedimento de Extração de Bézier, que será discutido na Seção 5.1.

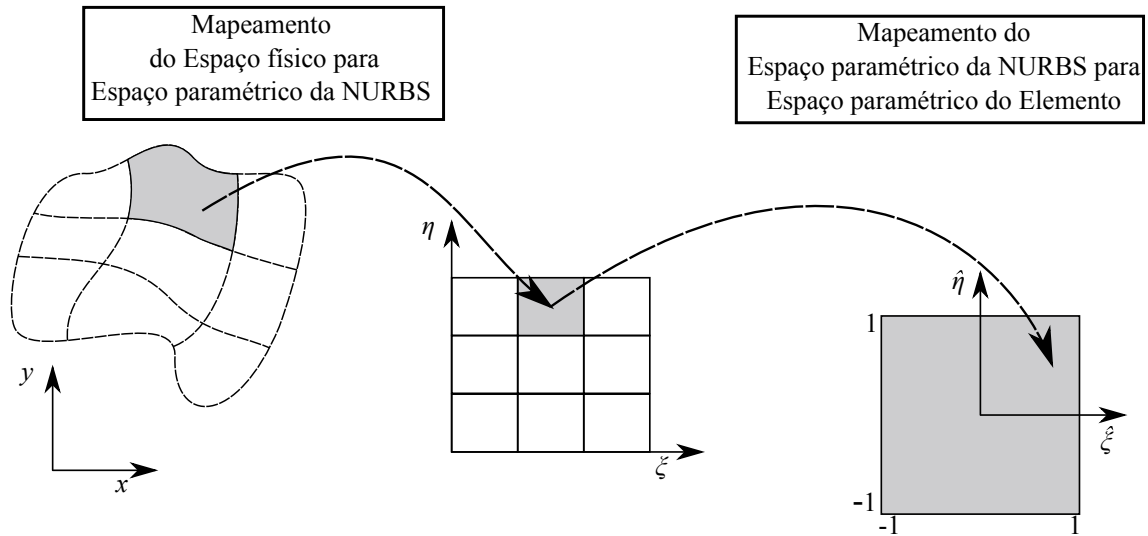
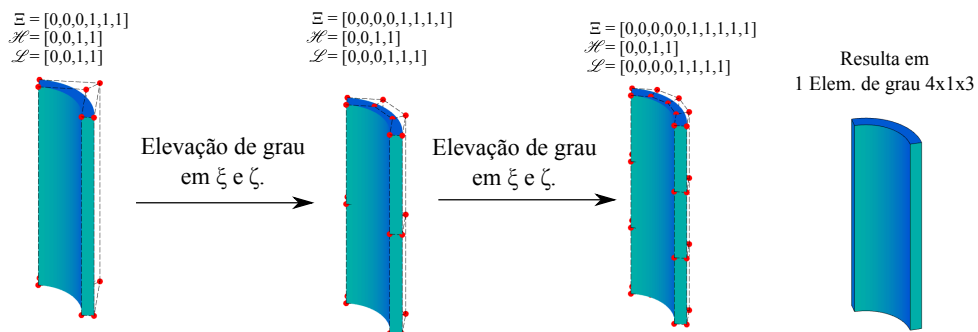


Figura 34 – Mapeamentos realizados na AIG para integração numérica.

4.5 Refinamentos

Existem três tipos de refinamentos utilizados no contexto de Análise Isogeométrica¹⁵, o refinamento p , o refinamento h e o refinamento k . O refinamento p da AIG está relacionado com o refinamento p tratado no MEF. Este processo pode ser obtido em NURBS utilizando o procedimento de Elevação de Grau abordado na Seção 3.3.1. Relembrando, a multiplicidade de cada *knot* aumenta em um para cada grau elevado, sendo necessário o cálculo de novos pontos de controle. O grau das funções base é aumentado, de maneira que a continuidade nas fronteiras do elemento é mantida. A resposta numérica melhora, uma vez que são utilizados mais graus de liberdade e funções de grau mais elevado. A Figura 35 mostra o processo do refinamento p sendo aplicado na NURBS definida pela Figura 13, é importante observar que neste processo não existe a geração de novos elementos isogeométricos.

Figura 35 – Refinamento p realizado na NURBS construída pela Figura 13.

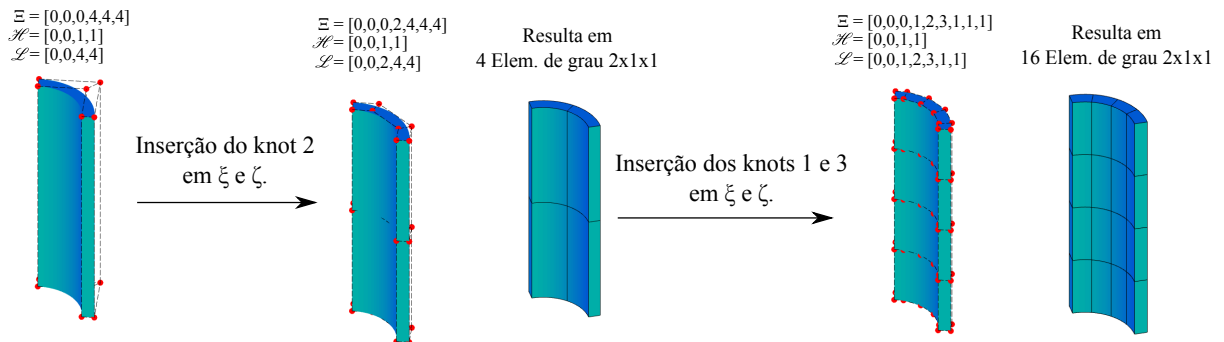


Fonte: Elaborada pelo autor.

O refinamento h da AIG está relacionado com o refinamento h tratado no MEF. Neste refinamento, o número de elementos aumenta pela aplicação do procedimento de *Inserção*

de *Knot*, abordado na Seção 3.3.1. Inserindo novos *knots*, aumenta-se o número de elementos isogeométricos (*knot spans*), melhorando a resposta numérica. A Figura 36 ilustra o processo do refinamento *h* sendo aplicado na NURBS definida pela Figura 13, o número de elementos aumenta a medida que são adicionado novos *knots*, porém o grau do elemento permanece constante.

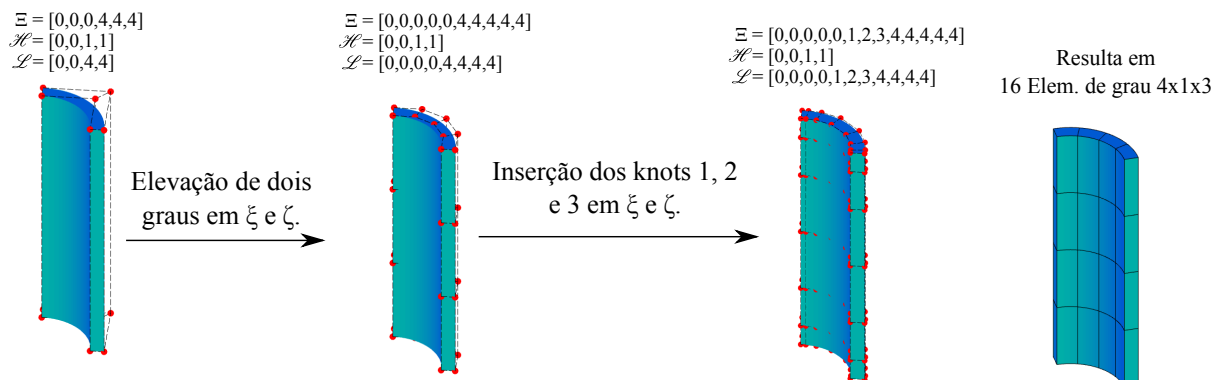
Figura 36 – Refinamento *h* realizado na NURBS construída pela Figura 13.



Fonte: Elaborada pelo autor.

O refinamento *k* é um novo tipo de refinamento inexistente no MEF. Neste refinamento, tanto o grau como o número de elementos é aumentado. Também é aumentada a continuidade entre os elementos. Isto é obtido pela aplicação em sequência da *Elevação de Grau* e *Inserção de Knot*. A geometria básica do modelo é elevada através do refinamento *p* e em seguida são adicionados novos *knots*. A Figura 37 mostra o processo do refinamento *k* sendo aplicado, considerando o refinamento *p* e *h* mostrados anteriormente. Pode-se observar que foi obtido o mesmo número de elementos obtidos pelo processo do refinamento *h* mostrado na Figura 36, porém com grau elevado e continuidade entre os elementos também elevada (C^3 em ξ , C^0 em η e C^2 em ζ).

Figura 37 – Refinamento *k* realizado na NURBS definida pela Figura 13.



Fonte: Elaborada pelo autor.

4.6 Avaliação das Tensões

Na presente formulação não linear isogeométrica, as tensões são avaliadas nos pontos de integração para o cálculo da matriz de rigidez tangente (\mathbf{K}_T) e do vetor de forças internas (\mathbf{g}), de maneira análoga ao MEF. Porém a tradicional extrapolação que é realizada nas tensões dos pontos de integração dos elementos para obtenção das tensões em qualquer ponto do modelo (por exemplo nas fronteiras do elemento, onde existem nós dos elementos) não possui a mesma aplicabilidade.

No Método dos Elementos Finitos, como a continuidade entre elementos é C^0 , as respostas de tensão nas fronteiras dos elementos são descontínuas e pior aproximadas, sendo melhor obter as respostas por extrapolação das tensões calculadas dos pontos de integração do elemento, pois tais pontos são ditos superconvergentes¹³. Por outro lado, na Análise Isogeométrica, os elementos no interior do *patch* possuem continuidade C^{p-1} (considerando que a multiplicidade dos *knots* seja 1). Como as tensões dependentes da derivada primeira da NURBS, não existe descontinuidade nas deformações na região de fronteira dos elementos caso seja utilizado grau igual ou maior que 2. Logo, as tensões nos extremos dos elementos (correspondentes aos *knot spans*) podem ser calculadas diretamente na coordenada paramétrica (ξ, η, ζ) nos extremos dos elementos utilizando as Equações (80) e (82). Vale lembrar que tal avaliação nos extremos do elemento apenas tem sentido em uma análise fisicamente linear.

É importante notar, que diferente do MEF, as tensões nas extremidades dos elementos não correspondem aos nós (i.e. pontos de controle), exceto nos vértices dos *patches* onde as bases NURBS são interpoladoras (caso seja utilizados *open knot vectors*).

4.7 Análise de Sólidos Laminados

Este trabalho utiliza a formulação tratada em Dantas Junior (2014), onde é realizada a formulação de sólidos laminados para elementos finitos BRICK8 e BRICK20. Nesta abordagem, os elementos sólidos laminados consideram que as lâminas estão dispostas ao longo da direção ζ . A matriz constitutiva de cada lâmina é transformada para o sistema global utilizando a Equação (8), sendo aplicada diretamente na integração da matriz de rigidez do elemento.

O campo de deslocamentos na direção da espessura do laminado é assumido contínuo. As deformações no interior do *patch* serão contínuas dependendo do vetor de *knots* considerado na direção ζ .

Portanto, neste trabalho os eixos do laminado são utilizados somente para a definição das direções das lâminas. Um vetor unitário $\hat{\mathbf{a}}$ é utilizado para definir a direção do vetor $\hat{\mathbf{x}}$ do laminado (ver Figura 2) em relação ao sistema global. Por exemplo, no caso de uma casca cilíndrica $\hat{\mathbf{a}}$ corresponde a um vetor unitário na direção axial do cilindro.

O vetor unitário $\hat{\mathbf{e}}_3$ é dirigido ao longo da espessura do laminado. Assim, este vetor é calculado a partir da derivada da geometria do sólido (\mathbf{x}) em relação à coordenada paramétrica

ζ :

$$\hat{\mathbf{e}}_3 = \frac{\mathbf{x}_{,\zeta}}{|\mathbf{x}_{,\zeta}|}, \quad \mathbf{x}_{,\zeta} = \sum_{a=1}^{np} R_{a,\zeta} \mathbf{x}_a \quad (104)$$

Em seguida, o vetor unitário $\hat{\mathbf{e}}_1$ que define o eixo x_1 é calculado através da rotação do vetor $\hat{\mathbf{a}}$ de um ângulo θ em relação ao eixo $\hat{\mathbf{e}}_3$:

$$\hat{\mathbf{e}}_1 = \mathbf{R} \hat{\mathbf{a}} \quad (105)$$

onde a matriz de rotação \mathbf{R} é obtida utilizando a fórmula de Rodriguez⁵¹:

$$\mathbf{R} = \mathbf{I} + \sin(\theta) \cdot \mathbf{S} + (1 - \cos(\theta)) \cdot \mathbf{S}^2 \quad (106)$$

Nesta equação, \mathbf{S} é uma matriz antissimétrica dada por

$$\mathbf{S} = \begin{bmatrix} 0 & -n_3 & n_3 \\ n_3 & 0 & -l_3 \\ -m_3 & l_3 & 0 \end{bmatrix} \quad (107)$$

onde l_3, m_3, n_3 são os cossenos diretores do vetor $\hat{\mathbf{e}}_3$. Finalmente, o vetor $\hat{\mathbf{e}}_2$ é calculado através do produto vetorial dos vetores anteriores:

$$\hat{\mathbf{e}}_2 = \hat{\mathbf{e}}_3 \times \hat{\mathbf{e}}_1 \quad (108)$$

Como a tríade $(\hat{\mathbf{e}}_1, \hat{\mathbf{e}}_2, \hat{\mathbf{e}}_3)$ é composta por vetores unitários ortogonais, suas componentes no sistema global são os cossenos diretores utilizados na Equação (4) para montar a matriz de transformação \mathbf{T} . É importante notar que como a geometria do modelo pode ser curva, o sistema de coordenadas local definido por estes três vetores pode variar ponto a ponto. Uma vantagem da formulação isogeométrica sobre o MEF é que nesta formulação os vetores que definem o sistema local, e portanto a matriz de transformação, são exatos, uma vez que a geometria do sólido é modelada de forma exata para qualquer discretização adotada.

4.7.1 Integração numérica

A principal diferença nas equações desenvolvidas anteriormente, entre elementos laminados e convencionais, é que a matriz constitutiva \mathbf{C} muda em cada lâmina. Por isso, as tensões e a matriz constitutiva são descontínuas na espessura, apesar dos deslocamentos e deformações serem contínuos no elemento. Por este motivo as integrações numéricas devem levar em consideração esta descontinuidade da matriz constitutiva.

A integração do vetor de forças internas e da matriz de rigidez tangente do elemento devem ser realizadas ao longo de cada camada. Tal procedimento será explicado utilizando a

matriz de rigidez elástica como exemplo:

$$\begin{aligned}
\mathbf{K}_E &= \int_V \bar{\mathbf{B}}^T \mathbf{C} \bar{\mathbf{B}} dV = \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \bar{\mathbf{B}}^T \mathbf{C} \bar{\mathbf{B}} |\mathbf{J}| d\xi d\eta d\zeta \\
&= \int_{-1}^{\zeta_1} \int_{-1}^1 \int_{-1}^1 \bar{\mathbf{B}}^T \mathbf{C}_1 \bar{\mathbf{B}} |\mathbf{J}| d\xi d\eta d\zeta + \dots + \int_{\zeta_k}^{\zeta_{k+1}} \int_{-1}^1 \int_{-1}^1 \bar{\mathbf{B}}^T \mathbf{C}_k \bar{\mathbf{B}} |\mathbf{J}| d\xi d\eta d\zeta \\
&\dots + \int_{\zeta_m}^1 \int_{-1}^1 \int_{-1}^1 \bar{\mathbf{B}}^T \mathbf{C}_m \bar{\mathbf{B}} |\mathbf{J}| d\xi d\eta d\zeta
\end{aligned} \tag{109}$$

onde m é o numero de lâminas do compósito e ζ_k é a coordenada paramétrica da face inferior da k -ésima lâmina. A integração numérica de cada lâmina é realizada por quadratura de Gauss, considerando que o intervalo de integração na direção ζ é igual a $[\zeta_k, \zeta_{k+1}]$. Uma forma de realizar a integração entre lâminas dentro do intervalo padrão $[-1,1]$ é mapear os valores do intervalo paramétrico $\tilde{\zeta} \in [-1, 1]$ para $\zeta \in [\zeta_k, \zeta_{k+1}]$ por:

$$\zeta = \frac{\zeta_{k+1} + \zeta_k}{2} + \frac{(\zeta_{k+1} - \zeta_k)}{2} \tilde{\zeta} \tag{110}$$

Derivando-se esta expressão verifica-se que o incremento $d\zeta$ para a k -ésima lâmina é dado por:

$$d\zeta = \frac{(\zeta_{k+1} - \zeta_k)}{2} d\tilde{\zeta} \tag{111}$$

Com isto a matriz de rigidez pode ser calculada por:

$$\begin{aligned}
\mathbf{K}_E &= \sum_{k=1}^m \int_{\zeta_k}^{\zeta_{k+1}} \int_{-1}^1 \int_{-1}^1 \bar{\mathbf{B}}^T \mathbf{C}_k \bar{\mathbf{B}} |\mathbf{J}| d\xi d\eta d\zeta \\
&= \sum_{k=1}^m \int_{-1}^1 \int_{-1}^1 \int_{-1}^1 \bar{\mathbf{B}}^T \mathbf{C}_k \bar{\mathbf{B}} |\mathbf{J}| \frac{(\zeta_{k+1} - \zeta_k)}{2} d\xi d\eta d\tilde{\zeta}
\end{aligned} \tag{112}$$

Utilizando a quadratura de Gauss:

$$\mathbf{K}_E = \sum_{k=1}^m \sum_{i=1}^{npg} (\bar{\mathbf{B}}^T \mathbf{C}_k \bar{\mathbf{B}} |\mathbf{J}|)_i \frac{(\zeta_{k+1} - \zeta_k)}{2} W_i \tag{113}$$

onde npg é o número de pontos de integração e W_i são os pesos de Gauss.

O número de pontos de integração considerados em cada lâmina é o mesmo utilizado no elemento sólido isotrópico. No caso de um elemento definido em por uma NURBS tri-cúbica com 3 lâminas, utiliza-se para integração completa 64 ($4 \times 4 \times 4$) pontos de integração por lâmina, resultando em 192 pontos de integração por elemento laminado.

5 IMPLEMENTAÇÃO COMPUTACIONAL DA AIG

Este capítulo descreve a implementação realizada no programa acadêmico de Elementos Finitos (FAST) para implementação da Análise Isogeométrica. A implementação realizada neste trabalho utiliza o conceito de Extração de Bézier para facilitar a reutilização do código de Elementos Finitos.

Inicialmente¹⁴, os programas de Análise Isogeométrica eram feitos considerando uma nova estratégia de implementação. Os tradicionais *loops* ao longo dos elementos realizados no MEF para montagem da matrizes (\mathbf{K} por exemplo) foram substituídos por *loops* a longo dos *patches*⁵². A nova estrutura de análise não era capaz de reaproveitar as implementações realizadas no MEF. Entretanto, trabalhos mostraram que era possível aproximar as estruturas dos programas da AIG e MEF caso fosse utilizado uma estratégia conhecida como Extração de Bézier.

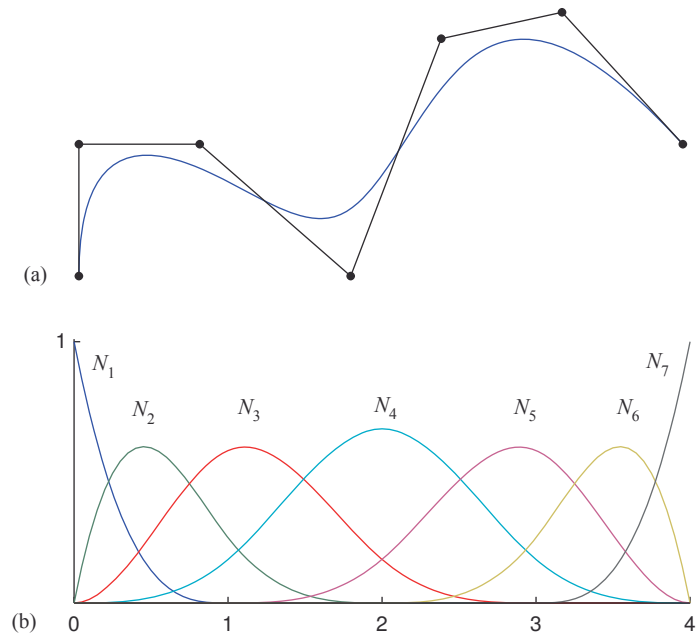
Este capítulo é dividido da seguinte forma. Inicialmente é mostrado o procedimento de Extração de Bézier na seção 5.1. A implementação realizada no programa FAST é apresentada na seção 5.2. Exemplos utilizando o programa de Análise Isogeométrica desenvolvido são apresentados na seção 5.3.

5.1 Extração de Bézier

A Extração de Bézier é um procedimento descrito em Borden et al. (2011) com finalidade de representar as NURBS por uma sequência de representações de Bézier de dimensão equivalente. O procedimento possibilita que os elementos isogeométricos possuam as mesmas funções base, independente da discretização utilizada no vetor de *knots*, necessitando que uma informação adicional seja fornecida para cada elemento. Tal informação é a matriz de extração do elemento $\bar{\mathbf{C}}^e$ de dimensão $(p+1) \times (p+1)$.

A Extração de Bézier consiste em representar as bases B-splines como uma combinação linear de polinômios de Bernstein. As funções de base em cada *knot span* serão iguais aos polinômios de Bernstein, discutidos no Item 3.2. O procedimento é obtido pela aplicação do algoritmo de Decomposição de Bézier descrito em Piegl e Tiller (1997). Este algoritmo consiste em aumentar a multiplicidade de cada *knot* até que seja igual a $p+1$. Entretanto, uma multiplicidade igual a p é suficiente⁵³, a única mudança provocada por isto é que as funções de base das fronteiras internas serão compartilhadas, de modo que cada elemento de Bézier irá compartilhar pontos de controle nas fronteiras. A maneira mais simples de se obter a Decomposição de Bézier é aplicando a operação de *Inserção de Knot* múltiplas vezes.

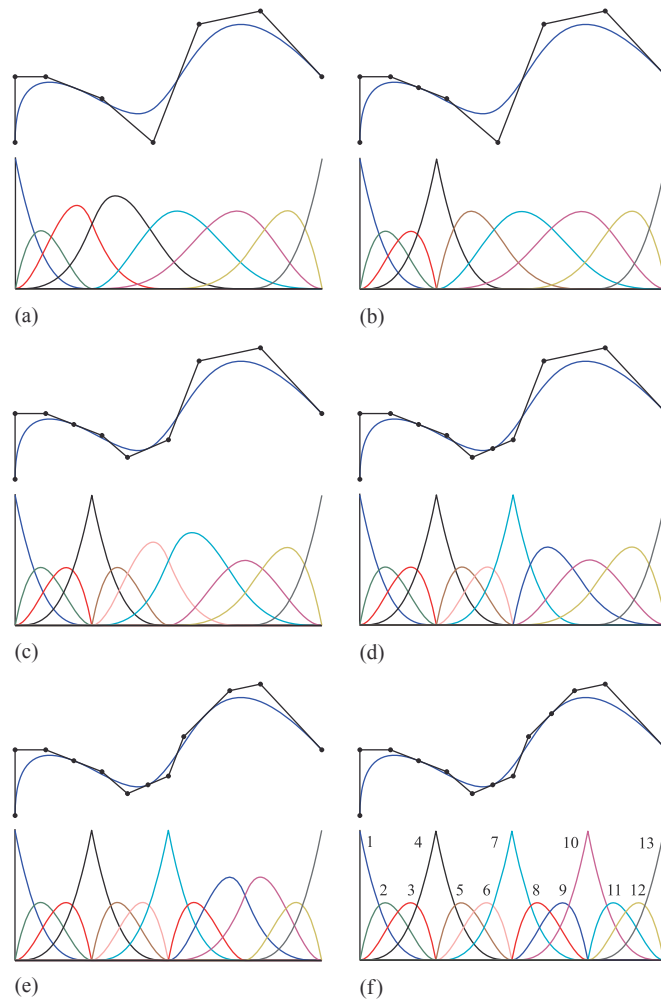
Figura 38 – Exemplo de uma curva B-Spline cúbica com $\Xi = [0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4]$.



Fonte: Borden et al. (2011).

A discussão do algoritmo de Extração de Bézier apresentada a seguir é baseada em Borden et al. (2011). Utilizando a uma curva NURBS cúbica como exemplo (Figura 38), com vetor de *knots* igual a $\Xi = [0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4]$, são inseridos os *knots* $\bar{\Xi} = [1, 1, 2, 2, 3, 3]$ a fim de se realizar a Extração de Bézier na curva. A Figura 39 mostra a sequência de modificações nos pontos de controle e nas funções base B-Spline $N_{i,p}$ provenientes da *Inserção de Knot*. Pôde-se observar que ao final do processo cada *knot span* possui funções base idênticas aos polinômios de Bernstein cúbicas apresentadas na Figura 8.(c). É importante ressaltar que tal procedimento não modifica a geometria da curva, sendo esta representada por outro conjunto de bases, vetor de *knots* e pontos de controle.

Figura 39 – Sequência de modificações nos pontos de controle e funções base obtidas pela inserção dos *knots* $\bar{\Xi} = [1, 1, 2, 2, 3, 3]$, na curva ilustrada pela Figura 38.



Fonte: Borden et al. (2011).

5.1.1 Operador de Extração Bézier

O operador de Extração Bézier é um operador linear que relaciona os pontos de controle antes e depois da *Inserção de Knots*. Considerando uma curva NURBS com vetor de *knots* igual a $\Xi = [\xi_1, \dots, \xi_{n+p+1}]$ tal que deseja-se inserir os *knots* $\bar{\Xi} = [\bar{\xi}_1, \dots, \bar{\xi}_m]$, os pontos de controle antes (\mathbf{p}^n) e depois ($\bar{\mathbf{p}}^{n+1}$) de cada inserção de *knot* podem ser relacionados por:

$$\bar{\mathbf{p}}^{n+1} = (\bar{\mathbf{C}}^n)^T \mathbf{p}^n \quad (114)$$

onde $\bar{\mathbf{C}}^n$ é a matriz de dimensão $(n \times n + 1)$ obtida dos coeficientes α definidos na Equação (46), dada por:

$$\bar{\mathbf{C}}^n = \begin{bmatrix} \alpha_1 & 1 - \alpha_2 & 0 & \dots & & & 0 \\ 0 & \alpha_2 & 1 - \alpha_3 & 0 & \dots & & 0 \\ 0 & 0 & \alpha_3 & 1 - \alpha_4 & 0 & \dots & 0 \\ & & & \ddots & & & \\ & & & & \ddots & & \\ 0 & \dots & & & 0 & \alpha_{(n)} & 1 - \alpha_{(n+1)} \end{bmatrix} \quad (115)$$

Considerando que o processo seja realizado m vezes, pode-se definir a matriz de extração como sendo $\bar{\mathbf{C}}^T = (\bar{\mathbf{C}}^{n+m})^T \cdot (\bar{\mathbf{C}}^{n+m-1}) \dots (\bar{\mathbf{C}}^n)^T$, logo os pontos de controle finais $\bar{\mathbf{p}}^{n+m}$ podem ser relacionados aos pontos de controle iniciais \mathbf{p} por:

$$\bar{\mathbf{p}}^{n+m} = \bar{\mathbf{C}}^T \mathbf{p} \quad (116)$$

Após a aplicação da Extração de Bézier, as funções B-Spline $N_{i,p}(\xi)$ serão iguais as funções Bernstein $B_{i,p}(\xi)$ definidas no intervalo paramétrico que são atuantes. Como a curva permanece a mesma durante todo o processo de *Inserção de Knot* realizado, e definindo as funções base B-Spline pelo vetor $\mathbf{n}(\xi)$ e os polinômios de Bernstein como $\mathbf{b}(\xi)$, obtêm-se:

$$\mathbf{C}(\xi) = (\bar{\mathbf{p}}^{n+m})^T \mathbf{b}(\xi) = (\bar{\mathbf{C}}^T \mathbf{p})^T \mathbf{b}(\xi) = \mathbf{p}^T \bar{\mathbf{C}} \mathbf{b}(\xi) = \mathbf{p}^T \mathbf{n}(\xi) \quad (117)$$

Assim:

$$\mathbf{n}(\xi) = \bar{\mathbf{C}} \mathbf{b}(\xi) \quad (118)$$

onde $\bar{\mathbf{C}}$ é o operador de Extração Bézier, que é uma matriz de dimensão $n \times (n + m)$. É importante notar que para realizar o cálculo da matriz $\bar{\mathbf{C}}$ é necessário apenas as informações do vetor de knots Ξ .

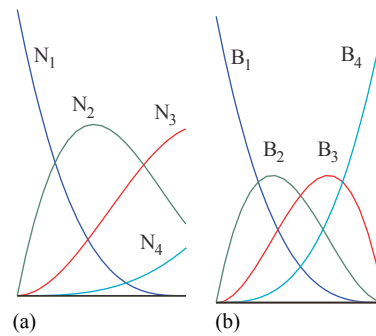
No presente trabalho não foram utilizados os pontos de controle referente a malha de Bézier ($\bar{\mathbf{p}}^{n+m}$). A Extração de Bézier é utilizada apenas para reescrever as funções base de uma forma mais cômoda para implementação dos elementos isogeométricos (*knot span*).

Aplicando a Equação (118) na curva apresentada pela Figura 38, obtêm-se:

$$\begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \\ N_5 \\ N_6 \\ N_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \frac{1}{2} & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{7}{12} & \frac{2}{3} & \frac{1}{3} & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{6} & \frac{1}{3} & \frac{2}{3} & \frac{2}{3} & \frac{1}{3} & \frac{1}{6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{6} & \frac{1}{3} & \frac{2}{3} & \frac{7}{12} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{2} & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \\ B_6 \\ B_7 \\ B_8 \\ B_9 \\ B_{10} \\ B_{11} \\ B_{12} \\ B_{13} \end{bmatrix} \quad (119)$$

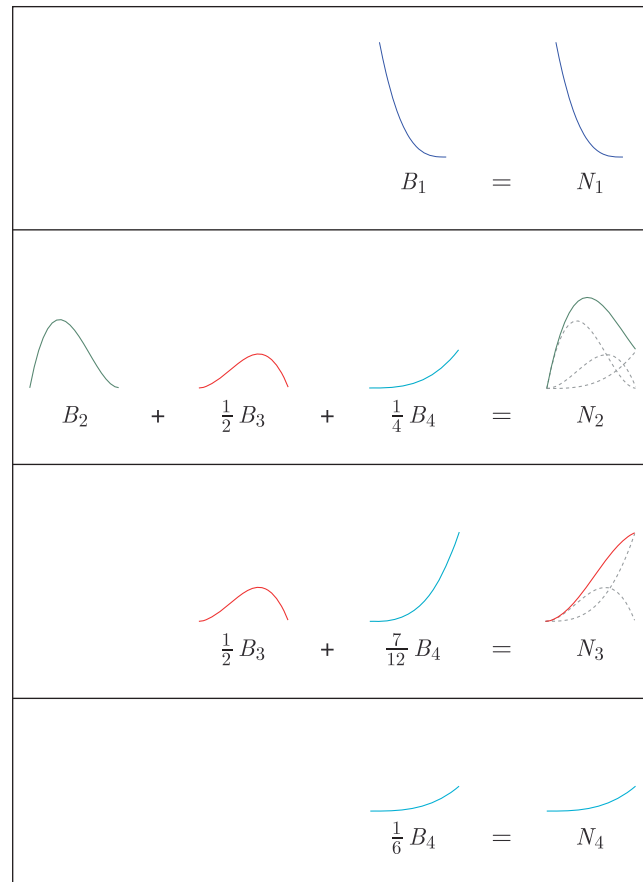
O primeiro intervalo paramétrico $[0,1]$ é calculado pelas bases N_1 , N_2 , N_3 e N_4 . Cada base da B-spline original pode ser calculada por uma combinação linear das funções de bases Bernstein B_1 , B_2 , B_3 e B_4 , como apresentado na Figura 40 e Figura 41.

Figura 40 – Bases B-Spline e Bernstein do primeiro *knot span* da curva dada pela Figura 38.



Fonte: Borden et al. (2011).

Figura 41 – Obtenção das bases B-Spline por combinação linear das bases Bernstein.



Fonte: Borden et al. (2011).

A estrutura da matriz de extração $\bar{\mathbf{C}}$ pode ser utilizada para definir uma matriz de extração $\bar{\mathbf{C}}^e$ para cada *knot span* da NURBS. Fazendo isto na Equação (119), chegamos em:

$$\begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \\ N_5 \\ N_6 \\ N_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \frac{1}{2} & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{7}{12} & \frac{2}{3} & \frac{1}{3} & \frac{1}{6} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{6} & \frac{1}{3} & \frac{2}{3} & \frac{2}{3} & \frac{1}{3} & \frac{1}{6} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{6} & \frac{1}{3} & \frac{2}{3} & \frac{7}{12} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{4} & \frac{1}{2} & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \\ B_6 \\ B_7 \\ B_8 \\ B_9 \\ B_{10} \\ B_{11} \\ B_{12} \\ B_{13} \end{bmatrix} \quad (120)$$

onde cada cor representa um *knot span* das bases B-spline univariantes. As matrizes $\bar{\mathbf{C}}^e$ correspondem às submatrizes marcadas na Equação (120). A equação que relaciona as funções base para cada *knot span* é dada por:

$$\mathbf{n}^e = \bar{\mathbf{C}}^e \mathbf{b}^e \quad (121)$$

onde \mathbf{n}^e são as bases B-spline univariantes do elemento e \mathbf{b}^e são as bases Bernstein do elemento. Aplicando a Equação (121) para todos os *knot spans* da curva tratada na Equação (120), obtemos:

$$\mathbf{n}^1 = \bar{\mathbf{C}}^1 \mathbf{b}^1 \Rightarrow \begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & \frac{1}{2} & \frac{1}{4} \\ 0 & 0 & \frac{1}{2} & \frac{7}{12} \\ 0 & 0 & 0 & \frac{1}{6} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix} \quad (122)$$

$$\mathbf{n}^2 = \bar{\mathbf{C}}^2 \mathbf{b}^2 \Rightarrow \begin{bmatrix} N_2 \\ N_3 \\ N_4 \\ N_5 \end{bmatrix} = \begin{bmatrix} \frac{1}{4} & 0 & 0 & 0 \\ \frac{7}{12} & \frac{2}{3} & \frac{1}{3} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{3} & \frac{2}{3} & \frac{2}{3} \\ 0 & 0 & 0 & \frac{1}{6} \end{bmatrix} \begin{bmatrix} B_4 \\ B_5 \\ B_6 \\ B_7 \end{bmatrix} \quad (123)$$

$$\mathbf{n}^3 = \bar{\mathbf{C}}^3 \mathbf{b}^3 \Rightarrow \begin{bmatrix} N_3 \\ N_4 \\ N_5 \\ N_6 \end{bmatrix} = \begin{bmatrix} \frac{1}{6} & 0 & 0 & 0 \\ \frac{2}{3} & \frac{2}{3} & \frac{1}{3} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{3} & \frac{2}{3} & \frac{7}{2} \\ 0 & 0 & 0 & \frac{1}{4} \end{bmatrix} \begin{bmatrix} B_7 \\ B_8 \\ B_9 \\ B_{10} \end{bmatrix} \quad (124)$$

$$\mathbf{n}^4 = \bar{\mathbf{C}}^4 \mathbf{b}^4 \Rightarrow \begin{bmatrix} N_4 \\ N_5 \\ N_6 \\ N_7 \end{bmatrix} = \begin{bmatrix} \frac{1}{6} & 0 & 0 & 0 \\ \frac{7}{12} & \frac{1}{2} & 0 & 0 \\ \frac{1}{4} & \frac{1}{2} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} B_{10} \\ B_{11} \\ B_{12} \\ B_{13} \end{bmatrix} \quad (125)$$

É importante ressaltar que todos os vetores \mathbf{b}^e são idênticos para todos os elementos do *patch*, independente da discretização considerada.

A matriz de extração do elemento $\bar{\mathbf{C}}^e$ pode ser calculada pelo algoritmo proposto em Borden et al. (2011), ilustrado na Figura 42. Como comentado anteriormente, somente as informações da base B-Spline são necessárias para o cálculo da matriz de extração do elemento $\bar{\mathbf{C}}^e$, estas informações são o vetor de *knots* e o grau da base.

Figura 42 – Algoritmo de cálculo das matrizes de extração $\bar{\mathbf{C}}^e$ univariante de cada elemento.

```

Entrada: Vetor de knots  $\Xi = [\xi_1, \dots, \xi_m]$ ; Número de knots  $m$ ; Grau da base  $p$ .
Saída : Número de elementos  $nb$ ; Operadores de extração  $\mathbf{C}^e$ , com  $e = 1, 2, \dots, nb$ .

// Inicializações:
 $a = p + 1$ ;
 $b = a + 1$ ;
 $nb = 1$ ;
 $\bar{\mathbf{C}}^1 = \mathbf{I}$ ;
enquanto  $b < m$  faça
     $\bar{\mathbf{C}}^{nb+1} = \mathbf{I}$ ; // Inicializa o próximo operador de extração.
     $i = b$ ;

    // Conta a multiplicidade do knot na localidade  $b$ .
    enquanto ( $b < m$  &&  $\Xi(b+1) == \Xi(b)$ ) faça  $b++$ ;
     $mult = b - i + 1$ ;

    se  $mult < p$  então
        // Usa a Equação (46) para avaliar os alfas.
         $number = \Xi(b) - \Xi(a)$ ;
        para  $j = p, p-1, \dots, mult + 1$  faça
            |  $alphas(j - mult) = number / (\Xi(a + j) - \Xi(a))$ 
        fim
         $r = p - mult$ ;
        // Atualiza os coeficientes da matriz para os  $r$  novos knots.
        para  $j = 1, 2, \dots, r$  faça
            |  $save = r - j + 1$ ;
            |  $s = mult + j$ ;
            para  $k = p + 1, p, \dots, s + 1$  faça
                |  $\alpha = alfas(k - s)$ ;
                | // A próxima linha corresponde a Equação (45).
                |  $\bar{\mathbf{C}}^{nb}(:, k) = \alpha \bar{\mathbf{C}}^{nb}(:, k) + (1.0 - \alpha) \bar{\mathbf{C}}^{nb}(:, k - 1)$ ;
            fim
            se  $b < m$  então
                | // Atualiza os coeficientes repetidos do próximo operador.
                |  $\mathbf{C}^{nb+1}(save : j + save, save) = \mathbf{C}^{nb}(p - j + 1 : p + 1, p + 1)$ ;
            fim
        fim
         $nb = nb + 1$  // Finaliza com o operador atual.
        se  $b < m$  então
            | // Atualiza os índices para o próximo operador.
            |  $a = b$ ;
            |  $b++$ ;
        fim
    fim
fim

```

Fonte: Modificado de Borden et al. (2011)

As matrizes de extração em caso de superfícies e sólidos NURBS são obtidas pelo produto tensorial de cada matriz de extração univariante⁵³. Uma maneira mais eficiente de utilizar a Extração Bézier é avaliar as matrizes de extração em cada direção paramétrica: \overline{C}_ξ^e ; \overline{C}_η^e e \overline{C}_ζ^e , utilizar tais matrizes para o cálculo das bases B-Spline univariantes em função dos polinômios de Bernstein e depois continuar o procedimento padrão de avaliação das funções de forma e suas derivadas.

Não é mais necessário realizar o mapeamento dos pontos de Gauss do espaço de referência do elemento para o espaço paramétrico da NURBS (ver Figura 34), uma vez que se defina os polinômios de Bernstein no intervalo de $[-1,1]$ por⁵²:

$$B_{i,p}(\xi) = \frac{1}{2}(1 - \xi)B_{i,p-1}(\xi) + \frac{1}{2}(1 + \xi)B_{i-1,p-1}(\xi) \quad (126)$$

$$B_{i,p}(\xi) = 0 \quad \text{para } i < 1 \quad \text{e } i > p + 1, \quad B_{1,0}(\xi) = 1$$

A partir desta equação, a primeira derivada de $B_{i,p}$ pode ser calculada no intervalo de $[-1,1]$ por:

$$\frac{d}{d\xi}B_{i,p} = \frac{p}{2}(B_{i-1,p-1}(\xi) - B_{i,p-1}(\xi)) \quad (127)$$

Desta forma, todo o procedimento de mapeamento fica encapsulado na matriz de extração do elemento \overline{C}^e .

A utilização da Extração de Bézier permite que a implementação da AIG se torna ainda mais parecida com a implementação do MEF, facilitando a reutilização do código. Uma vez definindo as incidências do elemento e pré-calculando as matrizes de extração de cada elemento \overline{C}^e , a única modificação necessária reside em uma rotina de cálculo das funções de forma e suas derivadas. Considerando o elemento sólido isogeométrico, o cálculo das funções de forma implementadas neste trabalho é apresentada na Figura 43. Nesta rotina de cálculo, as funções *CalcBernBas* e *CalcBernBasDerv* recebem o grau da curva e a coordenada paramétrica (ξ, η, ζ) definidas de $[-1,1]$, e retornam os $p + 1$ polinômios de Bernstein e suas derivadas.

Figura 43 – Rotina para cálculo das funções de forma e suas derivadas para o elemento sólido isogeométrico.

```

Entrada: Coordenada paramétrica  $(\xi, \eta, \zeta)$  do ponto de Gauss no intervalo paramétrico de  $[-1$  a  $1]$ ; Grau do elemento isogeométrico  $(p \times q \times l)$ ; Pesos dos pontos de controle do elemento armazenados num tensor de terceira ordem  $\mathbf{W}^e$ ; Matrizes de extração do elemento em cada direção paramétrica:  $\overline{\mathbf{C}}_\xi^e, \overline{\mathbf{C}}_\eta^e, \overline{\mathbf{C}}_\zeta^e$ .
Saída : Função de forma trivariante  $R(\xi, \eta, \zeta)$  e derivadas parciais  $\frac{\partial}{\partial \xi} R(\xi, \eta, \zeta), \frac{\partial}{\partial \eta} R(\xi, \eta, \zeta), \frac{\partial}{\partial \zeta} R(\xi, \eta, \zeta)$  armazenados respectivamente nos vetores  $\mathbf{R}, \mathbf{dR}_\xi, \mathbf{dR}_\eta$  e  $\mathbf{dR}_\zeta$ , de tamanho  $(p + 1) \times (q + 1) \times (l + 1)$ .

// Cálculo das bases Berntein  $B_{i,p}$  e derivadas  $B'_{i,p}$ 
 $\mathbf{b}_\xi \leftarrow \text{CalcBernBas}(p, \xi), \quad \mathbf{db}_\xi \leftarrow \text{CalcBernBasDerv}(p, \xi);$ 
 $\mathbf{b}_\eta \leftarrow \text{CalcBernBas}(q, \eta), \quad \mathbf{db}_\eta \leftarrow \text{CalcBernBasDerv}(q, \eta);$ 
 $\mathbf{b}_\zeta \leftarrow \text{CalcBernBas}(l, \zeta), \quad \mathbf{db}_\zeta \leftarrow \text{CalcBernBasDerv}(l, \zeta);$ 

// Cálculo das bases B-Spline  $N_{i,p}$  e derivadas  $N'_{i,p}$ 
 $\mathbf{n}_\xi = \overline{\mathbf{C}}_\xi^e \mathbf{b}_\xi, \quad \mathbf{dn}_\xi = \overline{\mathbf{C}}_\xi^e \mathbf{db}_\xi;$ 
 $\mathbf{n}_\eta = \overline{\mathbf{C}}_\eta^e \mathbf{b}_\eta, \quad \mathbf{dn}_\eta = \overline{\mathbf{C}}_\eta^e \mathbf{db}_\eta;$ 
 $\mathbf{n}_\zeta = \overline{\mathbf{C}}_\zeta^e \mathbf{b}_\zeta, \quad \mathbf{dn}_\zeta = \overline{\mathbf{C}}_\zeta^e \mathbf{db}_\zeta;$ 

// Cálculo dos numeradores e denominadores do produto tensorial
 $w = dw\xi = dw\eta = dw\zeta = 0;$ 
 $a = 1;$ 
para  $i = 1$  até  $p+1$  faça
    para  $j = 1$  até  $q+1$  faça
        para  $k = 1$  até  $l+1$  faça
             $\mathbf{R}(a) = \mathbf{n}_\xi(k) \times \mathbf{n}_\eta(j) \times \mathbf{n}_\zeta(i) \times \mathbf{W}_{ijk}^e;$ 
             $\mathbf{dR}_\xi(a) = \mathbf{dn}_\xi(k) \times \mathbf{n}_\eta(j) \times \mathbf{n}_\zeta(i) \times \mathbf{W}_{ijk}^e;$ 
             $\mathbf{dR}_\eta(a) = \mathbf{n}_\xi(k) \times \mathbf{dn}_\eta(j) \times \mathbf{n}_\zeta(i) \times \mathbf{W}_{ijk}^e;$ 
             $\mathbf{dR}_\zeta(a) = \mathbf{n}_\xi(k) \times \mathbf{n}_\eta(j) \times \mathbf{dn}_\zeta(i) \times \mathbf{W}_{ijk}^e;$ 
             $w+ = \mathbf{R}(a);$ 
             $dw\xi+ = \mathbf{dR}_\xi(a);$ 
             $dw\eta+ = \mathbf{dR}_\eta(a);$ 
             $dw\zeta+ = \mathbf{dR}_\zeta(a);$ 
             $a = a + 1;$ 
        fim
    fim
fim
// Realiza a divisão pelos denominadores.
para  $i = 1$  até  $(a-1)$  faça
     $\mathbf{R}(i) = \mathbf{R}(i)/w;$ 
     $\mathbf{dR}_\xi(i) = \mathbf{dR}_\xi(i)/dw\xi;$ 
     $\mathbf{dR}_\eta(i) = \mathbf{dR}_\eta(i)/dw\eta;$ 
     $\mathbf{dR}_\zeta(i) = \mathbf{dR}_\zeta(i)/dw\zeta;$ 
fim

```

Fonte: Elaborada pelo autor.

5.2 Implementação da AIG no programa FAST

O FAST⁵⁴ é um programa acadêmico de Elementos Finitos baseado no programa FEMOOP⁵⁵ e desenvolvido pelo Laboratório de Mecânica computacional e Visualização da Universidade Federal do Ceará. O programa é escrito em linguagem C++ e vem sendo utilizado e expandido em várias dissertações^{18,20,56}.

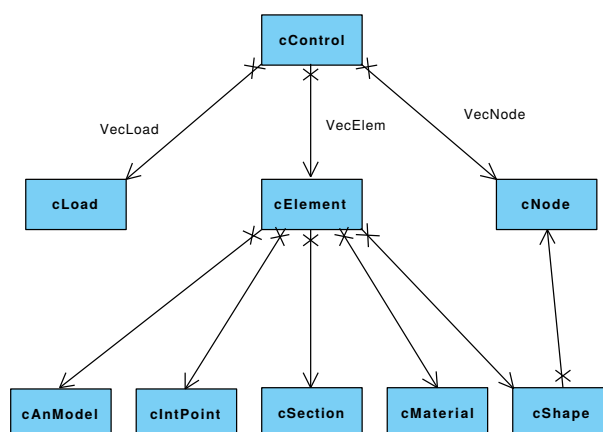
5.2.1 Arquitetura do programa FAST

O programa FAST foi concebido utilizando o Paradigma de Orientação ao Objeto (POO), sua estrutura é composta por classes de objetos, onde cada um destes possui função específica no programa.

Os conceitos de Herança e Polimorfismo são utilizados, de modo que classes com funções semelhantes são organizados em uma hierarquia. Os métodos presentes na hierarquia podem ser reaproveitados, e reimplementados caso seja necessário. A interação entre os objetos é adequadamente encapsulada, de modo que as contribuições realizadas no programa possam ser desenvolvidas de forma segura, evitando que operações ilegais sejam realizadas indevidamente.

A arquitetura geral do programa FAST é apresentada na Figura 44, onde é mostrado a interação entre as principais classes do programa. As setas indicam a interação existente entre as classes, o “x” ao final da seta significa que a classe que sofreu a interação não têm acesso a classe que realizou a ação.

Figura 44 – Diagrama de classes UML⁵⁷ descrevendo a arquitetura geral do programa FAST.

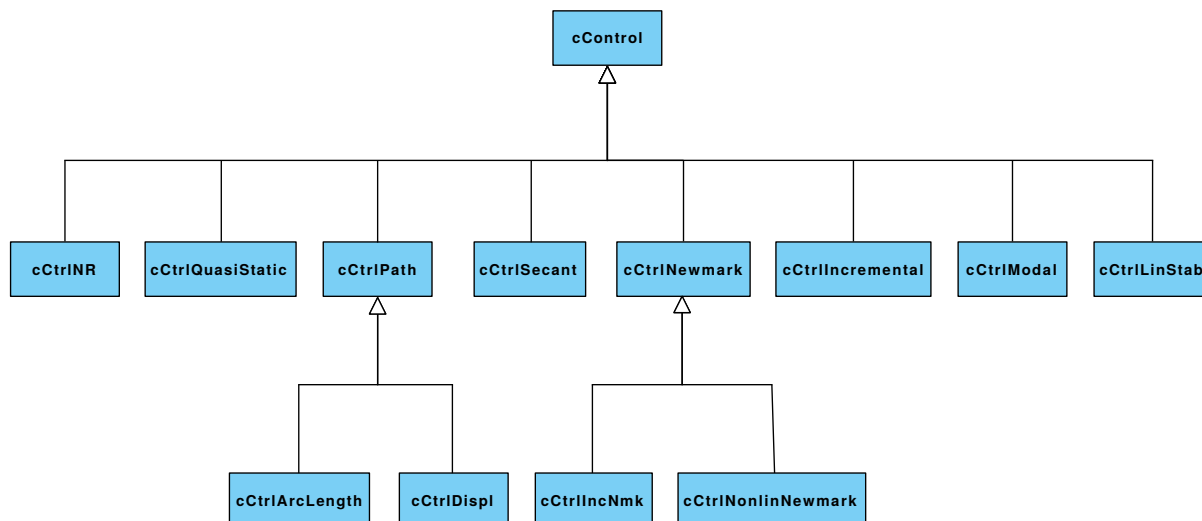


Fonte: Elaborada pelo autor.

A classe *cControl* é responsável por gerenciar os métodos de solução do problema de análise estrutural. Tais métodos de solução são implementados na função virtual *Solver()*. Os diversos métodos de solução são implementados em sub-classes da *cControl*, como a análise estática linear implementada na classe *cLinStat* e a análise estática não-linear utilizando método

de controle de carga (Newton-Rhapson) implementado na classe *cCtrlNR*. A Figura 45 mostra todas as sub-classes da classe *cControl*.

Figura 45 – Diagrama classes UML da classe *cControl*.



Fonte: Elaborada pelo autor.

A classe *cLoad* trata da montagem do vetor de forças externas do problema. A classe *cNode* armazena as coordenadas nodais do modelo. Todos os nós do modelo são alocados em um vetor de nós (*VecNode*).

A classe *cElement* contém as implementações relacionadas aos elementos finitos, como a avaliação da matriz de rigidez do elemento, vetor de forças internas e tensões. A classe consegue lidar com elementos diferentes (barras, elementos bidimensionais e sólidos) de forma genérica. A classe também armazena e dá acesso à objetos necessários para realização destes cálculos, como material (*cMaterial*) e seção (*cSection*).

A classe *cShape* armazena os atributos geométricos dos elementos (incidências nodais), e realiza a avaliação das funções de forma e suas derivadas. A classe *cAnModel* é responsável por gerenciar os aspectos relacionados com as equações diferenciais/variacionais que governam o problema de análise, como o número e tipo dos graus de liberdade em cada nó do elemento e estrutura. A classe *cIntPoint* cria e gerencia os pontos de integração utilizados na análise numérica. Também gerencia o tipo de integração utilizada, como quadratura de Gauss ou Lobato. Uma descrição mais elaborada do programa pode ser encontrada em Dantas Junior (2014), inclusive da parte de Sólidos Laminados.

Toda a estrutura do programa foi aproveitada para implementação computacional da AIG. As modificações realizadas se restringiram à criação de novos sub-tipos da classe *cShape* e de uma nova classe para lidar com as informações referentes aos *patches*, a classe *cPatch*.

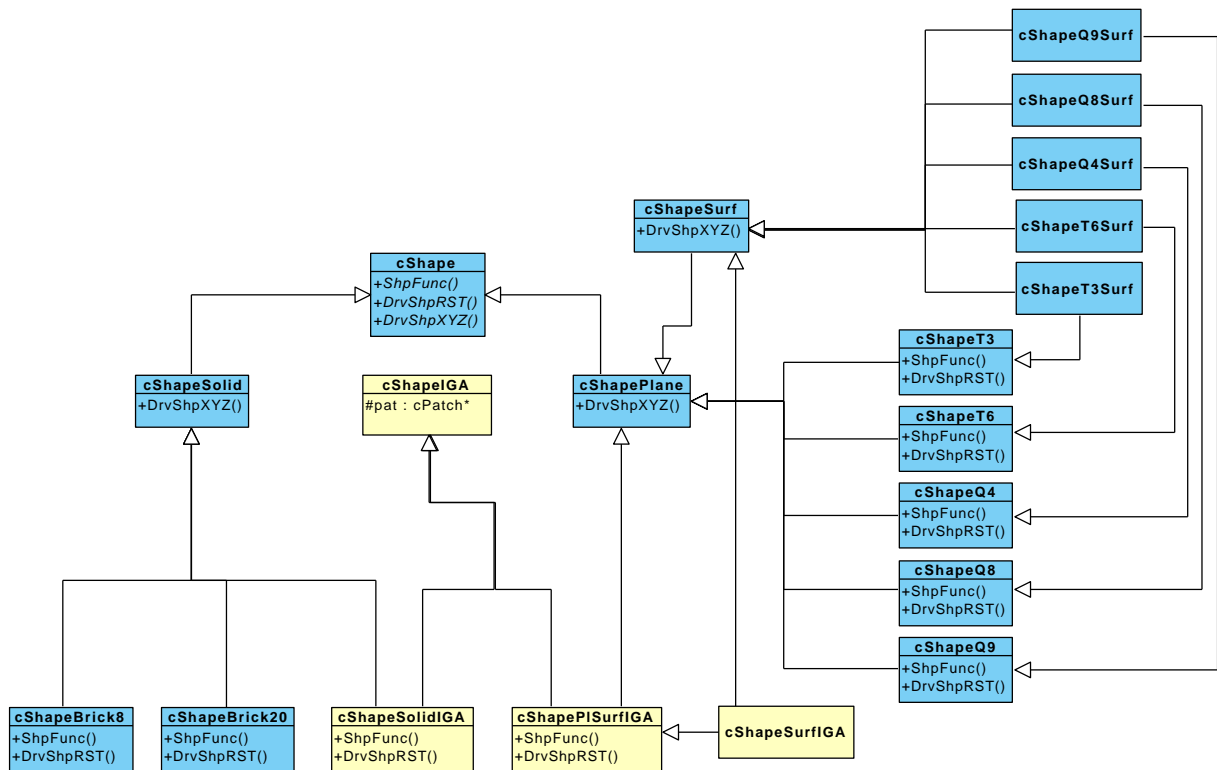
5.2.2 Implementação da classe *cShapeIGA*

A classe *cShape* armazena informações geométricas dos elementos. Tais informações são descritas pelas incidências nodais, sendo estas armazenadas por um vetor de referências (ponteiros) de tipos *cNode*. A classe possui três métodos de grande importância, tais métodos são:

- ShpFunc()*: responsável por calcular o valor das funções de forma do elemento, para uma dada coordenada paramétrica;
- DrvShpRST()*: responsável por avaliar as derivadas das funções de forma no sistema paramétrico do elemento, para uma dada coordenada paramétrica;
- DrvShpXYZ()*: responsável por avaliar as derivadas das funções de forma no sistema físico, para uma dada coordenada paramétrica. Esta função envolve o cálculo do Jacobiano, expresso pela Equação (72), e necessita utilizar o método *DrvShpRST()* em sua implementação.

Uma descrição detalhada da classe *cShape* está apresentada na Figura 46, mostrando várias sub-classes existentes na hierarquia. As classes de cor amarela representam as novas classes implementadas neste trabalho.

Figura 46 – Diagrama classes UML da classe *cShape*.

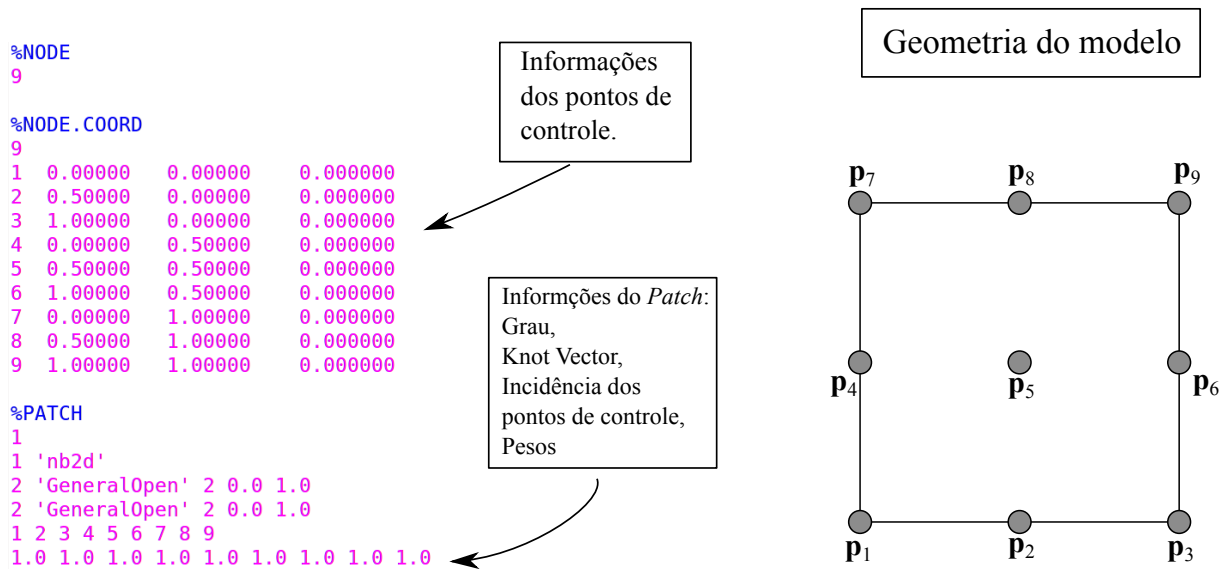


Fonte: Elaborada pelo autor.

A classe *cShapeIGA* é utilizada para armazenar as incidências dos pontos de controle, bem como da acesso ao objeto *patch* associado ao elemento isogeométrico. Os pontos

de controle são armazenados na mesma estrutura de dados utilizada para armazenar os nós de elementos finitos (*VecNode*), inclusive utilizando as mesmas funções de leitura. A incidência do elemento é calculada baseada no *knot span* do elemento, considerando a topologia do *patch* associado ao elemento. A Figura 47 mostra o trecho referente a descrição da geometria de um arquivo de entrada do programa FAST.

Figura 47 – Trecho referente descrição da geometria de um arquivo de entrada do FAST AIG.



Fonte: Elaborada pelo autor.

A estrutura da hierarquia das classes utiliza herança múltipla. Esta estrutura foi elaborada de forma para que não seja necessário a reimplementação dos métodos *DrvShpXYZ*. Estes métodos são implementados nas classes genéricas *cShapeSolid*, *cShapePlane* e *cShapeSurf*. É importante notar que embora este trabalho tenha focado em elementos sólidos, também foram implementados elementos bidimensionais, inclusive de superfícies. Os elementos de superfície são necessários para fazer a integração de cargas distribuídas ao longo da superfície de sólidos NURBS.

A classe *cShapeSolidIGA* é responsável pelo cálculo das funções de forma e suas derivadas no espaço paramétrico da NURBS, tais tarefas são realizadas respectivamente pelos métodos *ShpFunc()* e *DrvShpRST()*. Estes métodos são realizados pelo algoritmo descrito na Figura 43.

5.2.3 Implementação da classe *cPatch*

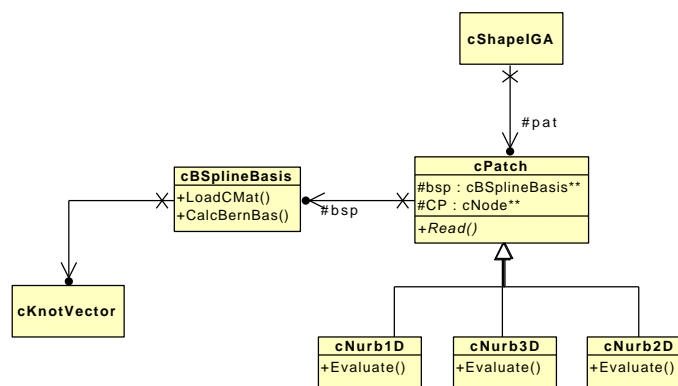
A classe *cPatch* é utilizada para armazenar dados e implementar métodos relacionados a NURBS. Três sub-classes foram implementadas para tratar curvas, superfícies e sólidos NURBS. Apenas as incidências dos pontos de controle do *patch* são armazenadas, uma vez que estes pontos de controle já tenham sido instanciados e armazenados pela classe *cNode*, como

foi mostrado na Figura 47. O método *Evaluate()* avalia a NURBS para uma dada coordenada paramétrica.

A classe *cBsplineBasis* trata dos dados utilizados para cálculo das funções de base B-spline. A classe *cKnotVector* armazena os valores dos *knot vectors* considerados. Os objetos *B-Spline* também são responsáveis por avaliar e armazenar as matrizes de extração \bar{C}^e de cada *knot span*. Esta tarefa é realizada pelo método *LoadCMat()* após a leitura do objeto *cBsplineBasis*, utilizando o algoritmo descrito na Figura 42.

A diagrama contendo as classes *cBsplineBasis*, *cKnotVector*, *cPatch* e suas subclasses está apresentado na Figura 48.

Figura 48 – Diagrama classes UML da classe *cPatch*.



Fonte: Elaborada pelo autor.

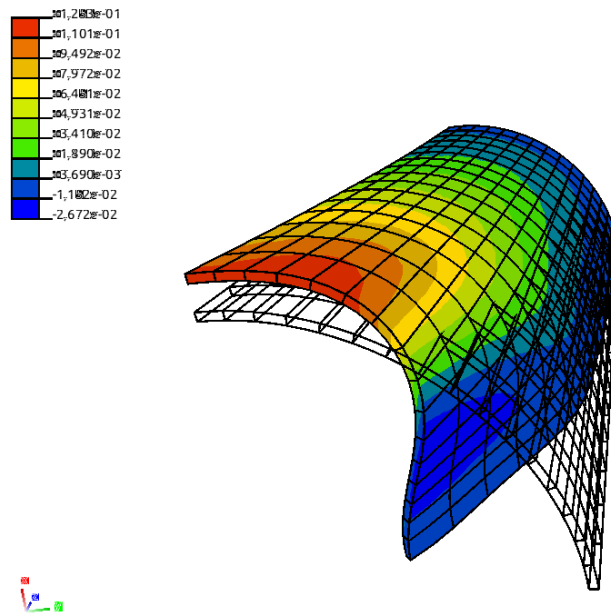
Vale lembrar que praticamente nenhuma alteração foi realizada na classe *cElement* e suas derivadas. O elemento sólido laminado desenvolvido em Dantas Junior (2014) foi obtido em sua versão isogeométrica quase que imediatamente após a implementação das novas classes e métodos descritos anteriormente.

5.2.4 Implementação do Pós-processamento

O pós-processamento da análise isogeométrica requer que um pós-processador adaptado à análise isogeométrica seja utilizado. Uma forma de aproveitar um programa de pós-processamento utilizado no MEF é gerar uma malha fictícia de elementos finitos.

Os *knot spans* do *patch* podem ser utilizados para gerar elementos finitos fictícios. As coordenadas nodais dos elementos são calculadas em função da coordenada paramétrica do *knot span*. Por exemplo, um BRICK8 pode ser gerado de qualquer *patch* considerando que os nós sejam avaliados pelas coordenadas de extremidade do *knot span*. As tensões devem ser avaliadas nas mesmas coordenadas paramétricas onde estarão os nós fictícios. A Figura 49 mostra um exemplo de análise isogeométrica sendo visualizado através de um pós-processador de elementos finitos.

Figura 49 – Exemplo de resultados da AIG visualizado através de um pós-processador de elementos finitos.



Fonte: Elaborada pelo autor.

5.2.5 Paralelização na montagem da matriz de rigidez

Uma das contribuições deste trabalho foi implementar a paralelização da montagem da matriz de rigidez no programa *FAST*. Os elementos de sólido laminado apresentados neste trabalho possuem um elevado custo computacional devido à integração utilizada (uma quadratura completa por lâmina). Por este motivo foi implementado um procedimento de paralelização da montagem da matriz de rigidez global da estrutura a fim de diminuir o tempo de análise dos exemplos estudados.

Este procedimento foi realizado utilizando a biblioteca OpenMP⁵⁸. A Figura 50 mostra o trecho do código existente na classe *cControl* responsável por realizar a etapa de Montagem (*Assembly*) da Matriz de rigidez global. O trecho do código pode ser facilmente paralelizado adicionando as diretivas `"#pragma omp parallel for"`. O comando `"firstprivate(elm,Kelm)"` é utilizado para que cada *thread* possua uma variável *elm* e *Kelm* privadas durante o *loop*. A linha `"#pragma omp critical"` garante que cada matriz de rigidez do elemento seja adicionada sequencialmente na matriz de rigidez global.

Figura 50 – Código em linguagem C++ da etapa de montagem da matriz de rigidez global paralelizada no programa FAST.

```

// Declara e inicializa a matriz de rigidez do elemento.

int maxdof = cElement :: GetMaxDof( );
int nelm = cElement :: GetNumElm( );

cMatrix Kelm(maxdof,maxdof);
cElement* elm = 0;

#pragma omp parallel for firstprivate(elm,Kelm)
for (int i = 0; i < nelm; i++)
{
    // Avalia a matriz de rigidez do elemento.

    elm = cElement :: GetElm(i);
    elm->StiffMat(Kelm);

    // Adiciona na matriz de rigidez global do problema.

    #pragma omp critical
    elm->AddGlobMat(Kelm, K);
}

```

Fonte: Elaborada pelo autor.

É importante notar que apesar da simplicidade do código apresentado, o mesmo requer que as rotinas paralelizadas sejam *thread-safe*. Este termo significa que a manipulação das estruturas de dados compartilhadas, nos trechos em paralelo, deve ser realizada de modo a garantir uma correta execução do trecho paralelizado por múltiplos *threads* simultaneamente.

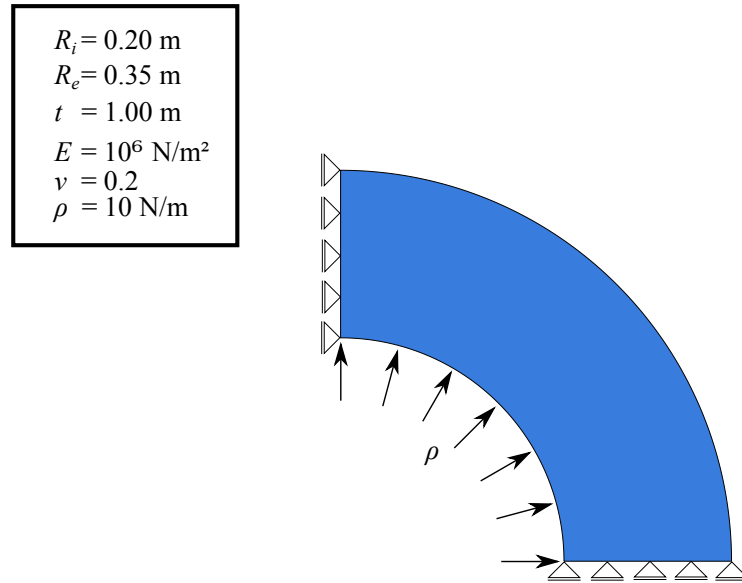
5.3 Exemplos Numéricos

Exemplos numéricos foram realizados para validar os resultados obtidos pelo programa FAST com resultados presentes na Literatura. Serão abordados exemplos lineares e não-lineares geométricos, considerando material isotrópico e compósito laminado. Também será realizado um estudo sobre o efeito da paralelização implementada no programa.

5.3.1 Cilindro espesso submetido à pressão interna

O primeiro exemplo trata da análise linear de um cilindro espesso submetido a pressão interna, considerando modelo 2D de estado plano de tensões (Apesar das formulações tratadas neste trabalho terem sido realizadas para elementos sólidos, pode-se facilmente obter a formulação para problemas planos). A Figura 51 mostra os dados utilizados no exemplo.

Figura 51 – Descrição do Exemplo 5.3.1.



Fonte: Elaborada pelo autor.

A solução analítica deste problema (Solução de Lamé) é dada em coordenadas polares por:

$$u(r) = \left(\frac{1-\nu}{E} \frac{R_i^2}{R_e^2 - R_i^2} r + \frac{1+\nu}{E} \frac{R_i^2 R_e^2}{(R_e^2 - R_i^2) r} \right) \rho \quad r \in [R_i, R_e] \quad (128)$$

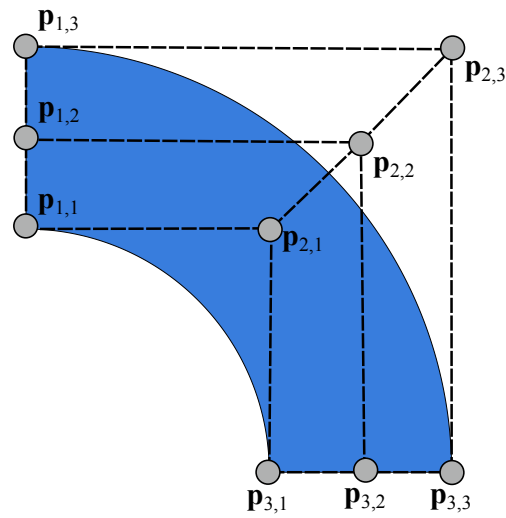
$$\sigma_r(r) = -\rho \frac{(R_e^2 - r^2) R_i^2}{(R_e^2 - R_i^2) r^2} \quad (129)$$

$$\sigma_\theta(r) = -\rho \frac{(R_e^2 + r^2) R_i^2}{(R_e^2 - R_i^2) r^2} \quad (130)$$

onde $u(r)$ é o deslocamento radial, R_i é o raio interno do cilindro, R_e é o raio externo do cilindro, σ_r é a tensão radial e σ_θ é a tensão circunferencial.

O problema é solucionado através de uma Análise Isogeométrica considerando uma superfície plana NURBS quadrática nas duas direções. A Figura 52 mostra a malha de controle da geometria básica do problema. A Tabela 3 mostra as coordenadas e pesos dos pontos de controle da geometria básica do problema.

Figura 52 – Malha de pontos de controle da geometria básica de um quarto de cilindro.



Fonte: Elaborada pelo autor.

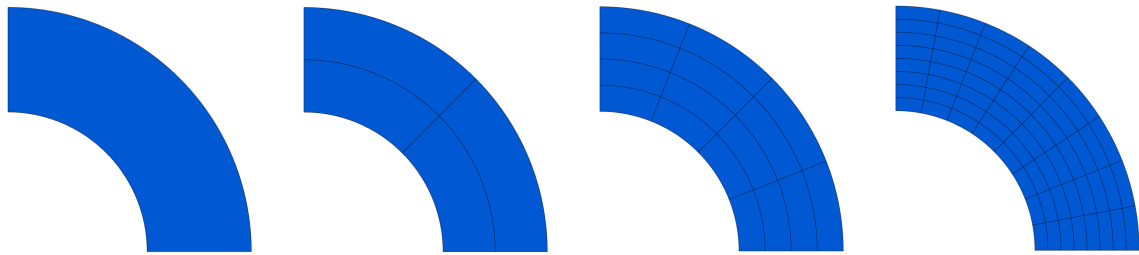
Tabela 3 – Pontos de controle da geometria básica de um quarto de cilindro.

$\mathbf{p}_{i,j}$	x	y	w
$\mathbf{p}_{1,1}$	0.00000	0.20000	1.00000
$\mathbf{p}_{2,1}$	0.20000	0.20000	0.70710
$\mathbf{p}_{3,1}$	0.20000	0.00000	1.00000
$\mathbf{p}_{1,2}$	0.00000	0.27500	1.00000
$\mathbf{p}_{2,2}$	0.27500	0.27500	0.70710
$\mathbf{p}_{3,2}$	0.27500	0.00000	1.00000
$\mathbf{p}_{1,3}$	0.00000	0.35000	1.00000
$\mathbf{p}_{2,3}$	0.35000	0.35000	0.70710
$\mathbf{p}_{3,3}$	0.35000	0.00000	1.00000

Fonte: Elaborada pelo autor.

Foram utilizadas quatro malhas, refinadas através do refinamento h . O refinamento foi realizado inserindo *knots* igualmente espaçados no espaço paramétrico. As malhas são mostradas na Figura 53.

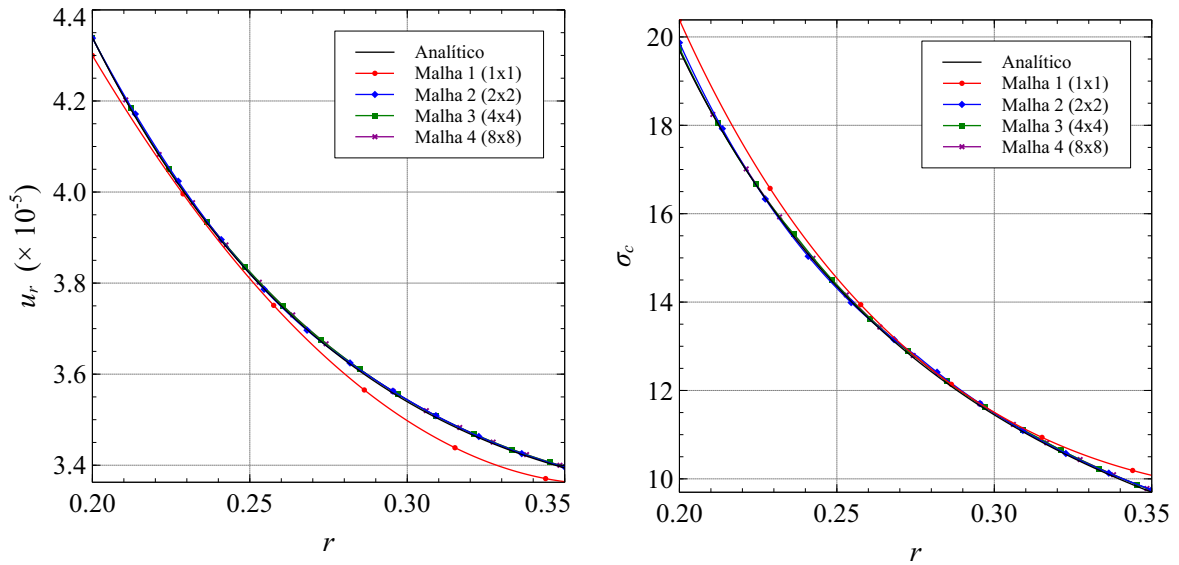
Figura 53 – Malhas de 1-4 utilizadas no Exemplo 5.3.1.



Fonte: Elaborada pelo autor.

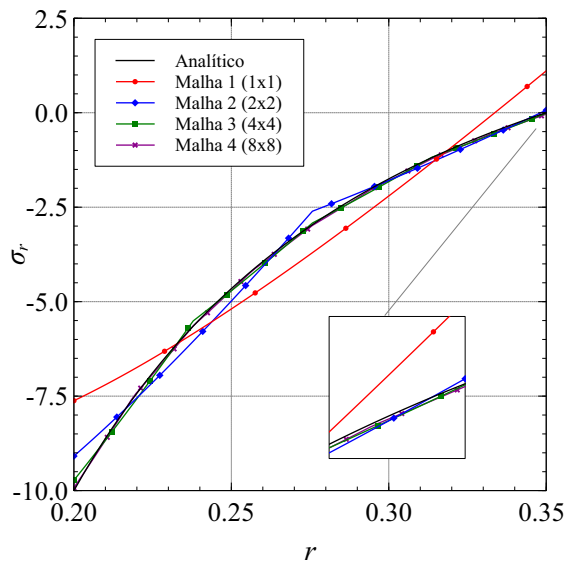
A integração numérica foi realizada considerando quadratura de Gauss completa. Neste trabalho, a quadratura completa é aquela capaz de integrar exatamente o polinômio de maior grau que aparece na matriz de rigidez de um elemento não distorcido (jacobiano constante). Vale lembrar que as NURBS são racionais, logo não é garantido que uma quadratura completa seja suficiente para uma integração exata. Existem outros trabalhos que abordam técnicas mais sofisticadas de integração numérica para Análise Isogeométrica⁵⁹, o uso de tais técnicas está fora do escopo deste trabalho.

A Figura 54 mostra os resultados obtidos com refinamento h . Todas as respostas convergiram para solução exata do problema com a discretização do modelo. As respostas de deslocamento radial e tensão circunferencial apresentaram maior facilidade para convergência. A malha (8x8) apresentou bons resultados para todas as respostas.

Figura 54 – Resultados utilizando refinamento h .

a) Deslocamento radial.

b) Tensão circunferencial.

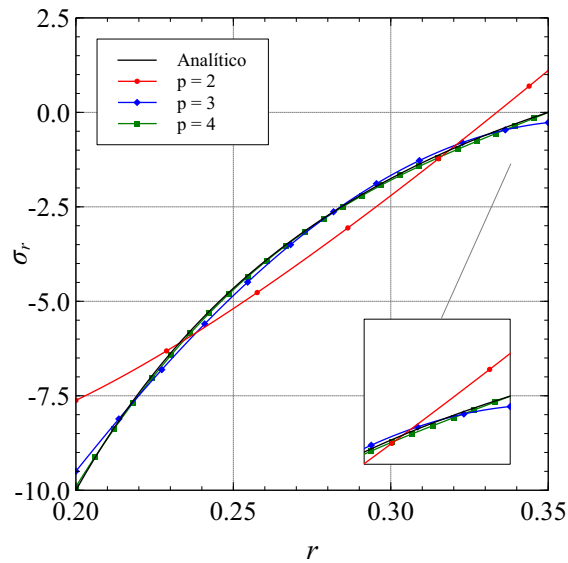


c) Tensão radial.

Fonte: Elaborada pelo autor.

O mesmo exemplo foi realizado utilizando refinamento p , neste caso a geometria básica do modelo foi modificada através do algoritmo de Elevação de Grau. Vale lembrar que este tipo de discretização mantém o número de elementos iniciais constante (igual a um), porém adiciona novos graus de liberdade e eleva o grau destes. A Figura 55 mostra os resultados da tensão radial para o refinamento p realizado.

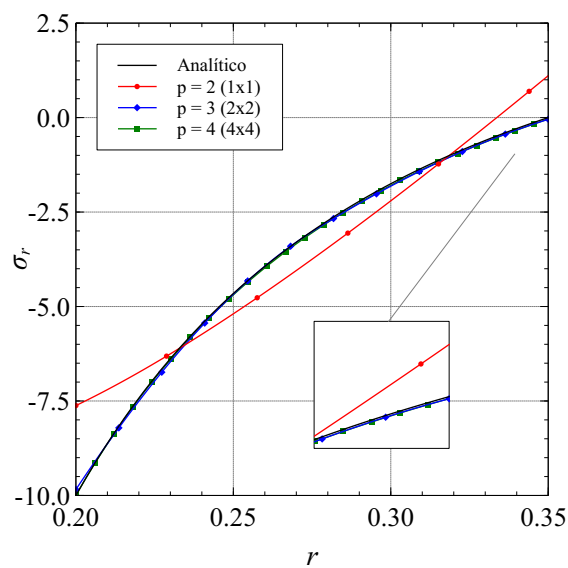
Figura 55 – Resultados referentes à tensão radial utilizando refinamento p .



Fonte: Elaborada pelo autor.

Finalmente, o mesmo exemplo é realizado utilizando uma estratégia de refinamento k , neste caso a geometria básica do modelo tem seu grau elevado e então são inseridos novos *knots* igualmente espaçados. É importante notar que a continuidade entre os elementos internos ao *patch* aumenta. Caso o procedimento fosse aplicado em ordem inversa (ou seja, primeiro a inserção de *knots* e depois a elevação de grau), seria gerado um modelo com continuidade entre os elementos iguais a 1 independente do grau final do *patch*. A Figura 56 mostra os resultados da tensão radial utilizando o refinamento k abordado.

Figura 56 – Resultados referentes à tensão radial utilizando refinamento k .



Fonte: Elaborada pelo autor.

Em todos os refinamentos foram obtidos respostas em concordância com a solução analítica do problema.

5.3.2 Cilindro fino submetido à pressão interna

Este exemplo trata da análise de um cilindro fino engastado nas extremidades submetido a pressão interna, como ilustrado na Figura 57. A solução analítica do campo de deslocamentos radial $u(x)$ foi obtida por Timoshenko e Woinowsky-Krieger (1959):

$$u(x) = -\frac{PR^2}{Et} (1 - C_1 \operatorname{sen}\beta x \operatorname{senh}\beta x - C_2 \cos \beta x \operatorname{cosh}\beta x) \quad x \in \left[-\frac{L}{2}, \frac{L}{2}\right] \quad (131)$$

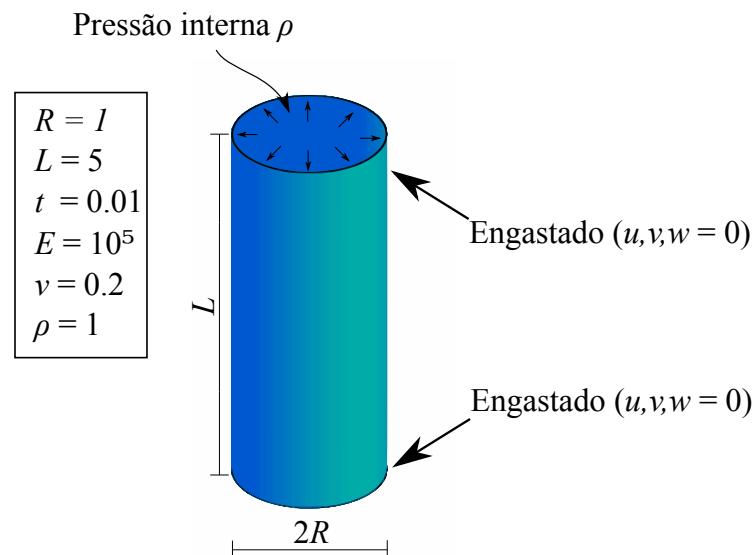
$$C_1 = \frac{\operatorname{sen}\alpha \operatorname{cosh}\alpha - \cos \alpha \operatorname{senh}\alpha}{\operatorname{senh}\alpha \operatorname{cosh}\alpha + \operatorname{sen}\alpha \cos \alpha} \quad (132)$$

$$C_2 = \frac{\cos \alpha \operatorname{senh}\alpha + \operatorname{sen}\alpha \operatorname{cosh}\alpha}{\operatorname{senh}\alpha \operatorname{cosh}\alpha + \operatorname{sen}\alpha \cos \alpha} \quad (133)$$

$$\beta = \left(\frac{Et}{4R^2D}\right)^{\frac{1}{4}}; \quad \alpha = \frac{\beta L}{2}; \quad D = \frac{Et^3}{12(1-\nu^3)} \quad (134)$$

A utilização desta solução requer que o módulo de Young e coeficiente de Poisson sejam substituídos respectivamente por $\frac{E}{1-\nu^2}$ e $\frac{\nu}{1-\nu}$, para consideração do engastamento.

Figura 57 – Descrição do Exemplo 5.3.2.

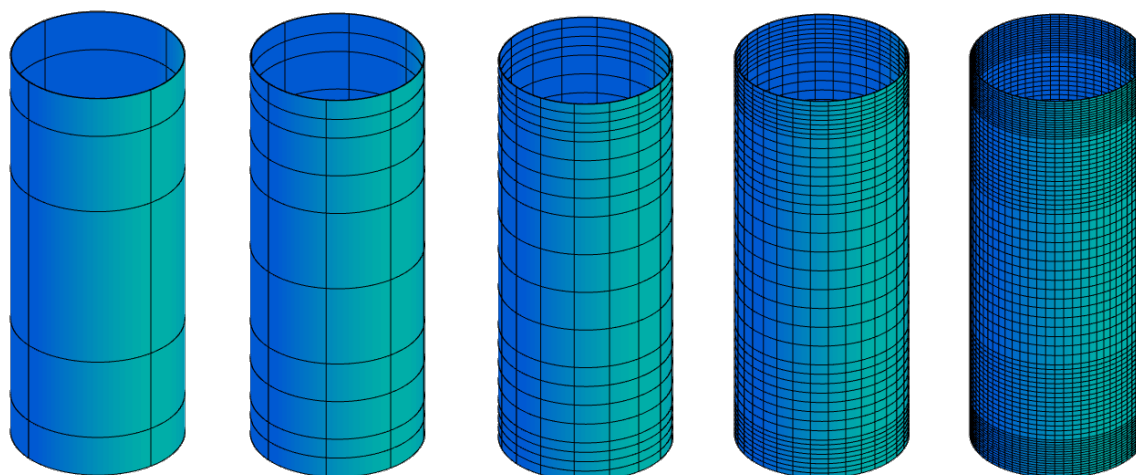


Fonte: Elaborada pelo autor.

O problema é solucionado através de uma Análise Isogeométrica considerando sólido NURBS. A malha inicial (Malha 1) utilizada possui 4 divisões ao longo da circunferência e 5 divisões ao longo do comprimento, priorizando a região próxima ao apoio, como ilustrado

na Figura 58. Malhas mais refinadas são obtidas dobrando o número de subdivisões existente em cada região da malha inicial. O número de divisões na espessura do modelo permanece igual a um para todas as malhas. É importante lembrar que as malhas isogeométricas são definidas pelos *knot spans* presentes na NURBS.

Figura 58 – Malhas de 1-5 utilizadas no Exemplo 5.3.2.

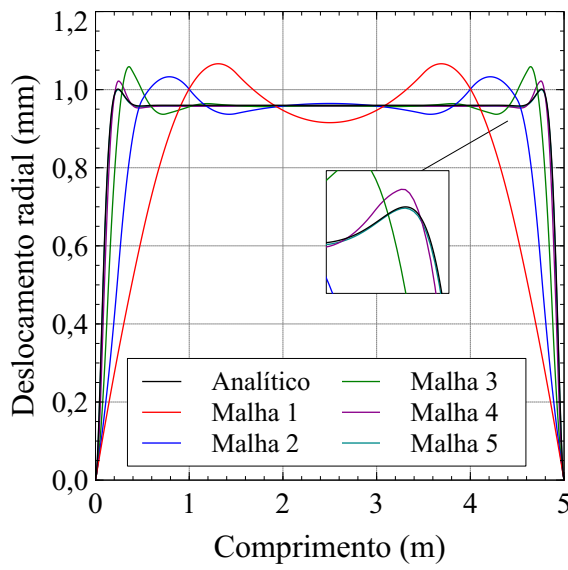


Fonte: Elaborada pelo autor.

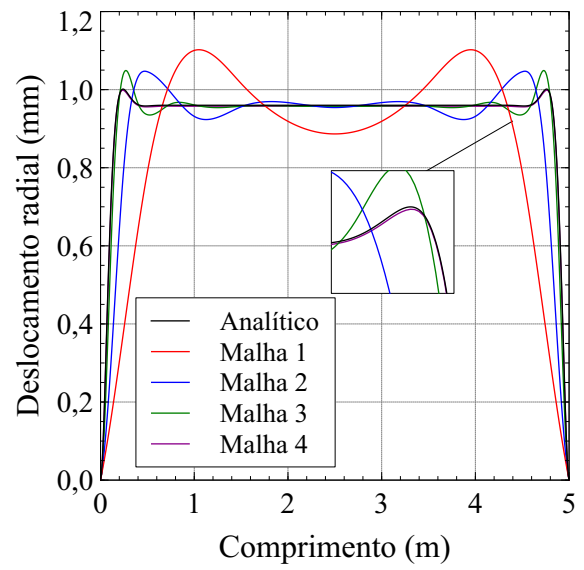
Foram realizadas análises considerando NURBS de diferentes graus, através do refinamento k . O grau quadrático é inicialmente considerado ao longo de todas as direções paramétricas da NURBS, o refinamento k é realizado de modo a elevar o grau da NURBS exceto na direção da espessura, onde o grau permanece quadrático ao longo do processo. A integração dos elementos isogeométricos é realizada sempre com uma quadratura de Gauss completa, utilizando o número de pontos de integração necessário para integrar o maior grau em todas as direções (e.g. $3 \times 3 \times 3$ para o caso quadrático). Todos os exemplos deste capítulo consideram quadratura de Gauss completa na sua integração numérica.

Os campos de deslocamentos obtidos pelas análises isogeométricas estão apresentados na Figura 59. As respostas conseguem uma boa aproximação da solução analítica a medida que são utilizadas malhas mais refinadas. A utilização de elementos de grau mais elevado melhora a velocidade de convergência solução, de modo que a solução analítica possa ser obtida utilizando malhas menos refinadas. A Figura 60 mostra a deformada do modelo obtida considerando diferentes fatores de escala.

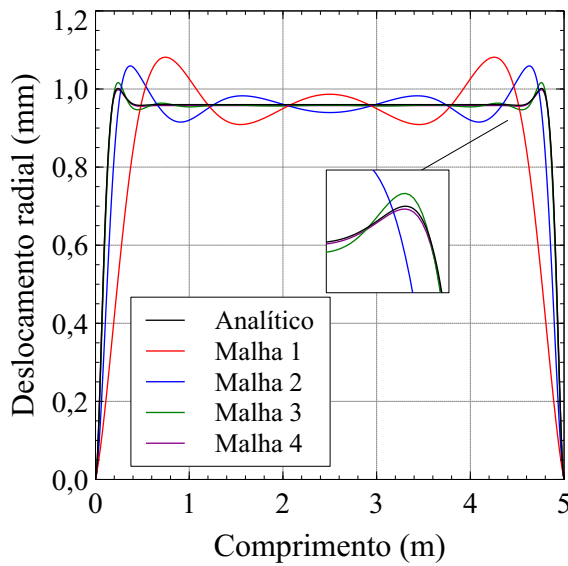
Figura 59 – Deslocamento radial considerando diferentes graus.



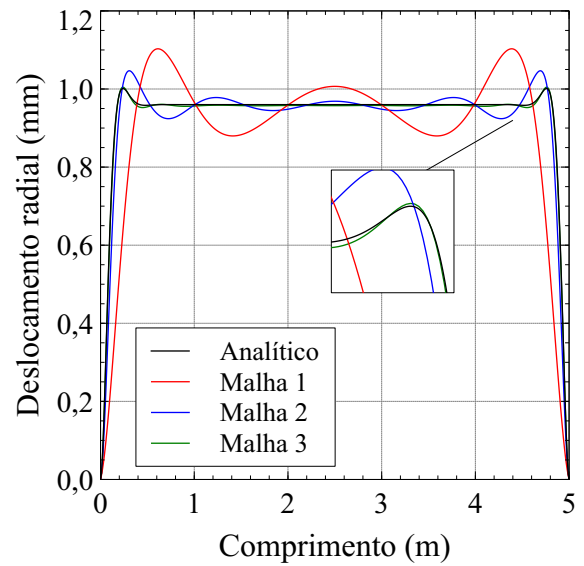
a) Quadrática.



b) Cúbica.



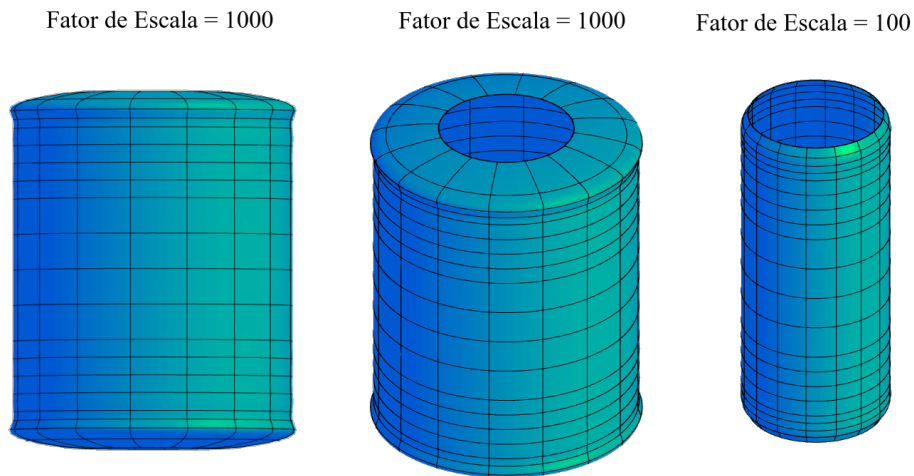
c) Quártica.



d) Quíntica.

Fonte: Elaborada pelo autor.

Figura 60 – Deformada do Exemplo 5.3.2.

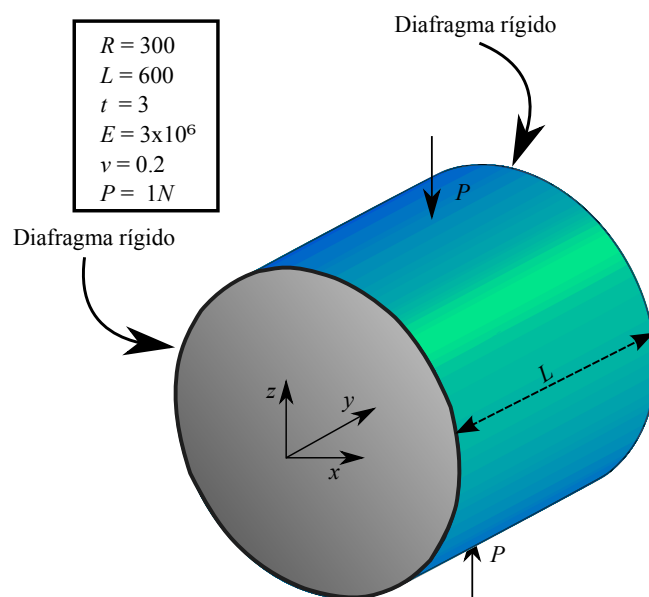


Fonte: Elaborada pelo autor.

5.3.3 Cilindro fino com carga concentrada no centro

Este exemplo trata de um cilindro fino restrito nos extremos por um diafragma rígido e submetido a forças pontuais iguais e opostas localizadas no centro do comprimento, como apresentado na Figura 61. Este problema teste faz parte do conhecido *Shell Obstacle Course*, que consiste em um conjunto de problemas teste de grande relevância em análise de cascas. O mesmo problema foi solucionado utilizando AIG com elementos sólidos por Hughes et al. (2005).

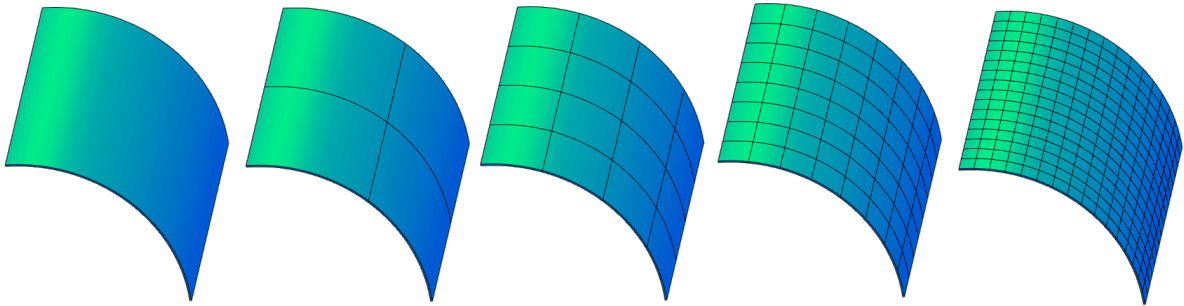
Figura 61 – Descrição do Exemplo 5.3.3.



Fonte: Elaborada pelo autor.

A solução numérica considerou a simetria do problema, de forma que apenas um oitavo do cilindro foi modelado. As malhas utilizadas seguem um mesmo padrão dobrando o número de elementos ao longo da direção do comprimento e da circunferência do modelo, como ilustrado na Figura 62. O número de elementos na espessura é sempre igual a um.

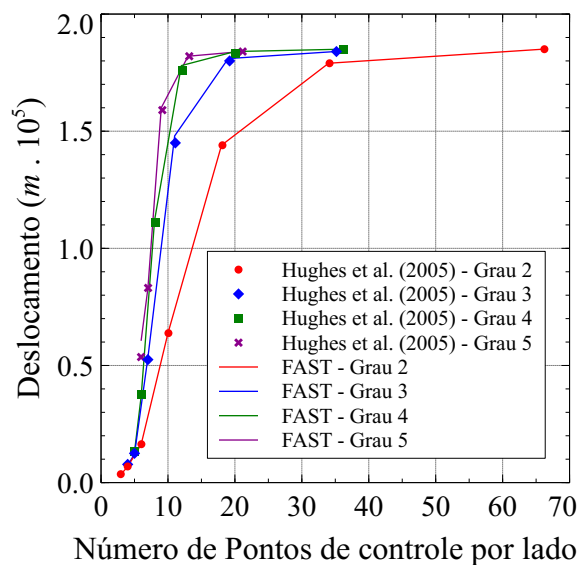
Figura 62 – Malhas 1-5 utilizadas no Exemplo 5.3.3.



Fonte: Elaborada pelo autor.

Os modelos analisados consideraram diferentes graus de NURBS, de maneira análoga ao Exemplo 5.3.2. O deslocamento w_p no ponto de aplicação de carga é comparado com o número de pontos de controle por lado da superfície. A Figura 63 mostra os resultados obtidos no presente trabalho comparando com os resultados de Hughes et al. (2005). Os pontos de cada curva representam cada malha utilizada.

Figura 63 – Convergência de w_p considerando NURBS de vários graus.

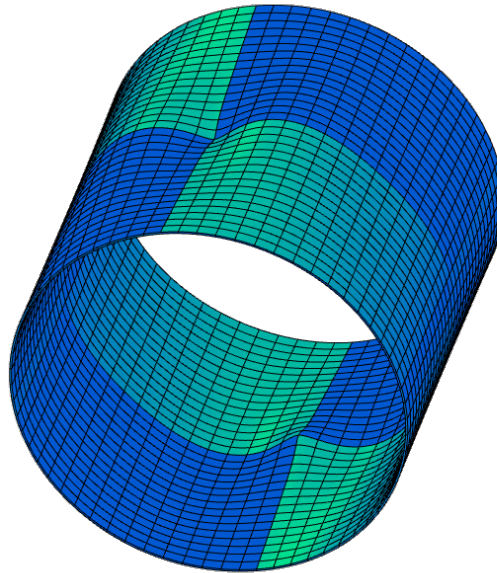


Fonte: Elaborada pelo autor.

Os resultados obtidos apresentaram boa concordância com os resultados de Hughes et al. (2005). O número de pontos de controle por lado necessário para a convergência do

problema diminuiu a medida são utilizados elementos de grau mais elevado. A deformada da estrutura está apresentada na Figura 64.

Figura 64 – Deformada com fator de escala = $2e6$.



Fonte: Elaborada pelo autor.

Um exemplo de mesma geometria foi solucionado por Reddy (2004) considerando material compósito laminado, utilizando elementos finitos de casca. As propriedades do material são apresentadas na Tabela 4. Quatro casos de laminados anti-simétricos são considerados: cross-ply com 2 e 10 lâminas: $[0^\circ 90^\circ]$; $[0^\circ 90^\circ]_5$, angle-ply com 2 e 10 lâminas: $[-45^\circ 45^\circ]$ e $[-45^\circ 45^\circ]_5$.

Tabela 4 – Propriedades do material compósito.

E_1	$E_2 = E_3$	$\nu_{12} = \nu_{21} = \nu_{23}$	$G_{12} = G_{21}$	G_{23}
3e6 (psi)	1.2e6 (psi)	0.25	6e4 (psi)	2.4e4 (psi)

Fonte: Elaborada pelo autor.

O problema foi modelado utilizando a formulação AIG implementado nesta trabalho, considerando o cilindro completo (sem simetria, considerando 8 *patches*) e utilizando uma malha equivale às Malha 5 e Malha 6 deste exemplo com grau cúbico. Os resultados do deslocamento w_p e das tensões σ_{xx} e σ_{yy} no ponto de aplicação da carga estão apresentados na Tabela 5 e Tabela 6. A tensão σ_{xx} é avaliada no topo do laminado, enquanto que a tensão σ_{yy} é avaliada na superfície inferior do laminado.

Tabela 5 – Resultados considerando pares de lâminas [0° 90°].

Núm. Lâminas	Resultado	Reddy	FAST M4	Difer (%)	FAST M5	Difer(%)
2	w_b	-1.3833e-4	-1.3741e-4	-0.673	-1.3978e-4	1.036
	σ_{xx}	4.9837e-1	4.6170e-1	-7.942	6.6223e-1	24.744
	σ_{yy}	-6.1343e-1	-6.3046e-1	2.701	-8.0145e-1	23.460
10	w_b	-8.2278e-5	-8.1777e-5	-0.612	-8.2997e-5	0.867
	σ_{xx}	4.1030e-1	3.8251e-1	-7.265	5.0082e-1	18.074
	σ_{yy}	-4.7351e-1	-4.6118e-1	-2.674	-5.7693e-1	17.926

Fonte: Elaborada pelo autor.

Tabela 6 – Resultados considerando pares de lâminas [-45° 45°].

Núm. Lâminas	Resultado	Reddy	FAST M4	Difer (%)	FAST M5	Difer(%)
2	w_b	-1.4517e-4	-1.4355e-4	-1.125	-1.4588e-4	0.492
	σ_{xx}	2.7389e-1	2.4533e-1	-11.643	3.4514e-1	20.645
	σ_{yy}	-3.2651e-1	-3.1605e-1	-3.309	-3.9502e-1	17.343
10	w_b	-8.1922e-5	-8.1110e-5	-1.002	-8.2385e-5	0.562
	σ_{xx}	1.9657e-1	1.8289e-1	-7.481	2.3490e-1	16.318
	σ_{yy}	-2.3219e-1	-2.1890e-1	-6.072	-2.6289e-1	11.678

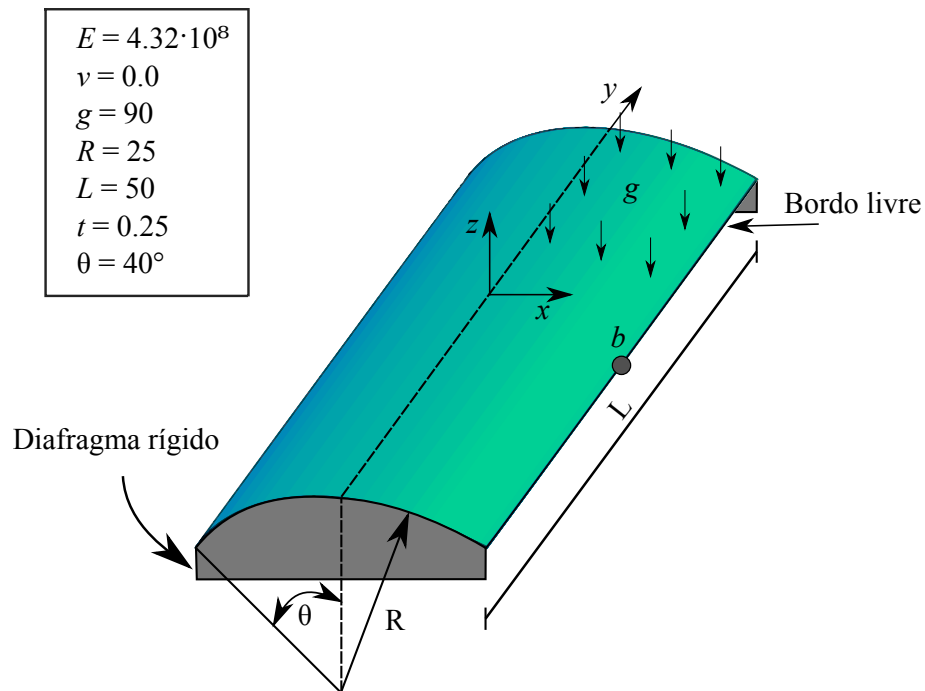
Fonte: Elaborada pelo autor.

Os resultados do deslocamento w_p apresentam concordância com os resultados de Reddy (2004). As tensões apresentaram diferenças crescentes à medida que as malhas foram refinadas. Esta diferença deve está associada a uma falta de discretização da malha utilizada em Reddy (2004), pois o mesmo não realizou um estudo de discretização para as tensões, apenas para o deslocamento w_p .

5.3.4 Casca cilíndrica com carga distribuída (Scordelis-Lo)

Este exemplo consiste em uma casca cilíndrica curva restrita nas extremidades por um diafragma rígido e carregada por uma carga vertical de peso próprio, como mostra a Figura 65. O carregamento g é distribuído pela área superficial da casca. Este problema teste conhecido como *Scordelis-Lo roof* também faz parte do *Shell Obstacle Course*. O problema também foi resolvido por Hughes et al. (2005)¹⁴ através da AIG.

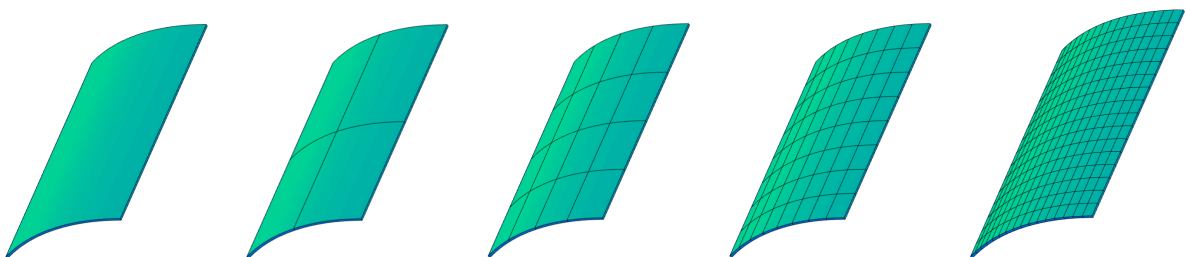
Figura 65 – Descrição do Exemplo 5.3.4



Fonte: Elaborada pelo autor.

A simetria do problema foi considerada na análise, de modo que apenas um quarto da casca foi modelada. A mesma estratégia de discretização das malhas descrita no Exemplo 5.3.3 foi utilizada neste exemplo. A Figura 66 ilustra as malhas utilizadas. Os modelos analisados consideraram elementos com NURBS de diferentes graus.

Figura 66 – Malhas 1-5 utilizadas no Exemplo 5.3.4.

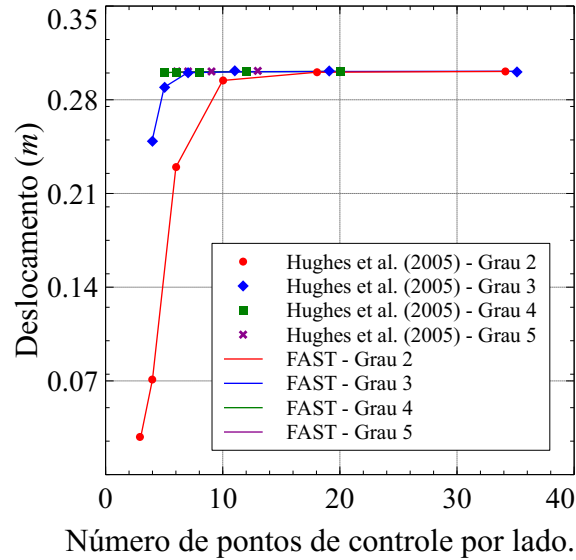


Fonte: Elaborada pelo autor.

Os deslocamentos do ponto b (w_b) obtidos neste trabalho e por Hughes et al. (2005) são apresentados na Figura 67. Os resultados apresentaram uma rápida convergência da resposta e uma excelente concordância com os resultados de Hughes et al. (2005). A velocidade de convergência das respostas foi maior neste exemplo, em comparação com o exemplo anterior. Isto se deve à acentuada descontinuidade ocasionada pela aplicação de uma carga concentrada,

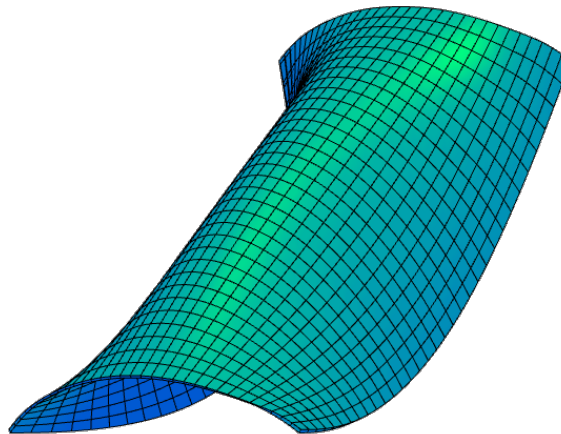
o que não ocorre neste exemplo por se tratar de uma carga distribuída. A deformada obtida nas análises está ilustrada no Figura 68.

Figura 67 – Resultados obtidos do Exemplo 5.3.4.



Fonte: Elaborada pelo autor.

Figura 68 – Deformada da casca Scordelis-Lo (Malha 5 com $p = 4$), com fator de escala = 40.



Fonte: Elaborada pelo autor.

Um exemplo similar foi analisado por Reddy (2004) considerando material compósito. A geometria do exemplo é multiplicada por 12, resultando em $R = 300$, $L = 600$ e $t = 3$. A carga $q = 0.625$. As propriedades do material compósito são o mesmos utilizado no Exemplo 5.3.3. O problema foi modelado sem consideração de simetria (apenas um *patch*), com grau quártico e utilizando a malha equivalente a malha 4 e malha 5 deste exemplo. Os resultados do deslocamento w_b e das tensões σ_{xx} e σ_{yy} no centro da casca estão apresentados na Tabela 7 e

Tabela 8. A tensão σ_{xx} é avaliada no topo do laminado, enquanto que a tensão σ_{yy} é avaliada na superfície inferior do laminado.

Tabela 7 – Resultados do Exemplo 5.3.4 considerando pares de lâminas $[0^\circ 90^\circ]$.

Núm. Lâminas	Resultado	Reddy	FAST M4	Difer (%)	FAST M5	Difer(%)
2	w_b	19.4944	19.4884	-0,031	19.4936	-0.004
	σ_{xx}	-279.750	-282.053	0.817	-281.959	0.783
	σ_{yy}	-3315.93	3305.42	-0.318	-3304.98	-0.332
10	w_b	11.7981	11.7903	-0.066	11.7916	-0.055
	σ_{xx}	110.813	108.245	-2.372	108.313	-2.308
	σ_{yy}	-3390.31	-3381.11	-0.318	-3379.15	-0.330

Fonte: Elaborada pelo autor.

Tabela 8 – Resultados do Exemplo 5.3.4 considerando pares de lâminas $[-45^\circ 45^\circ]$.

Núm. Lâminas	Resultado	Reddy	FAST M4	Difer (%)	FAST M5	Difer(%)
2	w_b	29.9744	29.6382	-1.134	29.7274	-0.831
	σ_{xx}	1839.56	1834.04	-0.301	1840.91	0.073
	σ_{yy}	-1891.06	-1876.71	-0.765	-1877.77	-0.708
10	w_b	15.1519	14.8430	-2.081	14.8525	-2.016
	σ_{xx}	1549.13	1531.24	-1.168	1533.02	-1.051
	σ_{yy}	-1207.31	-1183.37	-2.023	-1182.40	-2.107

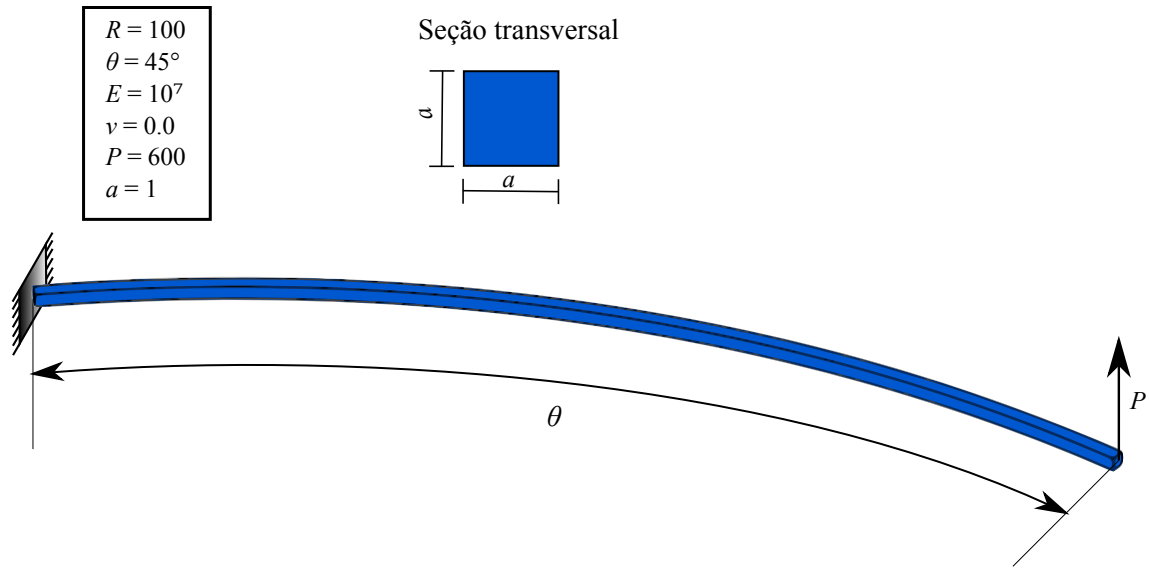
Fonte: Elaborada pelo autor.

Os resultados apresentam excelente concordância com Reddy (2004). A maior diferença percentual obtida foi da ordem de 2%. Esta diferença pode ser devida ao cisalhamento, uma vez que este é considerado de forma aproximada em elementos de casca.

5.3.5 Viga curva

Este exemplo trata de uma viga curva engastada e submetida a um carregamento na sua extremidade, como ilustrado na Figura 69. Este exemplo foi solucionado por Bathe e Bolourchi (1979) e Mororó (2013), utilizando elementos finitos não-lineares de viga e sólidos.

Figura 69 – Descrição do Exemplo 5.3.5.



Fonte: Elaborada pelo autor.

O modelo isogeométrico pode ser descrito exatamente por um sólido de bésier racional, considerando os pontos de controle apresentados na Tabela 9. É importante notar que tal sólido é construído pelo produto de um arco de 45° na direção ξ e duas retas nas direções η e ζ .

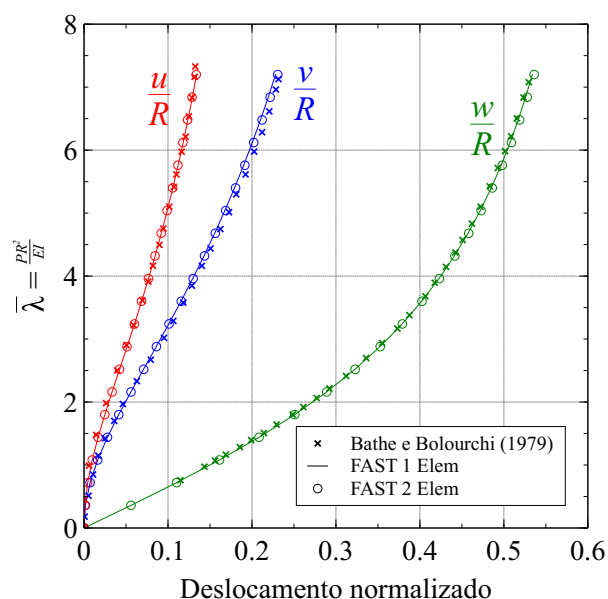
Tabela 9 – Pontos de controle da geometria básica da viga curva.

$\mathbf{p}_{i,j,k}$	x	y	z	w
$\mathbf{p}_{1,1,1}$	0.0000	99.5000	-0.5000	1.0000
$\mathbf{p}_{2,1,1}$	41.2140	99.5000	-0.5000	0.9239
$\mathbf{p}_{3,1,1}$	70.3670	70.3570	-0.5000	1.0000
$\mathbf{p}_{1,2,1}$	0.0000	100.5000	-0.5000	1.0000
$\mathbf{p}_{2,2,1}$	41.6280	100.5000	-0.5000	0.9239
$\mathbf{p}_{3,2,1}$	71.0640	71.0640	-0.5000	1.0000
$\mathbf{p}_{1,1,2}$	0.0000	99.5000	0.5000	1.0000
$\mathbf{p}_{2,1,2}$	41.2140	99.5000	0.5000	0.9239
$\mathbf{p}_{3,1,2}$	70.3570	70.3570	0.5000	1.0000
$\mathbf{p}_{1,2,2}$	0.0000	100.5000	0.5000	1.0000
$\mathbf{p}_{2,2,2}$	41.6280	100.5000	0.5000	0.9239
$\mathbf{p}_{1,2,2}$	71.0640	71.0640	0.5000	1.0000

Fonte: Elaborada pelo autor.

A análise não-linear geométrica é realizada considerando dois elementos de grau $(4 \times 2 \times 2)$. A Figura 70 mostra as curvas de carga \times deslocamento, para os deslocamentos u , v e w na ponta da viga. O fator de carga e os deslocamentos da estrutura são considerados de

Figura 70 – Curvas carga x deslocamento obtidas da análise da viga curva.



Fonte: Elaborada pelo autor.

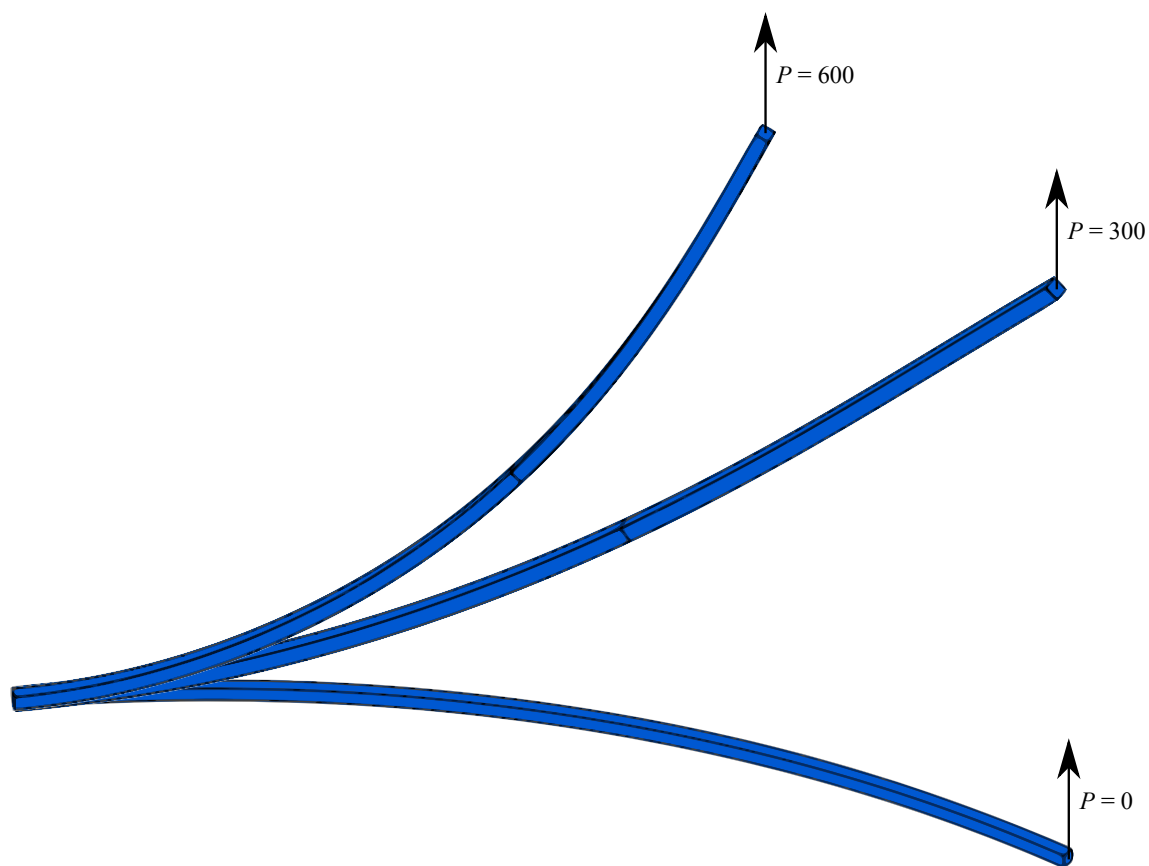
forma normalizada:

$$\bar{\lambda} = \frac{PR^2}{EI} \quad (135)$$

onde $\bar{\lambda}$ é o fator de carga normalizado, P é a carga considerada e I é o momento de inércia da seção da viga. Os deslocamentos são normalizados dividindo tais valores por R . Também é mostrado os resultados de Bath e Bolourchi (1979), estes resultados foram obtidos com 16 elementos sólidos.

Pode-se verificar que os resultados apresentam excelente concordância com a literatura. Foi possível obter respostas de alta qualidade com poucos elementos, considerando um problema que certamente necessitaria de uma adequada discretização via Elementos Finitos para modelar a geometria do problema de forma satisfatória. As deformadas do modelo considerando a configuração deformada e os carregamentos $P = 300$ e $P = 600$ estão apresentadas na Figura 71.

Figura 71 – Deformada Curved Beam.

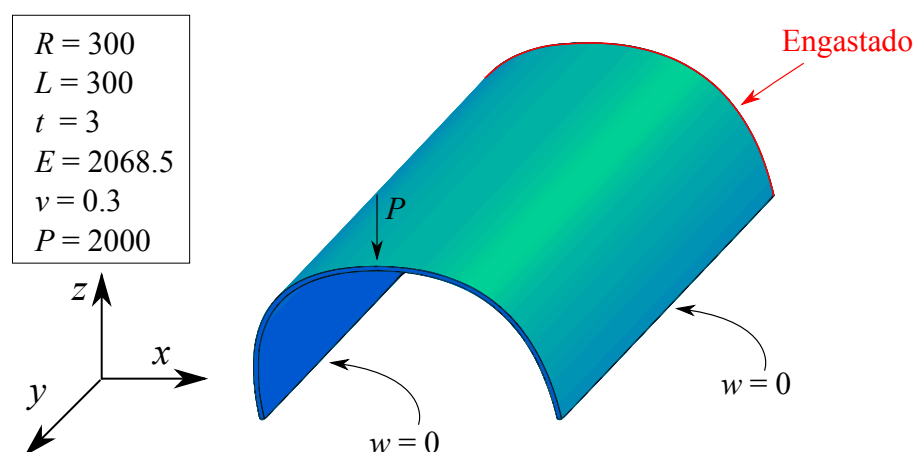


Fonte: Elaborada pelo autor.

5.3.6 Semicilindro engastado

Este exemplo consiste em um semicilindro engastado, submetido a uma carga concentrada em sua extremidade, como mostra a Figura 72. O problema foi solucionado utilizando elementos finitos de casca por Sze et al. (2004) e elementos finitos Sólidos por Dantas Junior (2014).

Figura 72 – Descrição do Exemplo 5.3.6.



Fonte: Elaborada pelo autor.

A simetria do problema foi considerada em sua modelagem, de modo que apenas metade do cilindro foi considerado. Foram utilizadas duas malhas de grau $(4 \times 4 \times 2)$. As malhas possuem 16×16 elementos e 20×20 elementos.

O exemplo foi analisado considerando material isotrópico e material compósito laminado, considerando laminações *cross-ply*. É importante notar que o uso da simetria do problema no caso do compósito é possível devido a laminação *cross-ply* considerada. As propriedades do material isotrópico são apresentadas na Figura 72. As propriedades do material compósito são apresentadas na Tabela 10.

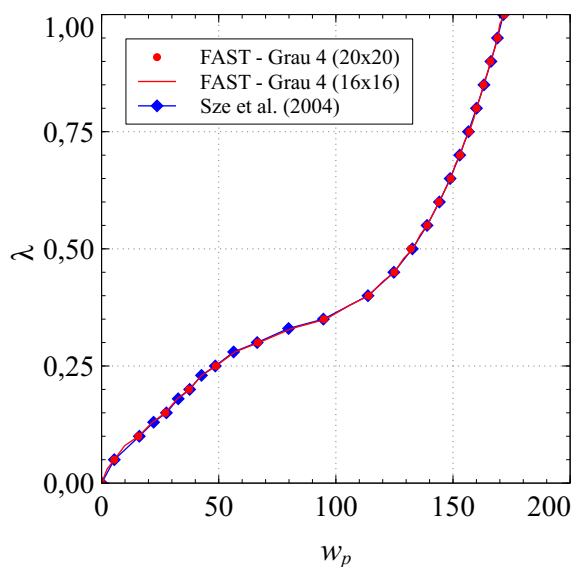
Tabela 10 – Propriedades elásticas do compósito considerado no Exemplo 5.3.6.

E_1	E_2	E_3	ν_{12}	ν_{13}	ν_{23}	G_{12}	G_{13}	G_{23}
2068.500	517.125	517.125	0.3	0.3	0.3	795.600	198.894	198.894

Fonte: Elaborada pelo autor.

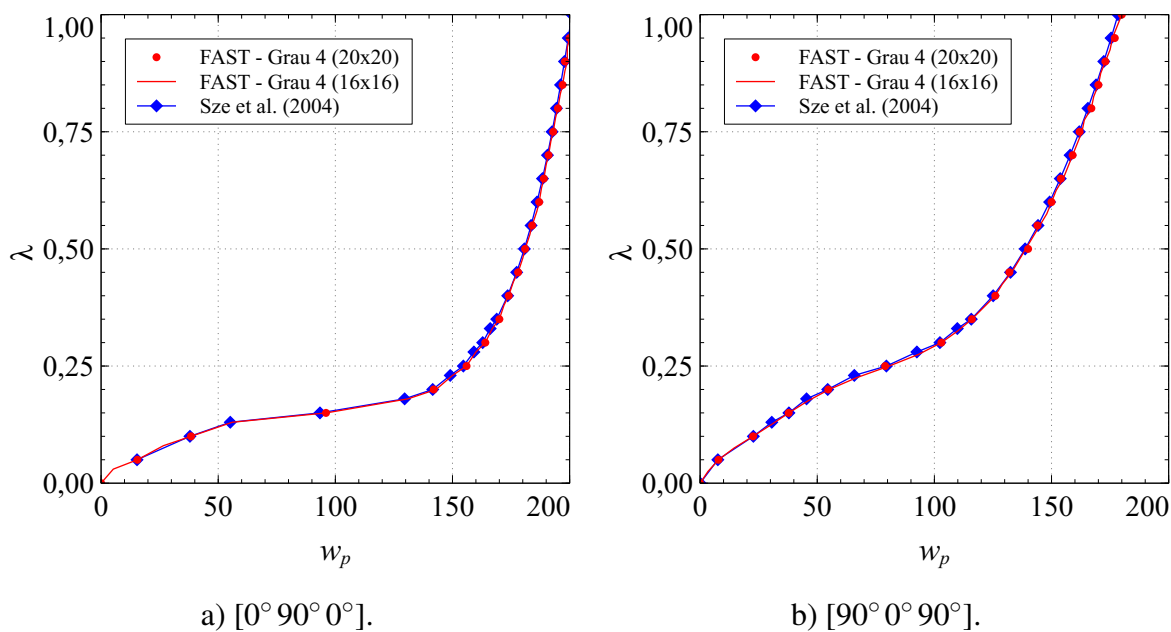
Dois esquemas de laminação foram utilizados: $[90^\circ 0^\circ 90^\circ]$ e $[0^\circ 90^\circ 0^\circ]$. O laminado está alinhado com o eixo do cilindro. As curvas carga \times deslocamento foram obtidas considerando o deslocamento w no ponto de aplicação da carga (w_p). Os resultados são apresentadas nas Figuras 74 e 75. Também são mostrados os resultados obtidos por Sze et al (2004).

Figura 73 – Curva carga \times deslocamento considerando material isotrópico.



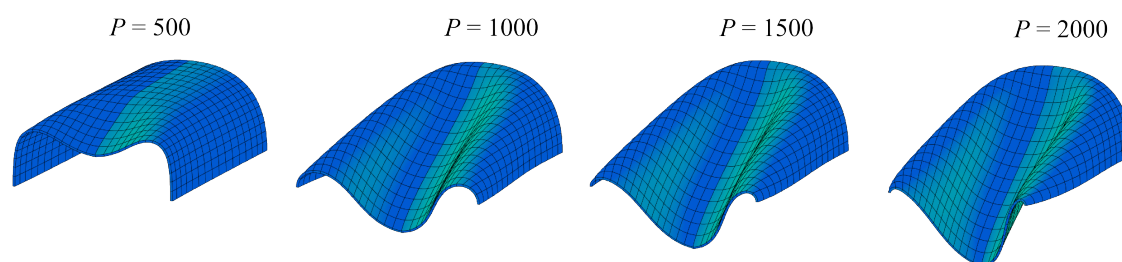
Fonte: Elaborada pelo autor.

Figura 74 – Curva carga \times deslocamento considerando material compósito.



Fonte: Elaborada pelo autor.

Os resultados obtidos apresentam excelente concordância com a literatura. As deformadas da estrutura com laminação $[0^\circ 90^\circ 0^\circ]$ para diferentes carregamentos estão apresentadas na Figura 75.

Figura 75 – Deformada semicilindro engastado $[0^\circ 90^\circ 0^\circ]$.

Fonte: Elaborada pelo autor.

5.3.7 Estudo do efeito da paralelização

O objetivo deste exemplo é mostrar o efeito da estratégia de paralelização abordada no item 5.2.5. Este estudo foi realizado considerando o exemplo anterior (semicilindro engastado) com carga concentrada igual a 50 e considerando análise linear geométrica. O material utilizado é o compósito apresentado na Tabela 10 com laminação de $[0^\circ 90^\circ 0^\circ]$. As análises foram realizadas em um notebook com processador Intel Core I7 de velocidade 2.4 GHz (4 núcleos de processamento).

Um estudo de discretização foi realizado considerando modelos com diferentes graus, sendo que na espessura (direção ζ) foi mantido o grau quadrático. Foi utilizada integração total considerando quadratura de gauss do maior grau da NURBS (e.g. $3 \times 3 \times 3$ para o caso quadrático e $4 \times 4 \times 4$ para o caso cúbico). A Tabela 11 mostra os resultados obtidos, mostrando o deslocamento w no ponto de aplicação da carga (w_p) e também a diferença percentual obtida em comparação com a malha anterior de mesmo grau (δ_{ant}). O objetivo é avaliar qual malha, para cada grau, é capaz de fornecer uma resposta precisa do problema analisado. As malhas foram refinadas até obter diferença percentual menor que 1%.

Tabela 11 – Estudo de discretização realizado no Exemplo 5.3.7.

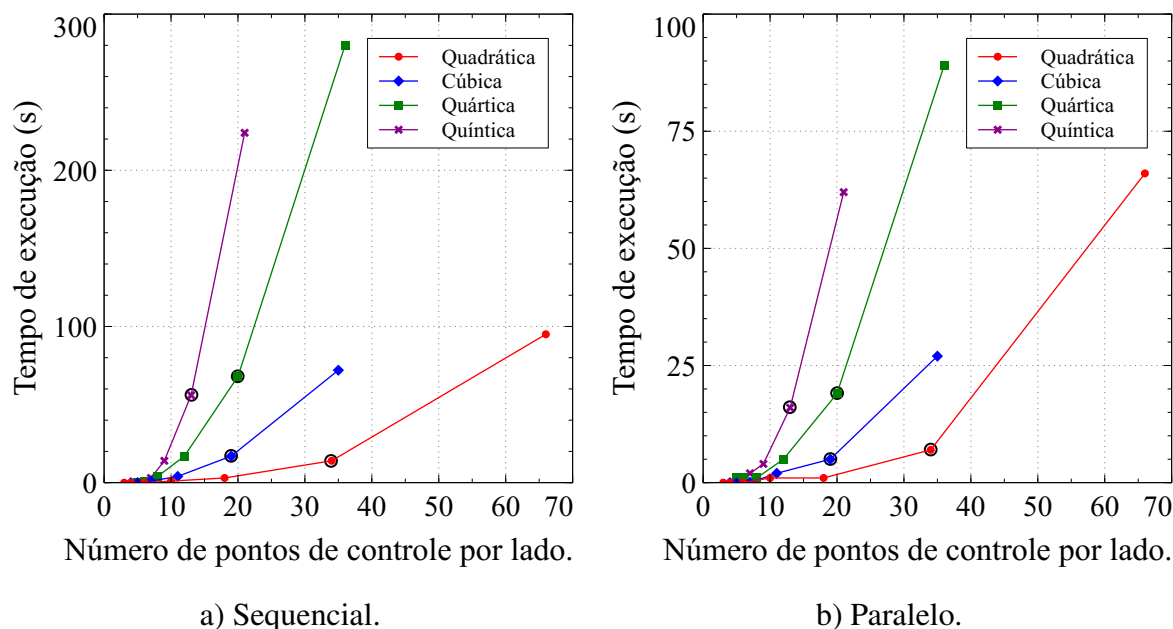
Malha	Quadrático		Cúbico		Quártico		Quíntico	
	w_p	δ_{ant}	w_p	δ_{ant}	w_p	δ_{ant}	w_p	δ_{ant}
(1x1x1)	0.1796		0.5052		1.4568		2.9798	
(2x2x1)	0.4890	172.25%	1.2803	153.42%	2.2724	55.99%	3.4115	14.49%
(4x4x1)	1.5030	207.35%	2.7807	117.19%	3.5907	58.01%	3.7390	9.60%
(8x8x1)	3.0569	103.39%	3.7278	34.06%	3.7939	5.66%	3.8183	2.12%
(16x16x1)	3.7268	21.92%	3.8222	2.53%	3.8435	1.31%	3.8546	0.95%
(32x32x1)	3.8320	2.82%	3.8560	0.88%	3.8605	0.44%		
(64x64x1)	3.8580	0.68%						

Fonte: Elaborada pelo autor.

Os tempos de execução de cada análise, considerando o programa sequencial e

paralelo, estão ilustrados na Figura 76. As malhas que obtiveram boa aproximação da resposta, considerando o critério mencionado anteriormente, estão marcada com um círculo.

Figura 76 – Tempos de execução das análises.



Fonte: Elaborada pelo autor.

A Tabela 12 mostra os tempos de execução das malhas consideradas para cada grau, bem como as medidas de *speedup* obtidas. O *speedup* é definido como a razão entre o tempo de execução sequencial e o tempo de execução em paralelo. Como foi utilizado um processador com quatro núcleos, é esperado um *speedup* máximo de 4.

Tabela 12 – Tempo de execução (s) e *speedup* obtidos.

	Sequencial	Paralelo	<i>Speedup</i>
Quadrática	14	7	2.00
Cúbica	17	5	3.40
Quártica	68	19	3.58
Quíntica	56	16	3.50

Fonte: Elaborada pelo autor.

Os resultados presentes na Tabela 12 mostram que foram obtidos *speedup* maiores para os casos de grau mais elevados. Como a estratégia de paralelização utilizada atua apenas na montagem da matriz de rigidez global, não existe efeito sobre a etapa de solução da equação de equilíbrio.

Uma análise do percentual do tempo de execução de cada função do programa foi realizada utilizando o programa *gprof*⁶⁴. A Figura 77 mostra um exemplo de utilização do

gprof em uma das análises realizadas. Entre as funções apresentadas, destacam-se as funções: *MatTripBtCB*, responsável pela realização do triplo produto $\mathbf{B}^T \mathbf{C} \mathbf{B}$; *CroutSolver*, responsável por solucionar o sistema $\mathbf{K} \mathbf{u} = \mathbf{f}$; *ShpFunc*, *DrvShpRST* e *DrvShpXYZ*, que consistem em avaliar as funções de formas e suas derivadas no sistema paramétrico e físico do modelo. Estas funções, na maioria dos casos estudados, representaram a maior parcela do tempo de execução do programa.

Figura 77 – Exemplo de saída do programa *gprof*.

```

1 Flat profile:
2
3 Each sample counts as 0.01 seconds.
4 % cumulative self self total
5 time seconds seconds calls Ts/call Ts/call name
6 60.21 3.98 3.98 MatTripBtCB(int, int, double**, double**, double, double**)
7 16.94 5.10 1.12 CroutSolver(int, double**, double*, int, int*)
8 4.99 5.43 0.33 cShapeSolidIGA::DrvShpRST(sNatCoord, sNatCoord*)
9 2.27 5.58 0.15 MatVecMult(int, int, double**, double*, double*)
10 2.12 5.72 0.14 cShapeSolidIGA::ShpFunc(sNatCoord, double*)
11 1.21 5.80 0.08 cPatch::GetCPInd(int, int, int)
12 1.06 5.87 0.07 memset
13 0.76 5.92 0.05 cSolid::BMatrix(int, double*, double*, sNodeCoord*, sNodeCoord*, cMatrix&)
14 0.76 5.97 0.05 cShapeSolid::DrvShpXYZ(sNatCoord, sNodeCoord*, double*, sNodeCoord*)
15 0.61 6.01 0.04 cSolid::BMatrix(int, double*, double*, cVector&, sNodeCoord*, sNodeCoord*, cMatrix&)
16 0.61 6.05 0.04 cBspPat3D::NumDim()
17 0.61 6.09 0.04 __int_malloc
18 0.45 6.12 0.03 MatAdd(int, int, double**, double**)
19 0.45 6.15 0.03 cPatch::GetMat(int, int)
20 0.45 6.18 0.03 cShapeIGA::EvalBspBas(int, double, cVector&)
21 0.45 6.21 0.03 __printf_fp
22 0.45 6.24 0.03 __mpn_dtvrem
23 0.45 6.27 0.03 __int_free
24 0.30 6.29 0.02 _L_unlock_13119
25 0.30 6.31 0.02 MatVecMultAcc(double, int, int, double**, double*, double*)
26 0.30 6.33 0.02 cElemParamTL::IntPntStress(cMatrix&, cMatrix&)
27 0.30 6.35 0.02 cKnotVector::GetMult(int)
28 0.30 6.37 0.02 cKnotVector::operator[](int)
29 0.30 6.39 0.02 cSymSkylMatrix::Add(int, int, double)
30 0.30 6.41 0.02 cBernsteinBasis::compBernsteinPols(int, double, cVector&)
31 0.30 6.43 0.02 std::basic_flebuf<char, std::char_traits<char>, >::xsputn(char const*, long)
32 0.30 6.45 0.02 __strncasemp_l_avx
33 0.15 6.46 0.01 VecAdd(int, double*, double*, double*)
34 0.15 6.47 0.01 cModLinear::CMatrix(cMatrix&)
35 0.15 6.48 0.01 cNurbPat3D::GetWeight(int)
36 0.15 6.49 0.01 cElemParamTL::StiffMat(cMatrix&)
37 0.15 6.50 0.01 cSymSkylMatrix::Get(int, int)
38 0.15 6.51 0.01 cPatch::GetDegree(int)
39 0.15 6.52 0.01 cShape::ShpMatInv(int, double (*) [3], double (*) [3], double*)
40 0.15 6.53 0.01 cMatrix::AllocMem()
41 0.15 6.54 0.01 cControl::PrintNodalStress()
42 0.15 6.55 0.01 cElement::AddGlobMat(cMatrix&, cSysMatrix*)
43 0.15 6.56 0.01 cElemLoad::ExtForce(double, cVector&)
44 0.15 6.57 0.01 free
45 0.15 6.58 0.01 malloc
46 0.15 6.59 0.01 malloc_consolidate
47 0.15 6.60 0.01 strlen
48 0.15 6.61 0.01 uselocale
--

```

Fonte: Elaborada pelo autor.

A Tabela 13 mostra os percentuais dos tempos de execução para as malhas que foram selecionadas para cada grau. O percentual referente ao triplo produto aumenta a medida que aumenta o grau do polinômio, chegando a quase 95% do tempo total para o caso quártico. Por outro lado, o percentual do tempo de solução do sistema de equações de equilíbrio tende a cair, tendo maior valor no grau quadrático. A utilização de polinômios maiores possibilita que menos graus de liberdade sejam utilizados para análise do problema. Entretanto, uma quadratura maior terá que ser utilizada, necessitando de um número maior de pontos de integração e consequentemente de maior custo computacional para avaliação da matriz de rigidez dos elementos.

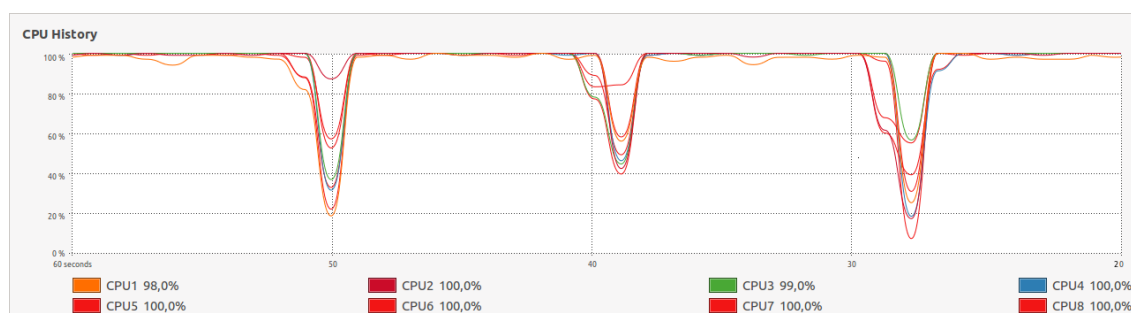
Tabela 13 – Percentual do tempo de execução total (%).

	Trip($\mathbf{B}^T \mathbf{CB}$)	Solver($\mathbf{Ku} = \mathbf{f}$)	FF + Der.	Outras
Quadrática	59.45	23.18	3.56	13.81
Cúbica	83.55	3.13	3.67	9.65
Quártica	90.08	1.65	2.50	5.77
Quíntica	94.36	0.36	1.61	3.67

Fonte: Elaborada pelo autor.

Por fim, a Figura 78 mostra o uso percentual de cada processador durante a execução de uma análise não-linear do programa FAST paralelizado. Pode-se identificar os intervalos em que o programa está executando a montagem da matriz de rigidez do problema, onde todos os núcleos de processamento estão trabalhando totalmente. Quando a montagem termina, o percentual de uso dos processadores diminui, neste momento ocorre a continuação de códigos não paralelizados, como a aplicação de técnicas iterativas de solução da equação de equilíbrio não-linear do problema. Quando o programa necessita avaliar mais uma vez a matriz de rigidez dos elementos, inicia-se de novo o ciclo com aproveitamento total dos núcleos de processamento.

Figura 78 – Uso percentual de cada processador durante a execução do FAST em paralelo.



Fonte: Elaborada pelo autor.

Uma importante observação é que o tempo adicional (*overhead*) devido à utilização de orientação ao objeto é desprezível. Além disso as funções de forma das NURBS não apresentaram custos computacionais significativos, quando comparadas às operações de triplo produto e solução da equação de equilíbrio. Isto mostra que os custo computacional em programas de elementos finitos ou AIG é definido principalmente por operações matemáticas como multiplicação de matrizes e solução de sistemas lineares.

6 OTIMIZAÇÃO DE LAMINADOS

A otimização tem papel fundamental no projeto de estruturas laminadas, uma vez que os materiais compósitos possuem um arranjo que possibilita uma maior adaptação da estrutura em relação aos esforços solicitantes. As características de cada camada (espessura, material e orientação) do esquema de laminação destas estruturas podem ser escolhidos de forma a melhorar significativamente seu desempenho estrutural.

Existem dois tipos básicos de problemas de otimização de laminados⁶⁵. O primeiro corresponde à maximização da resistência da estrutura, mantendo constantes a geometria e os materiais adotados e considerando restrições de desempenho e fabricação. O segundo corresponde à minimização do custo, considerando restrições de resistência, utilização e fabricação.

O problema de maximização da resistência considera normalmente que um mesmo material compósito é utilizado para toda a estrutura laminada. Este tipo de problema pode ser formulado como:

$$\left\{ \begin{array}{l} \text{Determinar } \mathbf{x} = [(e_1, \theta_1), (e_2, \theta_2), \dots, (e_{N_l}, \theta_{N_l})] \\ \text{que maximiza } f(\mathbf{x}) \\ \text{sujeito a } c_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, N_c \end{array} \right. \quad (136)$$

onde \mathbf{x} é o esquema de laminação (*layup*) contendo N_l lâminas, e é a espessura da lâmina, θ é a orientação das fibras, f é a função objetivo e N_c é o número de restrições c_i consideradas. Como exemplos de função objetivo tem-se o fator de segurança, a rigidez, a carga de flambagem e a frequência fundamental da estrutura. As restrições incluem requisitos de desempenho estrutural, boas práticas de projeto e limitações do processo de fabricação.

Um exemplo de boa prática de projeto é a limitação da espessura ou número de camadas adjacentes de mesmo ângulo^{5,8,66} utilizada para reduzir a fissuração da matriz. Outros exemplos são a utilização apenas de laminados simétricos ou laminados balanceados-simétricos, de maneira a evitar acoplamentos indesejados e limitar as deformações devido à absorção de umidade e variação de temperatura⁶⁷. A utilização destas laminações facilita a solução do problema de otimização, pois reduz o número de variáveis de projeto.

Uma versão mais simples do problema de maximização ocorre quando todas as lâminas tem a mesma espessura, como ocorre quando se usa lâminas de material compósito pré-impregnado conhecidos como *prepreg*. Neste caso, as únicas variáveis de projeto são as orientações das lâminas.

Os problemas de minimização de custo podem ser formulados como:

$$\left\{ \begin{array}{l} \text{Determinar } \mathbf{x} = [(e_1, \theta_1, m_1), (e_2, \theta_2, m_2), \dots, (e_{N_l}, \theta_{N_l}, m_{N_l})] \\ \text{que minimiza } f(\mathbf{x}) \\ \text{sujeito a } c_i(\mathbf{x}) \leq 0, i = 1, 2, \dots, N_c \end{array} \right. \quad (137)$$

onde m é o material de cada camada e f representa o custo da estrutura. Neste caso, tanto restrições de resistência, como de rigidez, frequência natural e flambagem são consideradas. Também podem ser consideradas restrições de utilização, de fabricação e de boas práticas de projeto. Neste caso, o número total de lâminas N_l pode ser variável. Quando apenas um material é considerado, a minimização do custo é similar à minimização da espessura, do peso ou do volume de material. Como a espessura total do laminado (ou número de lâminas) é variável, este tipo de problema é mais complexo que o de maximização da resistência.

Devido às limitações do processo de fabricação, muitas vezes a espessura e os ângulos de orientação das fibras são limitados a valores discretos^{1,3,66}. Assim, em muitos projetos as orientações são limitadas a 0° , $\pm 45^\circ$ e 90° ⁶⁸.

É importante notar que a otimização de laminados com variáveis discretas é um problema de otimização combinatória. Estes problemas de otimização são de difícil solução, pois o espaço de busca cresce de maneira exponencial com o número de variáveis de projeto:

$$N_{sol} = (N_e \times N_\theta \times N_m)^{N_l} \quad (138)$$

onde N_e , N_θ e N_m representam o número de espessuras, orientações e materiais, respectivamente. Por exemplo, na otimização de um laminado considerando 1 espessura, 12 orientações ([90, 75, 60, 45, 30, 15, 0, -15, -30, -45, -60, -75]) e 3 materiais diferentes, tem-se 36 combinações possíveis por lâmina, resultando em 60466176 projetos para 5 lâminas, 3.6562×10^{15} projetos para 10 lâminas e 1.3367×10^{31} para 20 lâminas.

Os problemas de otimização de laminados com variáveis discretas pertencem a classe dos problemas NP-difícil⁶⁹, para os quais não são conhecidos algoritmos capazes de resolvê-los em tempo polinomial. Adicionalmente, estes problemas também são multi-modais, pois podem apresentar vários mínimos locais.

A otimização de laminados pode ser realizada com o uso de métodos baseados no uso de gradientes (métodos de primeira ordem) ou de métodos de ordem zero, que utilizam apenas os valores da função objetivo e restrições. Neste sentido, é importante ressaltar que em muitos casos as variáveis de projeto de laminados são tratadas como contínuas^{9,68,70-72}. Neste caso, os métodos baseados em gradientes são muito rápidos e precisos. Estes métodos também pode ser utilizados no caso de variáveis discretas, sendo as soluções contínuas arredondadas para valores discretos mais próximos. Contudo, esta abordagem pode levar a soluções não-ótimas ou mesmo inviáveis. Adicionalmente, os métodos baseados em gradientes tendem a convergir para o mínimo local mais próximo do projeto inicial.

Assim, atualmente a abordagem mais utilizada para a solução de problemas de otimização de laminados é baseada na utilização de métodos de ordem zero, com destaque para os algoritmos bio-inspirados. Os primeiros trabalhos neste sentido foram baseados no uso de Algoritmos Genéticos^{1,3,66,68}. Posteriormente, foram utilizadas a Otimização por Nuvem de Partículas (*Particle Swarm Optimization* - PSO)^{9,12,69,73} e a Otimização por Colônia de Formi-

gas (*Ant Colony Optimization - ACO*)^{6-8,74}.

Apesar da otimização de compósitos laminados ter sido realizada tradicionalmente limitando as orientações a $\theta \in [0^\circ, \pm 45^\circ, 90^\circ]$, existem trabalhos considerando laminados com um conjunto maior de orientações possíveis, como por exemplo $\theta \in [0^\circ, \pm 15^\circ, \dots, \pm 75^\circ, 90^\circ]$ ou $\theta \in [0^\circ, \pm 5^\circ, \dots, \pm 85^\circ, 90^\circ]$. Estes laminados são conhecidos na literatura como Laminados Dispersos e tem atraindo atenção recentemente, pois possibilitam o projeto de estruturas com maior capacidade de carga⁷⁴.

O presente trabalho faz parte da linha de pesquisa de otimização de estruturas laminadas, desenvolvida no Laboratório de Mecânica Computacional e Visualização (LMCV) da Universidade Federal do Ceará (UFC). Trabalhos anteriores desta linha de pesquisa trataram da otimização de *risers* de material compósito utilizando um AG⁴ e do desenvolvimento de um AG paralelo para otimização de laminados em clusters com arquiteturas híbrida de memória compartilhada e memória distribuída^{5,18}.

No presente trabalho foi desenvolvido um método de otimização de laminados baseado na combinação da heurística do PSO com alguns operadores do AG. Esta combinação visa obter um algoritmo que reúna as qualidades do PSO e do AG. Por ser um método originado para otimização contínua, o PSO tende a apresentar um desempenho melhor na otimização de laminados dispersos, pois neste caso as variáveis de projeto aproximação de variáveis contínuas. Por outro lado, os operadores do AG ajudam a evitar a convergência prematura para mínimos locais, que é um dos maiores problemas do PSO.

O método desenvolvido foi comparado com um Algoritmo Genético apresentado na literatura¹⁸. Um estudo foi realizado para avaliar as melhores variantes do algoritmo proposto. Em tal estudo, cada variante do método teve seus parâmetros de entrada calibrados, através de um procedimento de meta-otimização. Tal procedimento é utilizado para obter os melhores parâmetros dos algoritmos estudados.

Toda a implementação foi realizada no sistema BIOS (*Bio-Inspired Optimization System*), que é um programa acadêmico implementado em C++ utilizando o paradigma de Programação Orientação a Objetos (POO). O desenvolvimento do BIOS começou na dissertação de Rocha (2013), onde a estrutura geral do programa foi concebida e o AG com paralelização híbrida^{5,18} foi implementado.

Os detalhes dos algoritmos utilizados, bem como os resultados obtidos na otimização de laminados, são apresentados nos itens a seguir.

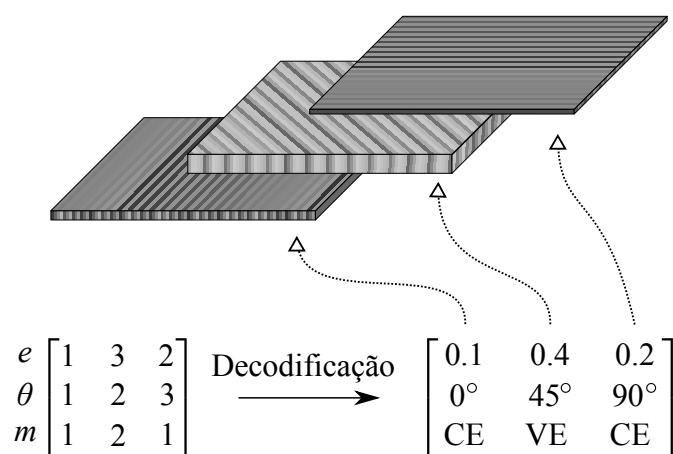
6.1 Codificação das variáveis

Neste trabalho, as variáveis de projeto do problema de otimização de compósitos laminado são tratadas como inteiras. O laminado é representado como uma matriz de inteiros com três linhas (espessura, orientação e material) e um dado número de colunas, cada coluna representando uma lâmina. As opções de espessura, orientação e material utilizados no pro-

blema de otimização são codificadas em listas de variáveis, como mostrado na Figura 79. Um processo de decodificação é aplicado para transformar a representação das variáveis inteiras na representação real dos parâmetros de projeto do laminado. Este processo de decodificação permite a utilização de laminações simétricas, balanceadas, simétricas-balanceadas e genéricas¹⁸.

Figura 79 – Processo de Codificação e Decodificação.

Código	Espessura	Código	Orientação	Código	Material
1	0.1 mm	1	0°	1	Carbono-Epoxi (CE)
2	0.2 mm	2	45°	2	Vidro-Epoxi (VE)
3	0.4 mm	3	90°		



Fonte: Elaborada pelo autor.

6.2 Tratamento de restrições

Os algoritmos bio-inspirados utilizados neste trabalho não conseguem resolver diretamente problemas restritos como os algoritmos de Programação Matemática, pois não consideram o cálculo de derivadas⁷⁵. Desta forma, técnicas de tratamento de restrições devem ser utilizadas para garantir a convergência ocorra para uma solução viável.

Os métodos de tratamento de restrições podem ser classificados em interiores e exteriores. Os métodos diretos ou interiores consideram apenas pontos viáveis durante a otimização. O exemplo mais simples de aplicação deste método é o da Pena de Morte, no qual sempre que o algoritmo de otimização obtém pontos inviáveis, estes pontos são descartados, sendo gerados novos valores até que se obtenha a um ponto viável. Este tipo de procedimento pode ser computacionalmente ineficiente em problemas com muitas restrições, apresentando alto custo computacional para avaliações das funções objetivo e das restrições. Outra desvantagem é a perda de boas características que as soluções inviáveis possam conter.

Outro método interior é o uso de Técnicas de Reparo, em que os pontos inviáveis são modificados a fim de retornarem para a região viável. A utilização deste método necessita de um grande conhecimento do problema, além de não ser possível em todos os casos. Existem

aplicações de tais técnicas em problemas de otimização de laminados⁷⁶.

Os métodos indiretos ou exteriores permitem que pontos inviáveis participem do processo de otimização, porém a definição da função objetivo é alterada para pontos inviáveis, diminuindo a influência destes pontos no processo de otimização. Neste trabalho são apresentados duas estratégias de penalidade.

O método de penalidade exterior estático acrescenta ao cálculo da função objetivo uma parcela que depende das violações de cada restrição e de uma constante conhecida como fator de penalidade (k). Considerando um problema de minimização, a função objetivo penalizada (f_p) é dada por:

$$f_p(\mathbf{x}) = f(\mathbf{x}) + \sum_{l=1}^m k \max(c_l(\mathbf{x}), 0) \quad (139)$$

onde f é a função objetivo, c_l é o valor da restrição e m é o número de restrições. O valor ideal do fator de penalidade é dependente de cada problema, sendo normalmente utilizado um valor bastante elevado que é ajustado em função dos resultados obtidos.

Outro método de penalidade é o proposto por Deb⁷⁷. Neste método, a função objetivo penalizada é igualada a do pior indivíduo viável f_{max} , sendo também acrescentado o valor da maior violação obtida. Assim, a função objetivo penalizada é dada por:

$$f_p(\mathbf{x}) = \begin{cases} f(\mathbf{x}), & \text{se } \mathbf{x} \text{ for viável} \\ f_{max} + \sum_{l=1}^m \max(c_l(\mathbf{x}), 0), & \text{se } \mathbf{x} \text{ for inviável} \end{cases} \quad (140)$$

É importante observar que o método de Deb não requer nenhum parâmetro definido pelo usuário, ao contrário do método de penalidade estática.

É importante ressaltar que uma correta utilização dos métodos de penalidade necessita que as restrições do problema estejam escritas de forma adimensional, pois valores de restrições com diferentes ordens de grandeza podem comprometer o efeito da penalização.

6.3 Busca Aleatória

O método da busca aleatória, ou busca randômica pura, é o método de otimização estocástico mais simples. A grande maioria dos outros métodos estocásticos pode ser classificados como variações da busca aleatória⁷⁵. O método consiste em gerar um conjunto de pontos a partir de uma distribuição randômica e avaliá-los. O menor valor obtido nas avaliações é considerado o mínimo do problema. O método pode ser extremamente ineficiente, quase sempre necessitando de um número muito grande de pontos para obter uma boa solução do problema.

Uma das motivações para utilização da busca aleatória neste trabalho é a consideração da teoria conhecida como *No Free Lunch*, apresentada inicialmente por Wolpert e Macready⁷⁸ e discutida em outros trabalhos⁷⁹. A teoria afirma resumidamente que qualquer algoritmo de

otimização estocástico, quando utilizado em uma grande quantidade de problemas diferentes, tendem obter resultados semelhantes aos da busca aleatória.

Apesar disto, a utilização de uma heurística de busca pode levar a melhores desempenhos em determinados problemas. Portanto, é importante avaliar se os algoritmos utilizados neste trabalho possuem desempenho superior à Busca Aleatória, considerando os problemas de otimização de laminados.

6.4 Algoritmos Genéticos

Os Algoritmos Genéticos são métodos de otimização cuja heurística se baseia na Teoria da Evolução de Darwin. Segundo esta teoria, indivíduos com maior adaptação ao meio ambiente possuem maiores chances de transmitir suas características genéticas para as próximas gerações, enquanto que indivíduos com menor aptidão tendem a serem extintos ao longo das gerações.

Em problemas de otimização, cada indivíduo representa um projeto diferente e a qualidade de cada projeto é medida por uma função aptidão (*fitness function*) obtida a partir da função objetivo penalizada. Assim, a avaliação da função aptidão envolve a realização de uma análise estrutural para determinação das respostas mecânicas do laminado (deslocamentos, tensões, etc.) e o cálculo das restrições. A população corresponde ao conjunto de indivíduos que evoluem ao longo de um conjunto de gerações.

Existem diversas variações de Algoritmos Genéticos na literatura, porém grande parte destes funcionam de maneira similar. A discussão apresentada neste capítulo é baseada no AG implementado no sistema BIOS para otimização de laminados¹⁸. Inicialmente é gerada uma população de indivíduos de forma aleatória, cada um representando um ponto no espaço de busca. Alguns indivíduos são selecionados para participar do cruzamento, gerando uma nova população a partir da combinação dos indivíduos existentes.

Neste processo, os indivíduos com maior aptidão tem maior probabilidade de serem selecionados para cruzamento. Os filhos gerados pelo processo de cruzamento podem sofrer mutações aleatórias, de forma a evitar a convergência prematura para soluções não ótimas. Os melhores indivíduos de uma geração podem ser copiados diretamente para a geração seguinte, num processo conhecido como elitismo. O elitismo garante que as informações dos melhores indivíduos não sejam perdidas e que a melhor solução nunca piore de uma geração para outra.

O processo se repete até que uma condição de parada seja satisfeita. Diversos critérios de parada podem ser utilizados, sendo os mais comuns o número máximo de gerações e o número de gerações sem melhoria na função objetivo penalizada. Neste trabalho a condição de parada utilizada é o número máximo de gerações.

6.4.1 Função Aptidão e Seleção

Como discutido anteriormente, a probabilidade de seleção para cruzamento aumenta com o valor da função aptidão do indivíduo. É importante notar que os AGs foram desenvolvidos inicialmente para problemas de maximização de funções sem restrições. Nestes problemas, a função aptidão pode ser tomada como a própria função objetivo, desde que esta tenha sempre um valor numérico positivo, pois não faz sentido uma probabilidade nula.

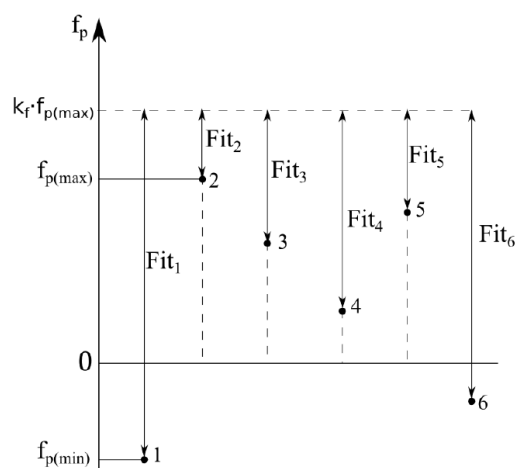
O cálculo da função aptidão (Fit) é um pouco mais complexo no caso de problemas de projeto ótimo, que normalmente são formulados na forma de minimização de uma função objetivo considerando um conjunto de restrições. Neste caso, a aptidão depende da função objetivo penalizada:

$$Fit_i = k_f \max(|f_{p(min)}|, |f_{p(max)}|) - f_{pi} \quad (141)$$

onde $f_{p(min)}$ e $f_{p(max)}$ são respectivamente os valores mínimo e máximo das funções objetivo penalizadas da população e k_f é um coeficiente utilizado para evitar que o pior indivíduo possua probabilidade nula de seleção.

A Figura 80 mostra uma representação do processo de mapeamento linear utilizado. Verifica-se que este mapeamento garante que a função aptidão seja sempre positiva.

Figura 80 – Mapeamento da função objetivo.



Fonte: Rocha (2013).

O processo de seleção da população para cruzamento é realizado em duas etapas. Inicialmente é gerada uma população intermediária (*mating pool*) formada por indivíduos aptos a participar do cruzamento. Em seguida são selecionados os pares de pais que irão cruzar, gerando novos indivíduos (filhos). A seleção dos indivíduos pode ser feita por roleta, ranking ou torneio⁷⁵.

Neste trabalho as duas etapas de seleção são realizadas por ranking, onde os indivíduos são ordenados de acordo com o valor de sua aptidão, de modo que os indivíduos mais

aptos possuem maior valor de ranking. A probabilidade de seleção de cada indivíduo p_i é calculado como:

$$p_i = \frac{Rnk_i}{\sum_{k=1}^{N_{ind}} Rnk_k} \quad (142)$$

onde N_{ind} é o número de indivíduos e Rnk_i é o ranking do indivíduo. A Figura 81 mostra um exemplo da utilização de seleção por ranking.

Figura 81 – Probabilidades de seleção por ranking.

Individual	Fitness Function	Ranking	p_i
1	30	3	0.30
2	25	2	0.20
3	2	1	0.10
4	40	4	0.40

$$p_i = \frac{Rnk_i}{\sum_{k=1}^{N_{ind}} Rnk_k}$$

Fonte: Elaborada pelo autor.

6.4.2 Cruzamento e Elitismo

O cruzamento é o principal operador dos algoritmos genéticos, sendo o principal responsável pela convergência para a solução. Ele é aplicado a dois indivíduos (pais), gerando dois novos indivíduos (filhos) com características de ambos os pais.

Existem diversas formas de realizar o cruzamento. O método utilizado neste trabalho calcula as variáveis de projeto dos filhos por meio de uma combinação linear de ambos os pais. Para cada lâmina é gerado um número randômico de 0 a 1, sendo aplicado uma média ponderada entre as camadas dos dois pais para obtenção dos dois filhos, como mostrado na Figura 82. Como as variáveis são inteiras, os valores obtidos pela combinação linear são arredondados para o inteiro mais próximo.

Figura 82 – Processo de cruzamento sendo aplicado a uma lâmina.

$$\begin{array}{c}
 \text{Pai 1} \left| \begin{array}{cc} T1 & T2 \\ A1 & A2 \\ M1 & M2 \end{array} \right| \quad \text{Filho 1} \left| \begin{array}{cc} r1 \times T1 + (1-r1) \times T4 & r2 \times T2 + (1-r2) \times T5 \\ r1 \times A1 + (1-r1) \times A4 & r2 \times A2 + (1-r2) \times A5 \\ r1 \times M1 + (1-r1) \times M4 & r2 \times M2 + (1-r2) \times M5 \end{array} \right| \\
 \\
 \text{Vetor} \\
 \text{Randômico} \left| \begin{array}{cc} r1 & r2 \end{array} \right| \rightarrow \\
 \\
 \text{Pai 2} \left| \begin{array}{cc} T4 & T5 \\ A4 & A5 \\ M4 & M5 \end{array} \right| \quad \text{Filho 2} \left| \begin{array}{cc} r1 \times T4 + (1-r1) \times T1 & r2 \times T5 + (1-r2) \times T2 \\ r1 \times A4 + (1-r1) \times A1 & r2 \times A5 + (1-r2) \times A2 \\ r1 \times M4 + (1-r1) \times M1 & r2 \times M5 + (1-r2) \times M2 \end{array} \right|
 \end{array}$$

Fonte: Rocha (2013).

Neste trabalho, adota-se uma estratégia elitista para a geração da nova população.

Nesta estratégia, os melhores indivíduos da geração atual são copiados sem modificação para a geração seguinte, sendo o restante da nova população gerado por cruzamento. O número de indivíduos gerados por cruzamento (N_{cross}) é definido como:

$$N_{cross} = \text{round}(r_{cross} \cdot N_{ind}) \quad (143)$$

onde N_{ind} é o número de indivíduos (i.e. tamanho da população) e r_{cross} é taxa de cruzamento. O número de indivíduos copiados no elitismo é dado pela diferença:

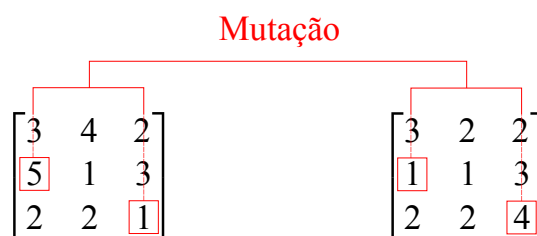
$$N_{elit} = N_{ind} - N_{cross} \quad (144)$$

Como discutido anteriormente, o elitismo garante que as melhores soluções encontradas até o momento não sejam desprezadas, garantindo que a melhor solução nunca piore de uma geração para outra.

6.4.3 Mutação

A mutação é um operador utilizado para manter a diversidade da população durante as gerações, evitando a convergência prematura para um mínimo local e aumentando a chance do algoritmo achar o mínimo global do problema. Neste trabalho, a mutação pode atuar de forma aleatória sobre cada variável do indivíduo, com uma dada probabilidade (p_{mut}).

Figura 83 – Processo de mutação sendo aplicado.



Fonte: Elaborada pelo autor.

A mutação é implementada em duas etapas. Inicialmente, gera-se um número aleatório no intervalo $[0, 1]$ para cada variável, se este número for menor que p_{mut} , então a variável sofrerá mutação. Em seguida, a mutação é aplicada gerando-se um número inteiro aleatório entre os limites mínimo e máximo da variável. A aplicação deste operador é ilustrada pela Figura 83.

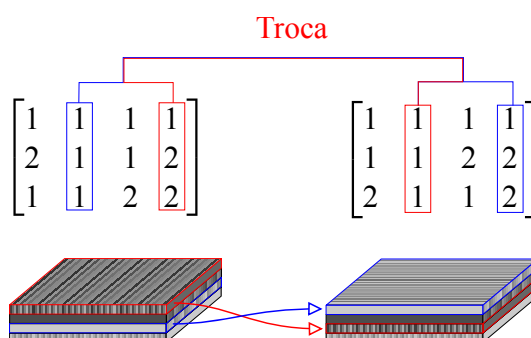
6.4.4 Operadores Adicionais para Laminados

Muitos trabalhos estudaram a aplicação de mecanismos de busca local em métodos de otimização heurísticos para melhorar sua aplicabilidade no contexto da otimização de lami-

nados. Uma contribuição importante foi a criação do operador *Troca de Lâmina*, introduzido em Le Riche e Haftka (1993) e utilizado em outros trabalhos^{3,5,80}.

Este operador promove a troca de duas camadas do laminado, como ilustra a Figura 84. No presente trabalho, todas as lâminas tem a mesma probabilidade (p_{swap}) de sofrer uma troca. A implementação é realizada em duas etapas: na primeira gera-se um número aleatório entre $[0, 1]$ e a troca ocorrerá se este número for menor que p_{swap} , na segunda etapa escolhe-se a lâmina a ser trocada mediante a geração de um número inteiro aleatório. Este operador é aplicado após a etapa de mutação.

Figura 84 – Exemplo de aplicação do operador troca de lâmina.

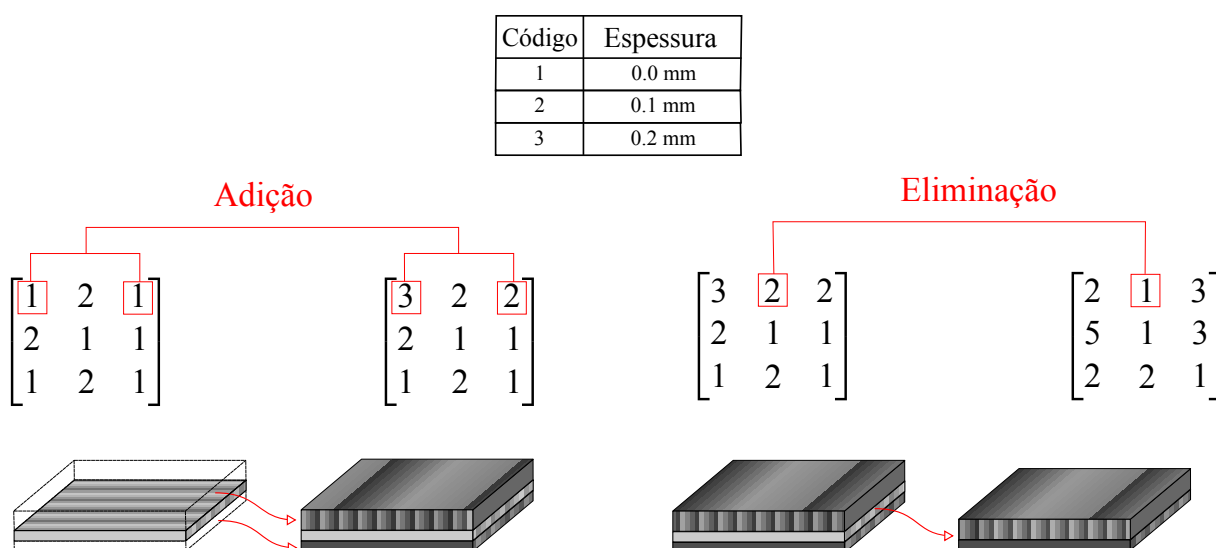


Fonte: Elaborada pelo autor.

De acordo com Teoria Clássica da Laminação, a troca de lâminas altera a rigidez à flexão do laminado (matriz **D**), mas não afeta a rigidez de membrana (matriz **A**). Assim, este operador é importante em problemas que consideram flambagem, frequências naturais e deslocamentos transversais, seja como função objetivo ou restrição.

Outros operadores importantes para problemas de minimização de custo são os operadores *Eliminação de Lâmina* e *Adição de Lâmina*. Neste trabalho, o primeiro código da espessura utilizada em problemas de minimização representa o valor nulo. Este valor é decodificado como ausência da lâmina, reduzindo o número de camadas do laminado, como apresentado na Figura 85.

Figura 85 – Exemplo de aplicação dos operadores de adição e eliminação de lâmina.



Fonte: Elaborada pelo autor.

O operador de Eliminação de Lâmina tem uma probabilidade p_{del} de atribuir o valor nulo para a espessura da camada. O operador de Adição de Lâmina tem uma probabilidade p_{add} de ativar uma camada de espessura nula. Estes operadores são aplicados após o operador Troca de Lâmina.

É importante notar que a codificação utilizada possibilita que camadas sejam adicionadas e deletadas naturalmente sem estes operadores. Por exemplo, o operador de cruzamento pode ativar ou deletar camadas.

6.4.5 Pseudo-código

O Algoritmo Genético utilizado neste trabalho incorpora os elementos discutidos nos itens anteriores. O pseudo-código deste algoritmo, mostrando a sequência de aplicação dos operadores, é apresentado na Figura 86.

Figura 86 – Pseudo-código do Algoritmo Genético.

```

Criar as populações Pop, MatingPool, Pais, Filhos;
Inicialização(Pop);
para Cada geração até MaxGen faça
    Avaliação(Pop);
    Penalidade(Pop);
    Aptidão(Pop);
    MatingPool ← Seleção(Pop);
    Pais ← Seleção(MatingPool);
    Filhos ← Cruzamento(Pais);
    Mutação(Filhos);
    TrocaLâmina(Filhos);
    AdiçãoLâmina(Filhos);
    EliminaçãoLâmina(Filhos);
    Pop ← Filhos
fim

```

Fonte: Elaborada pelo autor.

Os parâmetros numéricos deste algoritmo são: o número máximo de gerações N_{gen} , um número indivíduos da população N_{ind} , a taxa de cruzamento r_{cross} e as probabilidades de mutação p_{mut} , troca de lâminas p_{swap} , adição de lâmina p_{add} e eliminação de lâmina p_{del} .

É importante notar que o custo computacional da aplicação dos operadores do AG é muito baixo. Desta forma, o custo computacional da otimização é dominado pelo tempo necessário para a avaliação dos indivíduos, uma vez que cada avaliação requer a análise de uma estrutura laminada. Portanto, o tempo de processamento do processo de otimização é proporcional ao número de avaliações N_{aval} realizado durante o processo de otimização:

$$N_{aval} = N_{ind} + N_{gen} \text{ round}(r_{cross} \cdot N_{ind}) \quad (145)$$

Como os AGs são algoritmos de ordem zero, eles requerem normalmente um grande número de avaliações para obter bons resultados no processo de otimização, resultando em um custo computacional elevado quando comparado com os algoritmos baseados em gradiente. Contudo, a avaliação de cada indivíduo é independente dos demais, podendo ser facilmente paralelizada. Estratégias de paralelização mais avançadas baseada no uso de sub-populações ou ilhas podem também ser utilizadas^{5,18}.

6.5 Otimização por Nuvem de Partículas

O método de Otimização por Nuvem de Partículas (em inglês *Particle Swarm Optimization*, ou PSO) foi proposto por Kennedy e Eberhart⁸¹, onde foi descrita sua formulação para otimização de problemas contínuos sem restrição. A heurística do método busca reproduzir o comportamento social de grupos de animais, como bandos de pássaros ou cardumes de peixes, na busca por alimento. A ideia básica é que cada integrante do bando procure no espaço por uma fonte de comida, sendo esta busca baseada na experiência própria do integrante, como também pela experiência de outros integrantes do bando.

Desta forma, cada partícula da nuvem representa um projeto, que percorre o espaço de projeto na busca por posições que forneçam um valor menor para a função objetivo. A cada passo de tempo, os valores das variáveis de projeto são atualizados com base velocidade da partícula e a melhor posição encontrada pela partícula é atualizada.

No início, um conjunto de partículas é gerado aleatoriamente dentro do espaço de busca. Cada partícula é representada um vetor \mathbf{x} , cuja dimensão é igual ao número de variáveis de projeto. A velocidade e a posição das partículas são atualizadas iterativamente por:

$$\begin{aligned} \mathbf{v}^{(i)} &= w\mathbf{v}^{(i)} + c_1 r_1 (\mathbf{x}_p^{(i)} - \mathbf{x}^{(i)}) + c_2 r_2 (\mathbf{x}_g^{(i)} - \mathbf{x}^{(i)}) \\ \mathbf{x}^{(i)} &= \mathbf{x}^{(i)} + \mathbf{v}^{(i)} \end{aligned} \quad (146)$$

onde $\mathbf{v}^{(i)}$ é a velocidade da partícula i , w é a constante de inércia das partículas, c_1 é o parâmetro cognitivo das partículas, c_2 é o parâmetro social das partículas, r_1 e r_2 são números randômicos uniformemente distribuídos no intervalo $[0, 1]$. Nesta equação, $\mathbf{x}_p^{(i)}$ representa a melhor posição obtida pela partícula até a presente iteração e $\mathbf{x}_g^{(i)}$ representa a melhor posição obtida pelas partículas vizinhas. A vizinhança da partícula depende da topologia da nuvem.

De acordo com a Equação (146), a velocidade representa na verdade o deslocamento que a partícula vai sofrer, como se o incremento de tempo fosse unitário ($\Delta t = 1$). Assim, a velocidade pode ser associada à direção de busca utilizada pelos algoritmos de Programação Matemática⁷⁵. No PSO, esta direção é resultado da combinação linear de três vetores: a velocidade anterior da partícula ($\mathbf{v}^{(i)}$), o vetor que aponta para a melhor posição da partícula ($\mathbf{x}_p^{(i)} - \mathbf{x}^{(i)}$) e o que aponta para a melhor posição da vizinhança ($\mathbf{x}_g^{(i)} - \mathbf{x}^{(i)}$). Assim, as partículas são atraídas para a melhor posição que ela já ocupou e a melhor posição que as partículas vizinhas já encontraram.

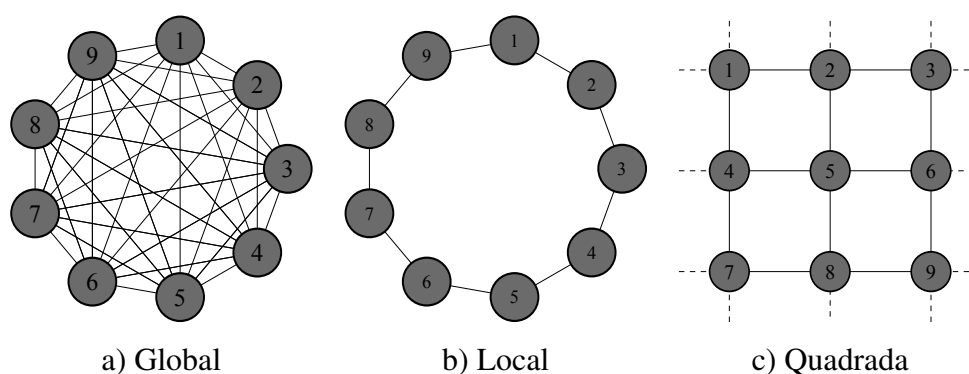
A topologia utilizada determina como a vizinhança da partícula influencia em seu aprendizado. Inicialmente⁸¹, a vizinhança era definida como a nuvem inteira, correspondendo à Topologia Global. Nesta topologia, a posição $\mathbf{x}_g^{(i)}$ corresponde à melhor posição (i.e. projeto) obtida até o momento por qualquer partícula da nuvem.

A interação entre as partículas e a vizinhança destas foi estudada em alguns trabalhos^{82,83}, mostrando que o uso da Topologia Local ou da Topologia Quadrada podem melhorar

o desempenho do PSO. Na Topologia Local (também conhecida na literatura como Topologia em Anel), cada partícula é influenciada apenas por outras duas partículas. Na topologia Quadrada o enxame é arrumado em forma matricial, e a vizinhança da partícula é definida em cada direção (acima, abaixo, direita e esquerda) na matriz de partículas.

A Figura 87 ilustra as topologias discutidas neste trabalho para o caso de uma nuvem com 9 partículas. É importante notar que estas vizinhanças são construídas levando em consideração o índice vetorial da partícula, não levando em conta a real proximidade entre as partículas no espaço de busca.

Figura 87 – Topologias da nuvem.



Fonte: Elaborada pelo autor.

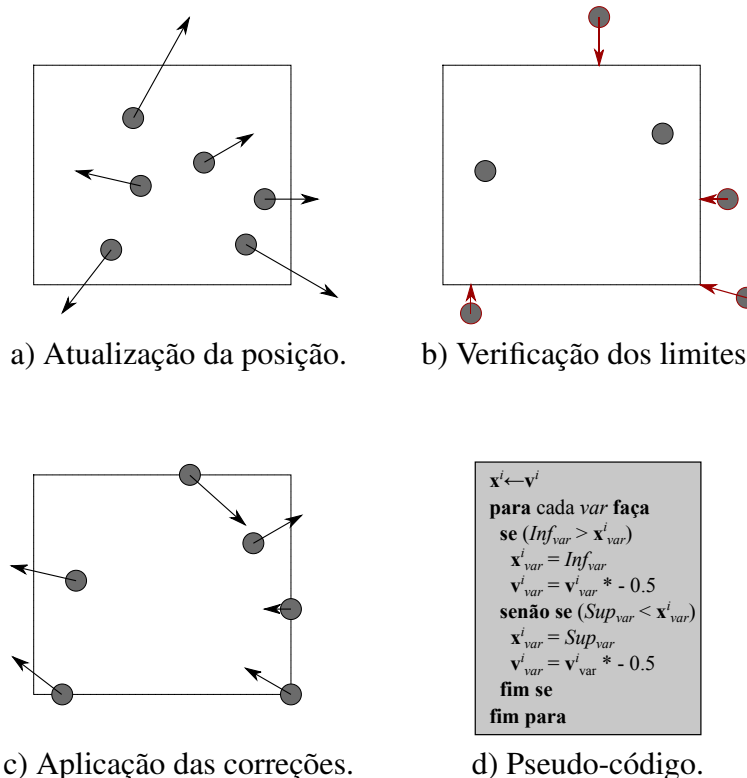
A Topologia Global dá ao método uma capacidade de convergência rápida, porém possui uma grande vulnerabilidade ao problema da convergência prematura. Muitos trabalhos propuseram alterações no algoritmo para solução do problema de convergência prematura, como a utilização de parâmetros adaptativos e operadores de mutação sobre a posição da partícula⁸⁴, mecanismo de repulsão de partícula⁸⁵ e a incorporação de efeitos de colisão entre partículas⁸⁶.

As topologias Local e Quadrada atribuem ao método uma capacidade maior de exploração do espaço de busca, preservando a diversidade da nuvem. Assim, normalmente a velocidade de convergência é reduzida em comparação com o uso da Topologia Global, porém uma menor suscetibilidade à convergência prematura é obtida. É importante ressaltar que o algoritmo denominado de PSO padrão, proposto por Bratton e Kennedy (2007), utiliza a Topologia Local.

Assim como no AG, o custo computacional das operações básicas do PSO, representado pela Equação (146), é muito baixo, sendo o esforço computacional dominado pela avaliação de cada partícula (i.e. projeto). Considerando um número máximo de iterações N_{iter} e um número de partículas N_{par} , o número de avaliações N_{aval} pode ser obtido por:

$$N_{aval} = N_{par} (N_{gen} + 1) \quad (147)$$

Figura 88 – Manipulação das restrições laterais.



Fonte: Elaborada pelo autor.

Assim, o custo computacional depende diretamente do número de partículas e do número de iterações utilizado. É interessante notar que estes parâmetros correspondem ao número de indivíduos e número de gerações, utilizados no AG. A utilização de paralelização também é muito atrativa para o método da Nuvem de Partículas.

6.5.1 Tratamento das Restrições

Assim como a maioria dos métodos bio-inspirados, o PSO foi proposto inicialmente para a solução de problemas de otimização sem restrições. Contudo, as estratégias de penalidade descritas anteriormente e utilizadas para o AG, podem ser utilizadas sem problemas para o tratamento das restrições de projeto.

Por outro lado, as restrições laterais (*bounds*) representando os limites inferior e superior de cada variável são tratados de forma diferenciada. O algoritmo implementado neste trabalho não permite que as partículas saiam do espaço de projeto, sendo aplicado um tratamento para restrições laterais utilizado por Clerc (2012)⁸⁸, na variante SPSO 2011. Logo, quando uma violação é identificada, a variável de projeto violada é corrigida para o valor do limite violado, sendo também aplicada modificação na componente da velocidade da partícula que violou a restrição, como ilustrado pela Figura 88.

6.5.2 PSO Híbrido para Otimização de Laminados

Este trabalho propõem um algoritmo híbrido para otimização de laminados. Neste algoritmo, a heurística básica do PSO é preservada, incluindo o comportamento cognitivo e social das partículas, mas são considerados operadores genéticos com objetivo de evitar a convergência prematura e melhorar a capacidade de busca local.

A posição da partícula é considerada uma matriz de inteiros, conforme apresentado no Item 6.1. Por outro lado, a velocidade da partícula é representada por uma matriz de números reais. Um arredondamento para o inteiro mais próximo é realizado durante a atualização da posição da partícula, expressa pela Equação (146). A codificação do problema de laminados é realizada da mesma forma utilizada no AG (Figura 79).

Os operadores genéticos de mutação e os operadores especiais para laminados, troca de lâmina, adição e eliminação de lâmina, são aplicados na posição da partícula após a etapa de atualização da posição. O pseudo-código do algoritmo resultante é apresentado na Figura 89.

Figura 89 – Pseudo-código do algoritmo PSO híbrido proposto.

```

Cria os vetores  $\mathbf{x}$ ,  $\mathbf{x}_p$  e  $\mathbf{v}$ ;
Inicialização( $\mathbf{x}, \mathbf{v}$ );
Avaliação( $\mathbf{x}$ );
Penalidade( $\mathbf{x}$ );

 $\mathbf{x}_p \leftarrow \mathbf{x}$ ;

para cada iteração até  $N_{iter}$  faça
  para cada partícula  $i$  faça
    // Operadores PSO
     $\mathbf{x}_g \leftarrow \text{MelhorVizinhança}(\mathbf{x}, i)$ ;
     $\mathbf{v}^{(i)} \leftarrow \text{AvaliaçãoVelocidade}(\mathbf{x}^{(i)}, \mathbf{v}^{(i)}, \mathbf{x}_p^{(i)}, \mathbf{x}_g)$ ;
     $\mathbf{x}^{(i)} \leftarrow \mathbf{x}^{(i)} + \mathbf{v}^{(i)}$ ;

    // Operadores Genéticos
     $\mathbf{x}^{(i)} \leftarrow \text{Muta\c{c}\~{a}\~{o}}(\mathbf{x}^{(i)})$ ;

    // Operação de Laminados
     $\mathbf{x}^{(i)} \leftarrow \text{Troca}(\mathbf{x}^{(i)})$ ;
     $\mathbf{x}^{(i)} \leftarrow \text{Adi\c{c}\~{a}\~{o}}(\mathbf{x}^{(i)})$ ;
     $\mathbf{x}^{(i)} \leftarrow \text{Elimina\c{c}\~{a}\~{o}}(\mathbf{x}^{(i)})$ ;

    Avaliação( $\mathbf{x}^{(i)}$ );
    Penalidade( $\mathbf{x}^{(i)}$ );
    Atualiza( $\mathbf{x}_p^{(i)}$ );
  fim
fim

```

Fonte: Elaborada pelo autor.

Tabela 14 – Variantes dos algoritmos.

Variante	Topologia	Operadores Genéticos
<i>PSO - G</i>	<i>Global</i>	
<i>PSO - R</i>	<i>Ring</i>	
<i>PSO - S</i>	<i>Square</i>	
<i>PSO - GM</i>	<i>Global</i>	$[p_{mut}]$
<i>PSO - GMS</i>	<i>Global</i>	$[p_{mut}, p_{swap}]$
<i>PSO - GMAD</i>	<i>Global</i>	$[p_{mut}, p_{add}, p_{del}]$
<i>PSO - RS</i>	<i>Ring</i>	$[p_{swap}]$
<i>PSO - RAD</i>	<i>Ring</i>	$[p_{add}, p_{del}]$
<i>PSO - SS</i>	<i>Square</i>	$[p_{swap}]$
<i>PSO - SAD</i>	<i>Square</i>	$[p_{add}, p_{del}]$
<i>GA</i>		$[p_{mut}]$
<i>GA - S</i>		$[p_{swap}]$
<i>GA - AD</i>		$[p_{add}, p_{del}]$

6.5.3 Variantes

Muitas variantes dos algoritmos apresentados anteriormente podem ser obtidas combinando diferentes topologias e operadores genéticos. Estas variantes podem ser divididas em dois grandes grupos, o primeiro onde não são utilizados operadores específicos de laminados e o segundo onde estes operadores são utilizados.

As variantes do PSO sem o uso de operadores de laminados são: Topologia Global (*PSO - G*), Topologia Local (*PSO - R*), Topologia Quadrada (*PSO - S*) e Topologia Global com Mutação (*PSO - GM*). O método considerando o operador Troca de Lâmina, com Topologia Global e com Mutação (*PSO - GMS*), com Topologia Local (*PSO - RS*) e com Topologia Quadrada (*PSO - SS*). O método considerando os operadores de Adição de Lâmina e Eliminação de Lâmina com Topologia Global e Mutação (*PSO - GMAD*), com Topologia Local (*PSO - RAD*) e com Topologia Quadrada (*PSO - SAD*). É importante notar que o uso da Topologia Global com Mutação é motivado pelo problema de convergência prematura observado nesta topologia. Por outro lado, as outras topologias não apresentam este problema. Portanto, a Mutação não foi utilizado em conjunto com as Topologias Local e Quadrada.

Também são utilizadas variantes do Algoritmo Genético, sem operadores de laminados (*GA*), com operador de Troca de Lâmina (*GA - S*) e com operador de Adição de Lâmina e Eliminação de Lâmina (*GA - AD*). A Tabela 14 mostra as variantes dos métodos AG e PSO utilizadas neste trabalho.

6.6 Meta-otimização

O funcionamento de algoritmos heurísticos de otimização é influenciado de maneira significativa pelos parâmetros numéricos utilizados. Assim, uma comparação isenta das vari-

antes propostas anteriormente requer o uso do melhor conjunto de parâmetros para cada uma delas.

A maioria dos trabalhos utiliza valores de parâmetros fornecido na literatura referente ao método utilizado ou ajusta o valor destes parâmetros por tentativa e erro. Em alguns trabalhos^{89,90} a calibração dos parâmetros é realizada utilizando uma bateria de exemplos teste, normalmente compostos por funções explícitas não-lineares.

Apesar de serem importantes como referência inicial, não há garantias de que os parâmetros encontrados na literatura sejam adequados a diferentes problemas. Desta forma, a escolha dos parâmetros ótimos pode ser tratada como um problema de otimização, cujo objetivo é determinar o valor dos parâmetros que levam ao melhor desempenho do algoritmo. Esta otimização dos parâmetros do otimizador é conhecida na literatura como meta-otimização¹⁹.

A meta-otimização utiliza um algoritmo de otimização para calibrar os parâmetros de outro algoritmo na otimização de um conjunto de problemas. Normalmente se utiliza como função objetivo da meta-otimização o melhor valor da função objetivo encontrada pelo otimizador. Por se tratar de um problema não determinístico, são empregadas várias otimizações para obtenção da média das respostas. Por este motivo, o custo computacional requerido é elevado, necessitando em vários casos da utilização de computação paralela para processamento das meta-otimizações.

Grefenstette⁹¹ utilizou o AG para meta-otimizar os parâmetros do AG. Uma abordagem similar foi realizada por Meissner⁹², utilizando o PSO como meta-otimizador de outra instância de PSO. Pedersen¹⁹ realizou a meta-otimização do PSO para problemas de Redes Neurais. Smit⁹³ abordou alternativas para realização de meta-otimização em algoritmos evolutivos. É importante notar que não existe uma grande quantidade de trabalhos disponíveis na literatura sobre o processo de meta-otimização, devido ao elevado custo computacional deste procedimento.

6.6.1 *Relação entre tamanho da população x número de gerações*

Um dos problemas relacionados a otimização dos parâmetros é obter um relação entre população e geração, visto que esta relação altera o funcionamento da otimização. Para descrever esta relação, um novo parâmetro (ρ) foi criado para cada método de otimização. Esta variável relaciona o tamanho da população (N_{ind}) e número de avaliações (N_{aval}). Considerando que o número de avaliações é obtido por:

$$N_{aval} = N_{ind} \cdot N_{gen} \quad (148)$$

onde N_{gen} é o número de gerações. Sendo ρ a razão entre tamanho da população e número de gerações:

$$\rho = \frac{N_{ind}}{N_{gen}} \quad (149)$$

Pode-se substituir N_{gen} da Equação (149) na Equação (148), obtendo a expressão:

$$N_{ind} = \text{round}(\sqrt{\rho \cdot N_{aval}}) \quad (150)$$

Deste modo, pode-se calcular o tamanho da população em função do número de avaliações e da variável ρ . O número de gerações (ou iterações) é obtido pela Equação (145) no caso do AG e pela Equação (147) no caso do PSO.

6.6.2 Algoritmo de meta-otimização

Este trabalho utilizou o PSO como meta-otimizador, considerando a Topologia Local e os parâmetros definidos em Bratton e Kennedy (2007), o PSO padrão. Os parâmetros dos algoritmos de otimização foram tratados como variáveis discretas.

A função objetivo da meta otimização pode ser considerada como a relação entre a média das funções objetivo obtida em N_{opt} otimizações e a função objetivo ótima do problema (f_{best}):

$$Mf_{obj} = \frac{\sum_{i=1}^{N_{opt}} f_{obj}}{f_{best}} \quad (151)$$

onde Mf_{obj} é o valor da meta-função objetivo. É importante notar que a Equação (151) considera que os problemas sejam de maximização da função objetivo. No caso de minimização, a meta-função objetivo é avaliada como:

$$Mf_{obj} = \frac{f_{best}}{\frac{\sum_{i=1}^{N_{opt}} f_{obj}}{N_{opt}}} \quad (152)$$

A meta-otimização pode ser aplicada para vários problemas utilizando a média das meta-funções objetivo de cada problema, desta forma a meta-função objetivo resultante é:

$$\overline{Mf}_{obj} = \frac{\sum_{i=1}^{N_{prob}} Mf_{obj}}{N_{prob}} \quad (153)$$

onde \overline{Mf}_{obj} é a meta-função objetivo referente à classe de problemas considerada e N_{prob} é o número de problemas considerados.

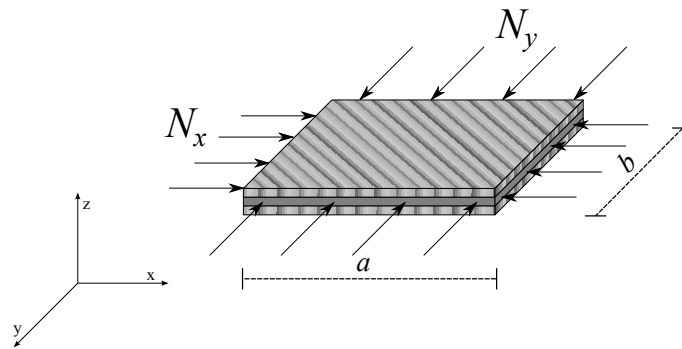
6.6.3 Resultados

Neste item são apresentados os resultados da calibração dos parâmetros via meta-otimização. Parte destes resultados foram obtidos no artigo publicado no CILAMCE 2014⁹⁴.

Inicialmente, foi realizado um estudo sobre o processo de meta-otimização com objetivo de analisar a influência do número de avaliações (N_{aval}) e do número de variáveis de projeto sobre a meta-função objetivo e os parâmetros calibrados de cada método. Neste estudo foram considerados as variantes *GA*, *GA - S*, *PSO - R* e *PSO - GMS*. Também foi considerado o Método da Busca Aleatória (*Random Search - RS*).

O problema de otimização considerados neste estudo consiste na maximização do fator de carga de uma placa retangular simplesmente apoiada com carregamento biaxial, como mostrado na Figura 90. A função objetivo é o menor fator de segurança considerando a flambagem e o critério de máxima deformação^{6,8,66}. Uma restrição que limita o número máximo de camadas adjacentes de mesmo ângulo é considerada.

Figura 90 – Placa laminada sujeita a carregamento biaxial.



Fonte: Elaborada pelo autor.

A verificação da estabilidade é importante para o caso de problemas de placas esbeltas. No caso de placas laminadas simplesmente apoiadas, o fator de segurança devido à flambagem (λ_b) pode ser obtido por²¹:

$$\frac{\lambda_b(p, q)}{\pi^2} = \frac{D_{11} \left(\frac{p}{a}\right)^4 + 2(D_{12} + 2D_{66}) \left(\frac{p}{a}\right)^2 \left(\frac{q}{b}\right)^2 + D_{22} \left(\frac{q}{b}\right)^4}{\left(\frac{p}{a}\right)^2 N_x + \left(\frac{q}{b}\right)^2 N_y} \quad (154)$$

onde p e q são o número de meias ondas nos eixos x e y , respectivamente. Este cálculo foi realizado considerando os valores p e q variando de 1 até 20, sendo o fator de segurança à flambagem o menor valor obtido.

O Critério de Máxima Deformação discutido no Item 2.1.1 pode ser aplicado ao problema de placas laminadas considerando apenas as deformações ϵ_1'' , ϵ_2'' e γ_{12}'' . Assim, o

critério é calculado no topo e na base de cada lâmina k por:

$$\lambda_s = \min_k \left(\min \left(\frac{\epsilon_1''}{S_f |\epsilon_1^k|}, \frac{\epsilon_2''}{S_f |\epsilon_2^k|}, \frac{\gamma_{12}''}{S_f |\gamma_{12}^k|} \right) \right) \quad (155)$$

onde o coeficiente de segurança S_f é igual a 1.5 e as deformações últimas são apresentadas na Tabela 15.

O fator de carga da estrutura é definido pelo menor fator obtido entre os dois critérios considerados:

$$\lambda = \min(\lambda_b, \lambda_s) \quad (156)$$

Tabela 15 – Propriedades do material utilizado na meta-otimização.

	E_1 (GPa)	E_2 (GPa)	G_{12} (GPa)	ν_{12}	Deformações últimas		
					ϵ_1''	ϵ_2''	γ_{12}''
Grafite-Epóxi	138	9.0	7.1	0.3	0.008	0.029	0.015

Fonte: Elaborada pelo autor.

O problema de otimização é definido matematicamente como:

$$\begin{aligned} & \text{Determinar } \mathbf{x} = [\theta_1, \theta_2, \dots, \theta_{N_v}] \text{ que} \\ & \text{Maximiza } \lambda \\ & \text{Sujeito a} \\ & N_{cp} \leq cp_{max} \end{aligned} \quad (157)$$

onde N_{cp} é o maior número de camadas adjacentes (*contiguous plies*) com a mesma orientação presentes no laminado e cp_{max} é o valor máximo aceitável para este parâmetro.

Considerando o problema de otimização de maximização de fator de carga abordado, 28 problemas diferentes serão considerados em cada meta-otimização, estes problemas serão obtidos variando a relação b/a e N_y/N_x do problema. Os valores de a e N_x serão fixos, respectivamente iguais a 0.508mm e 175 N/m. Os valores de b e N_y serão definidos de acordo com as relação utilizada. Serão considerandos os seguintes valores para a relação b/a : 0.25; 0.50; 0.75; 1.00; 2.00; 3.00; 4.00 e os seguintes valores para a relação N_y/N_x : 0.25; 0.50; 0.75; 1.00.

A laminação é simétrica-balanceadas com 48 lâminas, dando um total de 12 variáveis de projeto. O número máximo de camadas adjacentes de mesmo ângulo (cp_{max}) é 4. A espessura de cada lâmina é fixa com valor de 0.127mm e o material é grafite-epóxi, apresentado na Tabela 15. As meta-otimizações foram executadas considerando três casos de variação de ângulo das lâminas. No *Caso 1* é considerado uma variação $\Delta\theta = 45^\circ$, resultando em 3 ângulos disponíveis (0° , 45° e 90°). No *Caso 2* é considerado uma variação $\Delta\theta = 15^\circ$, resultando em 7

ângulos disponíveis ($0^\circ, 15^\circ, \dots, 75^\circ, 90^\circ$). No *Caso 3* é considerado uma variação $\Delta\theta = 5^\circ$, resultando em 19 ângulos disponíveis ($0^\circ, 10^\circ, \dots, 85^\circ, 90^\circ$).

O número de otimizações (N_{opt}) utilizado em cada avaliação dos indivíduos da meta-otimização é 100. Em função do custo computacional requerido, todas as meta-otimizações foram realizadas em um Cluster da marca SGI, que possui um total de 144 núcleos de processamento AMD Opteron (TM) 6234 de 2,4 MHz de frequência.

Os parâmetros utilizados pelas variantes são definidos pelas lista de valores apresentados na Tabela 16.

Tabela 16 – Valores de cada variável-parâmetro da meta-otimização.

Parâmetro	Valores
ρ	[0.50, 0.51, ..., 1.99, 2.00]
c_r	[0.10, 0.11, ..., 0.99, 1.00]
p_{mut}	[0.000, 0.000, ..., 0.9999, 1.0000]
w	[0.000, 0.001, ..., 1.9999, 2.0000]
c_1	[0.000, 0.001, ..., 3.9999, 4.0000]
c_2	[0.000, 0.001, ..., 3.9999, 4.0000]
p_{swap}	[0.000, 0.000, ..., 0.9999, 1.0000]

Fonte: Elaborada pelo autor.

A aplicação da Equação (151) e Equação (153) necessita do valor ótimo da função objetivo de cada problema. Como o número de possibilidades de cada caso é elevado, tornando inviável a obtenção dos valores por busca exaustiva, os valores foram determinados pela aplicação do algoritmo GA - S, utilizando um número de avaliações suficiente para obter uma taxa de sucesso maior que 90% ao longo de 100 otimizações.

A taxa de sucesso é definida pelo o número de otimizações que encontraram o mínimo global do problema (N_g) dividido pelo número total de otimizações (N_{opt}):

$$R(\%) = 100 \frac{N_g}{N_{opt}} \quad (158)$$

Tabela 17 – Funções objetivos ótimas nos problemas do *Caso 1* e *Caso 2*.

b/a	N_y/N_x				b/a	N_y/N_x			
	0.25	0.50	0.75	1.00		0.25	0.50	0.75	1.00
0.25	12690.7	10007.8	7967.54	6617.98	0.25	13442.5	10484.2	8357.13	6900.40
0.50	3447.57	2550.72	2024.23	1677.01	0.50	3564.13	2699.46	2114.29	1737.56
0.75	1517.08	1160.12	939.145	788.882	0.75	1569.56	1200.25	971.631	816.170
1.00	936.839	780.699	669.171	585.525	1.00	936.839	780.699	669.171	585.525
2.00	494.275	466.815	442.246	419.251	2.00	531.549	501.845	466.904	434.391
3.00	465.200	452.958	441.344	413.043	3.00	522.398	499.792	464.401	430.641
4.00	455.328	448.430	441.737	413.624	4.00	519.543	499.766	464.814	431.275

a) *Caso 1* ($\Delta\theta = 45^\circ$).b) *Caso 2* ($\Delta\theta = 15^\circ$).

Fonte: Elaborada pelo autor.

Tabela 18 – Funções objetivos ótimas nos problemas do *Caso 3* ($\Delta\theta = 05^\circ$).

b/a	N_y/N_x			
	0.25	0.50	0.75	1.00
0.25	13565.4	10569.2	8418.92	6930.55
0.50	3625.19	2710.36	2126.99	1742.42
0.75	1580.19	1208.38	978.211	821.697
1.00	936.839	780.699	669.171	585.525
2.00	535.302	502.159	467.022	435.606
3.00	531.353	500.191	464.681	432.914
4.00	530.394	500.136	464.865	433.159

Fonte: Elaborada pelo autor.

A Tabela 17 e a Tabela 18 mostram valores obtidos em cada caso. Em todos os casos, o fator de segurança relativo a flambagem passou a ser predominante a partir dos valores $b/a = 0.50$.

Foram utilizados para realização das meta-otimizações os seguintes parâmetros: 48 partículas e 50 iterações. Cada caso foi executado considerando dois números de avaliações: 300 e 1200 para o *Caso 1*, 1200 e 6000 para o *Caso 2* e 6000 e 30000 para o *Caso 3*. A mesma meta-função objetivo, considerando o mesmo caso e número de avaliações foi obtida pelo Método de Busca Aleatória (RS).

A Tabela 19 mostra os resultados das meta-função objetivos obtidos nas meta-otimizações realizadas. A Tabela 20 mostra o acréscimo percentual em relação à busca aleatória obtido por cada algoritmo.

Tabela 19 – Resultados das meta-funções objetivos obtidas

	Caso 1		Caso 2		Caso 3	
	$N_{aval} = 300$	$N_{aval} = 1200$	$N_{aval} = 1200$	$N_{aval} = 6000$	$N_{aval} = 6000$	$N_{aval} = 30000$
<i>RS</i>	0.946302	0.970053	0.954571	0.969717	0.966449	0.976845
<i>GA</i>	0.990593	0.998937	0.997824	0.999528	0.99889	0.999615
<i>GA - S</i>	0.995709	0.999870	0.998991	0.999842	0.999390	0.999824
<i>PSO - R</i>	0.974951	0.991226	0.996152	0.999041	0.999039	0.999622
<i>PSO - GMS</i>	0.994247	0.999786	0.999062	0.999797	0.999478	0.999759

Fonte: Elaborada pelo autor.

Uma importante observação é que o valor ótimo da meta-função objetivo é 1.0, para este valor todas as funções objetivo médias obteriam o valor máximo presente nas Tabelas 17 e 18, significando 100% de taxa de sucesso na otimização de cada problema.

Tabela 20 – Acréscimo percentual em relação a busca aleatória.

	Caso 1		Caso 2		Caso 3	
	$N_{aval} = 300$	$N_{aval} = 1200$	$N_{aval} = 1200$	$N_{aval} = 6000$	$N_{aval} = 6000$	$N_{aval} = 30000$
<i>GA</i>	4.68%	2.98%	4.53%	3.07%	3.36%	2.33%
<i>LamGA</i>	5.22%	3.07%	4.65%	3.11%	3.41%	2.35%
<i>PSO-R</i>	3.03%	2.18%	4.36%	3.02%	3.37%	2.33%
<i>LamPSO-GM</i>	5.07%	3.07%	4.66%	3.10%	3.42%	2.35%

Fonte: Elaborada pelo autor.

Observando os resultados apresentados na Tabela 19, considerando um mesmo caso, o aumento das avaliações permite aumenta o valor das meta-funções objetivo. Também é visível que o aumento das avaliações diminui o acréscimo percentual em relação a busca aleatória, o que sugere que para um $N_{aval} \simeq \infty$, todos os algoritmos terão rendimento igual ao da Busca Aleatória. Outra importante observação é de que considerando um mesmo número de avaliações, o aumento da dificuldade dos problemas (maior número de θ disponíveis) diminui o valor da meta-função objetivo, porém aumenta o acréscimo em relação a Busca Aleatória, o que sugere que um aumento na dificuldade do problema dificulta de forma mais acentuada a Busca Aleatória do que os outros métodos estudados.

Analisando os resultados de cada método estudado, observamos que a utilização do operador Troca de Lâmina para o Algoritmo Genético melhora os resultados obtidos. O método *GA - S* apresentou melhores resultados para o *Caso 1*, porém nos *Caso 2* e *Caso 3* obteve resultados comparáveis ao método *PSO-GMS*. A mesma conclusão é observada comparando os resultados dos métodos *GA* e *PSO - R*. As Tabela 21, Tabela 22, Tabela 23 e Tabela 24 mostram os parâmetros ótimos obtidos por cada método, bem como as suas médias e desvio padrão obtidos.

Tabela 21 – Parâmetros ótimos para o método *GA*.

$\Delta\theta$	Avaliações	Parâmetros <i>GA</i>		
		ρ	c_r (%)	p_{mut} (%)
45°	300	1.16	40	26.86
45°	1200	1.48	27	30.96
15°	1200	1.26	40	18.62
15°	6000	1.85	32	23.51
5°	6000	1.21	77	17.83
5°	30000	1.91	90	14.11
Média (Desvio Padrão)		1.48 (0.33)	51.00 (25.98)	22.17 (6.31)

Fonte: Elaborada pelo autor.

Tabela 22 – Parâmetros ótimos para o método *GA - S*.

$\Delta\theta$	Avaliações	Parâmetros <i>GA - S</i>			
		ρ	c_r (%)	p_{mut} (%)	p_{swap} (%)
45°	300	0.56	23	9.42	16.44
45°	1200	0.84	69	8.19	14.15
15°	1200	0.76	23	10.09	11.53
15°	6000	1.21	17	11.29	20.85
5°	6000	1.00	49	9.64	12.99
5°	30000	1.78	24	7.5	24.17
Média (Desvio Padrão)		1.00 (0.43)	34.17 (20.40)	9.33 (1.21)	16.33 (4.88)

Fonte: Elaborada pelo autor.

Tabela 23 – Parâmetros ótimos para o método *PSO - R*.

$\Delta\theta$	Avaliações	Parâmetros <i>PSO - R</i>			
		ρ	w	c_1	c_2
45°	300	0.60	0.7453	1.2165	1.4698
45°	1200	0.68	0.6384	0.9094	1.4806
15°	1200	0.53	0.7616	0.7217	1.0227
15°	6000	0.58	0.5678	0.6701	2.1084
5°	6000	0.55	0.6463	0.3447	1.4108
5°	30000	0.57	0.5208	0.5723	1.579
Média (Desvio Padrão)		0.59 (0.05)	0.65 (0.10)	0.74 (0.30)	1.51 (0.35)

Fonte: Elaborada pelo autor.

Tabela 24 – Parâmetros ótimos para o método *PSO - GMS*.

$\Delta\theta$	Avaliações	Parâmetros <i>PSO - GMS</i>					
		ρ	w	c_1	c_2	p_{mut} (%)	p_{swap} (%)
45°	300	0.60	0.3557	0.3793	1.3043	4.97	9.98
45°	1200	0.75	0.2717	0.4422	0.9700	4.84	14.74
15°	1200	0.52	0.3345	0.6997	0.7883	4.16	4.15
15°	6000	1.36	0.1310	0.7503	0.7513	6.56	8.28
5°	6000	1.12	0.2582	0.4827	0.8369	2.38	4.38
5°	30000	1.55	0.0423	0.7620	0.6106	8.23	4.16
Média (Desvio Padrão)		0.98 (0.42)	0.23 (0.12)	0.59 (0.17)	0.88 (0.24)	5.17 (2.14)	7.50 (4.46)

Fonte: Elaborada pelo autor.

Observando os resultados dos parâmetros obtidos, verifica-se que alguns apresentam pouca variação, como é o caso da taxa de mutação (p_{mut}) para o métodos *GA - S*, ou do parâmetro ρ nos métodos *PSO - R* e *PSO - GMS*. Entretanto, pôde-se observar que o número de avaliações influencia de maneira significativa os parâmetros dos algoritmos.

Por este motivo foi decidido que nos próximos procedimentos de calibração será utilizado um número fixo de 1000 avaliações, e será realizada uma calibração para laminados tradicionais e uma para laminados dispersos, com $\Delta\theta = 15^\circ$. Também foi realizada uma meta-otimização para o caso de problema de minimização de peso, considerando apenas laminados tradicionais.

Nestes novos procedimentos de calibração foi considerado todos os algoritmos citados na Tabela 14. A lista de valores das variáveis de meta-otimização abordada na Tabela 16 foi alterada, sendo considerado uma nova faixa de valores apresentada na Tabela 25. Esta nova tabela apresenta valores menos espaçados.

É importante notar que no caso do problema de minimização, as probabilidades de atuação dos operadores de Adição e Eliminação de lâmina são únicas, representando apenas 1 parâmetro de meta-otimização. Também é importante citar que os operadores de Adição e Eliminação de Lâmina não são considerados na calibração de problemas de maximização.

Tabela 25 – Novos limites das variável-parâmetro das meta-otimização.

Parâmetro	Valores
ρ	[0.25, 0.26, ..., 3.99, 4.00]
c_r	[0.40, 0.45, ..., 0.95, 1.00]
w	[0.00, 0.01, ..., 1.99, 2.00]
c_1	[0.00, 0.01, ..., 3.99, 4.00]
c_2	[0.00, 0.01, ..., 3.99, 4.00]
p_{mut}	[0.00, 0.01, ..., 0.49, 0.50]
p_{swap}	[0.00, 0.01, ..., 0.49, 0.50]
$p_{add/del}$	[0.00, 0.01, ..., 0.49, 0.50]

Fonte: Elaborada pelo autor.

As Tabelas 26 e 27 mostram os parâmetros ótimos obtidos no processo de meta-otimização para o caso de laminados tradicionais. A Tabela 28 mostra os valores das meta-função objetivo encontrados para o caso de laminados tradicionais. Os resultados referentes aos laminados dispersos são apresentados nas Tabelas 29, 30 e 31.

Tabela 26 – Parâmetros ótimos dos métodos PSO para caso de problemas de maximização ($\Delta\theta = 45^\circ$).

Algoritmo	ρ	w	c_1	c_2	p_{mut}	p_{swap}
<i>PSO - G</i>	3.17	0.89	1.49	0.45		
<i>PSO - GM</i>	0.31	0.56	1.35	0.51	19%	
<i>PSO - R</i>	0.25	1.02	1.10	2.76		
<i>PSO - S</i>	2.83	0.38	1.39	2.21		
<i>PSO - GMS</i>	1.50	0.14	1.22	1.28	5%	12%
<i>PSO - RS</i>	0.26	0.33	1.29	2.10		3%
<i>PSO - SS</i>	0.54	0.38	1.22	1.45		6%

Fonte: Elaborada pelo autor.

Tabela 27 – Parâmetros ótimos dos métodos AG para caso de problemas de maximização ($\Delta\theta = 45^\circ$).

Algoritmo	ρ	c_r	p_{mut}	p_{swap}
<i>GA</i>	1.91	40%	31%	
<i>GA - S</i>	0.37	50%	9%	18%

Fonte: Elaborada pelo autor.

Tabela 28 – Meta-função objetivo ótimas para o caso de problemas de maximização ($\Delta\theta = 45^\circ$)

Algoritmo	\overline{Mf}_{obj} ótima
<i>GA</i>	0.9984
<i>PSO - G</i>	0.9979
<i>PSO - GM</i>	0.9988
<i>PSO - R</i>	0.9964
<i>PSO - S</i>	0.9982
<i>PSO - GMS</i>	0.9997
<i>PSO - RS</i>	0.9989
<i>PSO - SS</i>	0.9995
<i>GA - S</i>	0.9998

Fonte: Elaborada pelo autor.

Tabela 29 – Parâmetros ótimos dos métodos PSO para caso de problemas de maximização ($\Delta\theta = 15^\circ$).

Algoritmo	ρ	w	c_1	c_2	P_{mut}	P_{swap}
<i>PSO - G</i>	1.54	0.74	1.37	0.59		
<i>PSO - GM</i>	0.26	0.62	0.88	0.62	6%	
<i>PSO - R</i>	0.25	0.81	0.76	0.77		
<i>PSO - S</i>	0.54	0.76	1.22	0.71		
<i>PSO - GMS</i>	0.27	0.19	0.50	0.91	4%	6%
<i>PSO - RS</i>	0.25	0.66	0.47	0.91		2%
<i>PSO - SS</i>	0.26	0.64	0.59	0.73		3%

Fonte: Elaborada pelo autor.

Tabela 30 – Parâmetros ótimos dos métodos AG para caso de problemas de maximização ($\Delta\theta = 15^\circ$).

Algoritmo	ρ	c_r	P_{mut}	P_{swap}
<i>GA</i>	1.58	40%	19%	
<i>GA - S</i>	0.26	75%	9%	8%

Fonte: Elaborada pelo autor.

Tabela 31 – Meta-função objetivo ótimas para o caso de problemas de maximização. ($\Delta\theta = 15^\circ$).

Algoritmo	\overline{Mf}_{obj} ótima
<i>GA</i>	0.9975
<i>PSO - G</i>	0.9980
<i>PSO - GM</i>	0.9986
<i>PSO - R</i>	0.9981
<i>PSO - S</i>	0.9984
<i>PSO - GMS</i>	0.9990
<i>PSO - RS</i>	0.9981
<i>PSO - SS</i>	0.9988
<i>GA - S</i>	0.9988

Fonte: Elaborada pelo autor.

Pode-se observar que os métodos *PSO - GMS*, *PSO - SS* e *GA - S* obtiveram os melhores resultados das meta-função objetivo.

No caso da calibração dos problemas de minimização, um problema abordado em Lopes et al. (2009) foi considerado. Este problema consiste na otimização de uma placa quadrada apoiada e sujeita a uma carga biaxial e de cisalhamento, como ilustrado pela Figura 91. A função objetivo é minimizar o peso da placa laminada, considerando o critério de Tsai-Wu abordado no Item 2.1.1. O problema de otimização é definido como:

Achar $\mathbf{x} = [(e_1, \theta_1), (e_2, \theta_2), \dots, (e_{N_v}, \theta_{N_v})]$ que

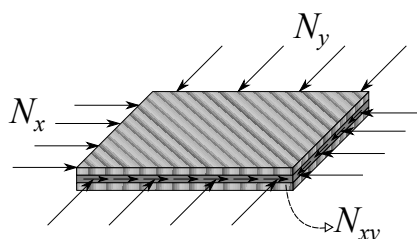
Minimize W

Sujeito a

$S_f \geq 1$

onde W é o peso do laminado.

Figura 91 – Placa Laminada com carregamento biaxial e de cisalhamento.



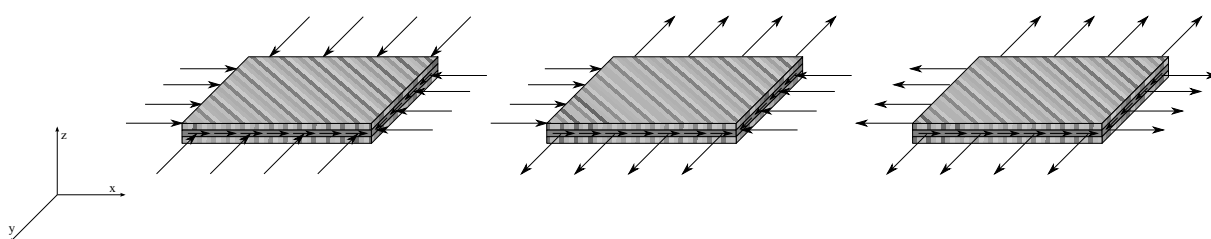
Fonte: Elaborada pelo autor.

É importante lembrar que o operador de Troca de Lâmina não foi considerado nestes problemas. Como não existem momentos envolvidos e a flambagem não é considerada, não

existe efeito nenhum em trocar posições das camadas sobre as tensões no compósito, consequentemente não haverá mudança no critério de falha avaliado.

A calibração para problemas de minimização considerou um conjunto de 9 problemas, sendo estes formados por vários casos de carga distintos. A placa quadrada possui dimensão igual a 1.0 m. O esquema de laminação considerado é simétrico e balanceado. A espessura de cada camada é 0.1 mm. As propriedades elásticas do material compósito são $E_1 = 116.600$ MPa, $E_2 = 7673$ MPa, $G_{12} = 4173$ MPa, coeficiente de Poisson $\nu_{12} = 0.27$ e densidade de massa igual a 1605 kg/m³. As propriedades de resistência do material são presentes na Tabela 32.

Figura 92 – Casos de carga biaxial considerados.



Fonte: Elaborada pelo autor.

Tabela 32 – Propriedades de resistência do compósito considerado nas meta-otimizações de minimização.

X_T	X_C	Y_T	Y_C	S_{21}
2062 MPa	1701 MPa	70 MPa	240 MPa	105 MPa

Fonte: Elaborada pelo autor.

Nove problemas diferentes são considerados na meta-otimização. Cada problema é obtido combinando três cargas biaxiais (ver Figura 92) e três cargas de cisalhamento. Tais casos de carga estão apresentados na Tabela 33, sendo também mostrado o valor mínimo da função objetivo obtida em cada caso, de modo análogo ao que foi feito no exemplo de maximização. Os resultados da meta-otimização estão apresentados na Tabela 35.

Tabela 33 – f_{best} para os problemas de minimização de peso.

N_x, N_y (MN/m)	N_{xy} (MN/m)		
	0.0	0.5	1.0
$N_x = 2, N_y = 2$	75.576	81.874	94.470
$N_x = 2, N_y = -2$	62.980	81.874	100.77
$N_x = -2, N_y = -2$	31.490	37.788	44.086

Fonte: Elaborada pelo autor.

Tabela 34 – Parâmetros ótimos dos métodos PSO para caso de problemas de minimização.

Algoritmo	ρ	w	c_1	c_2	p_{mut}	$p_{add} = p_{del}$
<i>PSO - G</i>	0.77	1.01	1.03	0.3		
<i>PSO - GM</i>	0.3	0.32	3.19	0.7	5%	
<i>PSO - R</i>	0.25	1.01	1.02	0.51		
<i>PSO - S</i>	0.51	1.01	0.67	0.64		
<i>PSO - GMAD</i>	0.36	0.36	2.4	0.67	4%	2%
<i>PSO - RAD</i>	0.27	0.27	3.38	0.78		4%
<i>PSO - SAD</i>	0.36	0.36	2.43	0.69		4%

Fonte: Elaborada pelo autor.

Tabela 35 – Parâmetros ótimos dos métodos AG para caso de problemas de minimização.

Algoritmo	ρ	c_r	p_{mut}	$p_{add} = p_{del}$
<i>GA</i>	0.99	40 %	8%	
<i>GA - AD</i>	0.99	40 %	7%	0%

Fonte: Elaborada pelo autor.

Tabela 36 – Meta-função o objetivo ótimas para o caso de problemas de minimização.

Algoritmo	\overline{Mf}_{obj}
<i>GA</i>	0.9854
<i>PSO - G</i>	0.9739
<i>PSO - GM</i>	0.9811
<i>PSO - R</i>	0.9481
<i>PSO - S</i>	0.9588
<i>PSO - GMAD</i>	0.9820
<i>PSO - RAD</i>	0.9828
<i>PSO - SAD</i>	0.9887
<i>GA - AD</i>	0.9858

Fonte: Elaborada pelo autor.

Os resultados mostrados na Tabela 35 indicam que o método com maior valor da meta-função objetivo foi o *PSO - SAD*, seguido do *GA - AD* e *GA*. Os resultados também indicam que os operadores específicos de laminados melhoraram os resultados dos métodos, apesar de alguns algoritmos como o *PSO - GM* terem obtido bons resultados sem o uso destes operadores. Também é importante observar que a variante *GA - AD*, que é o *GA* com os operadores de laminados, teve seu parâmetro relacionado ao operador de laminados zerado. Isto indica que neste caso os operadores de Adição de Lâmina e Eliminação de Lâmina foram desnecessários.

6.7 Exemplos de Aplicação

A seguir são apresentados os exemplos de otimização de laminados utilizando os algoritmos de otimização e suas variantes discutidas nos itens anteriores.

6.7.1 Maximização de resistência de uma placa

O primeiro exemplo abordado é similar ao que foi apresentado no procedimento de calibração relacionado ao problema de maximização de fator de carga. As propriedades do material e as variáveis do problema são as mesmas discutidas anteriormente. A dimensão da placa é $a = 508$ mm e $b = 203.2$ mm ($b/a = 0.4$) e as cargas são $N_x = 175$ N/m e $N_y = 105$ N/m ($N_y/N_x = 0.6$).

O problema foi solucionado considerando laminados convencionais ($\Delta\theta = 45^\circ$) e laminados dispersos ($\Delta\theta = 15^\circ$ e $\Delta\theta = 5^\circ$). Os parâmetros utilizados no caso dos laminados convencionais são os apresentados nas Tabelas 26 e 27. No caso dos laminados dispersos, os parâmetros utilizados são os apresentados nas Tabelas 29 e 30.

As taxas de sucesso de cada método, bem com a média dos melhores valores de

função objetivo, foram analisadas. Estas análises foram realizadas considerando diferentes número de avaliações, realizando 100 otimizações para traçado dos resultados. O método PSO padrão⁸⁷ (*PSO-STD*) também foi utilizado nos exemplos. Este método utiliza a Topologia Local e considera um número fixo de 50 partículas.

No caso do laminado tradicional ($\Delta\theta = 45^\circ$), a solução ótima do problema encontrada foi $\lambda = 3568.58$, sendo este valor relacionado ao fator de flambagem. Diferentes projetos ótimos foram encontrados pelos algoritmos de otimização, algumas destas soluções são apresentadas na Tabela 37.

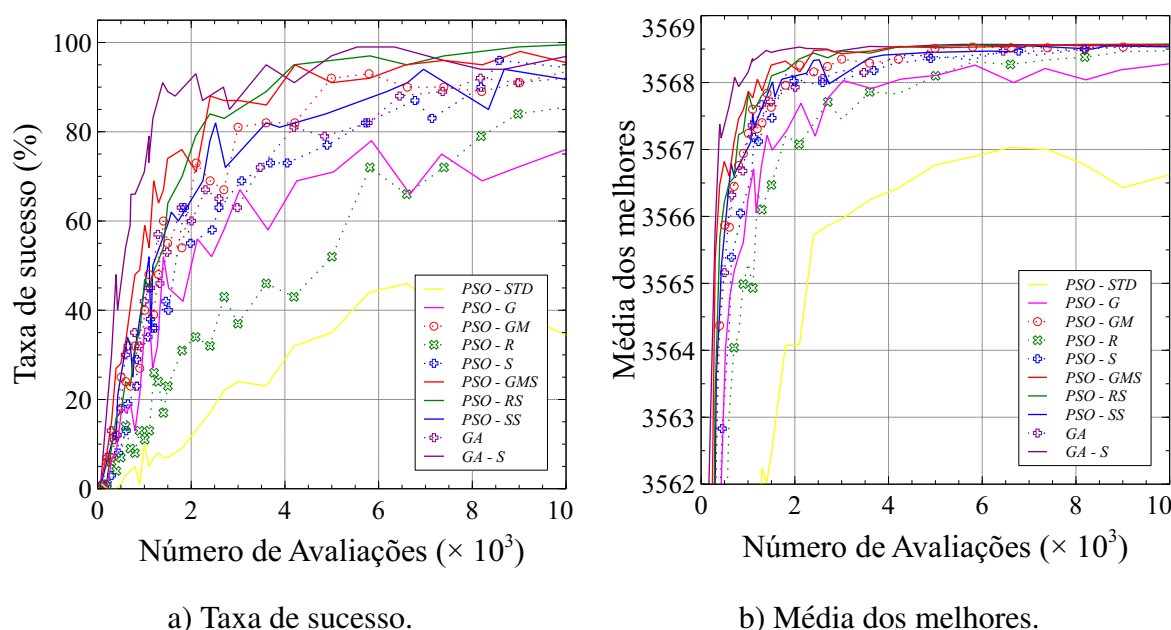
Tabela 37 – Esquemas de laminação ótimos encontrados no Exemplo 6.7.1. ($\Delta\theta = 45^\circ$)

Esquema de Laminação	λ_b	λ_s
$[(90_2^\circ \pm 45^\circ)_3 \pm 45_2^\circ (\pm 45^\circ 90_2^\circ)_2]_s$	3568.58	10534.30
$[\pm 45^\circ 90_4^\circ \pm 45_2^\circ 90_2^\circ (90_2^\circ \pm 45^\circ)_2 \pm 45_2^\circ]_s$	3568.58	10534.30
$[90_4^\circ \pm 45_4^\circ (90_2^\circ \pm 45^\circ)_2 \pm 45^\circ 90_2^\circ]_s$	3568.58	10534.30

Fonte: Elaborada pelo autor.

A Figura 93 mostra as curvas de taxa de sucesso e média das melhores partículas (ou indivíduos no AG) obtidas por número de avaliação nas otimizações. As melhores variantes para este problema de otimização foram o *GA - S*, *PSO - GMS* e *PSO - RS*. Pode-se observar que a consideração do operador Troca de Lâmina melhorou as respostas de todos os métodos. É importante notar que o desempenho das variantes do algoritmo proposto foi muito superior ao do algoritmo padrão do PSO (*PSO - STD*).

Figura 93 – Resultados obtidos do Exemplo 6.7.1, com $\Delta\theta = 45^\circ$.



a) Taxa de sucesso.

b) Média dos melhores.

Fonte: Elaborada pelo autor.

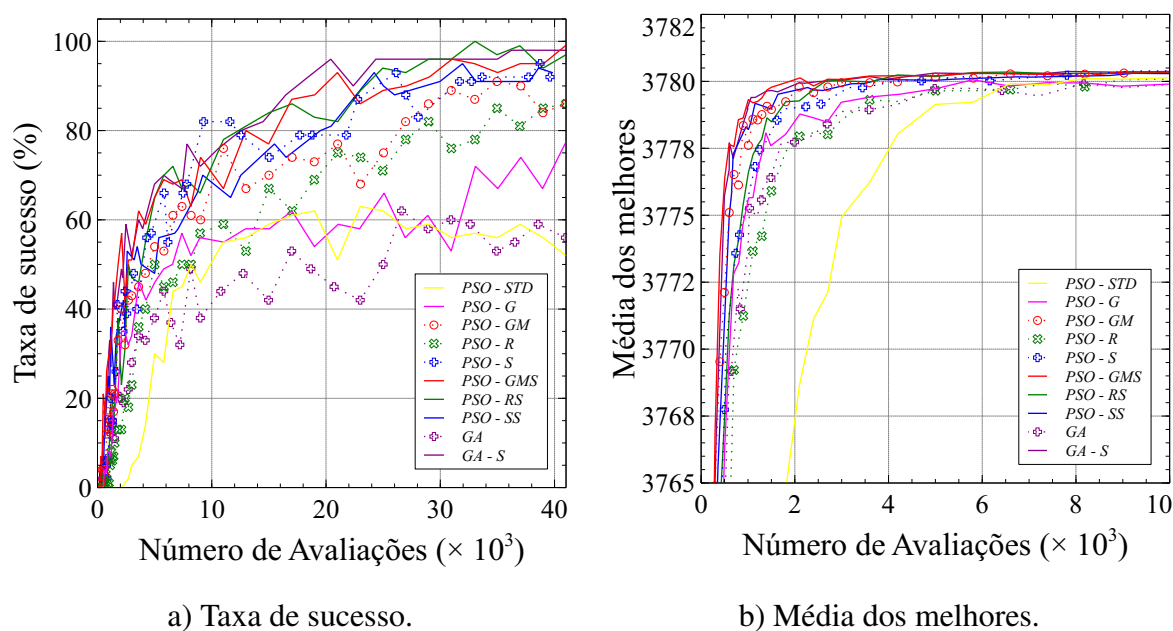
Tabela 38 – Esquemas de laminação ótimos encontrados no Exemplo 6.7.1. ($\Delta\theta = 15^\circ$)

Layup	λ_b	λ_s
$[\pm 60^\circ \pm 75^\circ \pm 60_2^\circ (\pm 60^\circ \pm 75^\circ)_2 \pm 60_4^\circ]_s$	3780.64	6238.82
$[\pm 60^\circ \pm 75^\circ \pm 60_2^\circ \pm 75^\circ \pm 60_4^\circ \pm 75^\circ \pm 60_2^\circ]_s$	3780.64	6238.82

Tabela 39 – Esquema de laminação ótimo encontrado no Exemplo 6.7.1. ($\Delta\theta = 5^\circ$)

Layup	λ_b	λ_s
$[\pm 65_4^\circ \pm 60^\circ \pm 65_2^\circ (\pm 60^\circ \pm 65^\circ)_2 \pm 70^\circ]_s$	3803.67	5807.15

No primeiro caso de laminado disperso ($\Delta\theta = 15^\circ$), a função objetivo ótima encontrada foi 3780.64. A Figura 94 mostra os resultados obtidos neste caso. A Tabela 38 mostra os esquemas de laminação ótimos obtidos.

Figura 94 – Resultados obtidos do Exemplo 6.7.1, com $\Delta\theta = 15^\circ$.

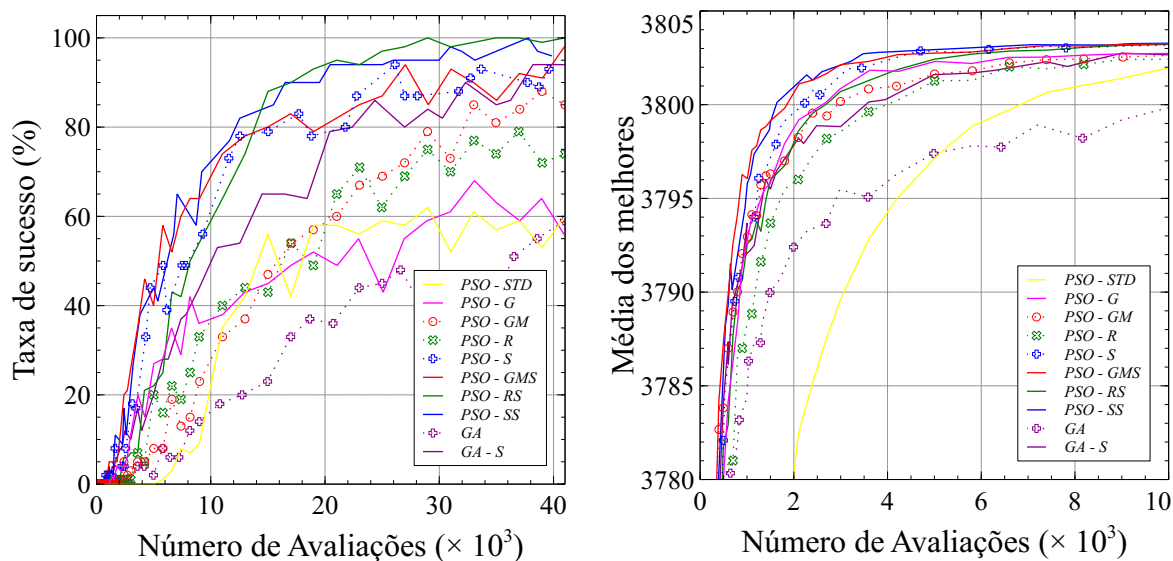
a) Taxa de sucesso.

b) Média dos melhores.

Fonte: Elaborada pelo autor.

Analisando os resultados, observamos que os melhores métodos foram os *PSO - GMS*, *GA - S* e *PSO - SS*. Pode-se notar que o desempenho das variantes do método PSO melhoraram em relação aos algoritmos genéticos analisados. Isto se deve ao fato de que o PSO foi desenvolvido para problemas contínuos, a medida que os intervalos $\Delta\theta$ vão diminuindo, o problema discreto se torna cada vez mais parecido com um problema contínuo.

No segundo caso de laminado disperso estudado ($\Delta\theta = 5^\circ$), foi obtido uma função objetivo ótima de 3803.67. Os resultados obtidos neste caso estão apresentados na Figura 95. A Tabela 38 mostra a laminação ótima obtida.

Figura 95 – Resultados obtidos do Exemplo 1, com $\Delta\theta = 5^\circ$.

a) Taxa de sucesso.

b) Média dos melhores.

Fonte: Elaborada pelo autor.

Os métodos *PSO - SS*, *PSO - GMS* e *PSO - RS* obtiveram os melhores resultados. Neste caso, várias variantes superaram o desempenho do método *GA - S*. Mais uma vez, o operador Troca de Lâmina apresentou melhorias significativas, em especial na Topologia Local. Também é importante ressaltar que o desempenho do algoritmo proposto se mostrou superior ao PSO padrão.

6.7.2 Minimização de peso de uma placa

O Exemplo 2 é um exemplo similar aos que foram utilizados na meta-otimização de minimização. As dimensões da placa, propriedade do material e as variáveis do problema são os mesmos descritos anteriormente. As cargas são $N_x = -2$ MN/m, $N_y = 6$ MN/m e $N_{xy} = 0.5$ MN/m. Foram considerados apenas laminados tradicionais ($\Delta\theta = 45^\circ$). Os parâmetros considerados são os presentes na Tabela 35.

O problema possui diversas soluções ótimas, com peso mínimo de 144.85 N (considerando $g = 9.81$ m/s²). É importante notar que a sequência de empilhamento das lâminas não muda o critério de falha para este problema, apenas o número de camadas de cada ângulo influencia. A Tabela 40 mostra alguns das soluções ótimas obtidas pelos algoritmos de otimização.

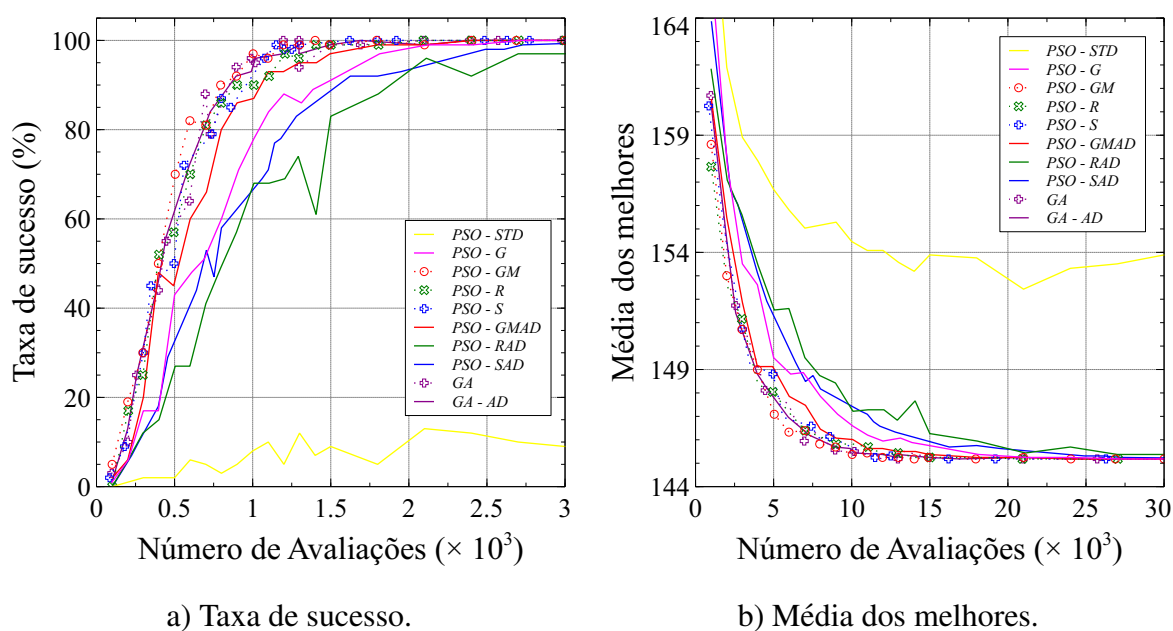
Tabela 40 – Número de camadas ótimo do Exemplo 02.

Número de camadas			Critério de falha de Tsai-Wu
0°	±45°	90°	
92	0	0	1.015
64	8	20	1.005
68	4	20	1.001
68	8	16	1.000

Fonte: Elaborada pelo autor.

Os resultados de taxa de sucesso e média das melhores soluções estão apresentadas na Figura 96.

Figura 96 – Placa Laminada com carregamento biaxial e de cisalhamento.



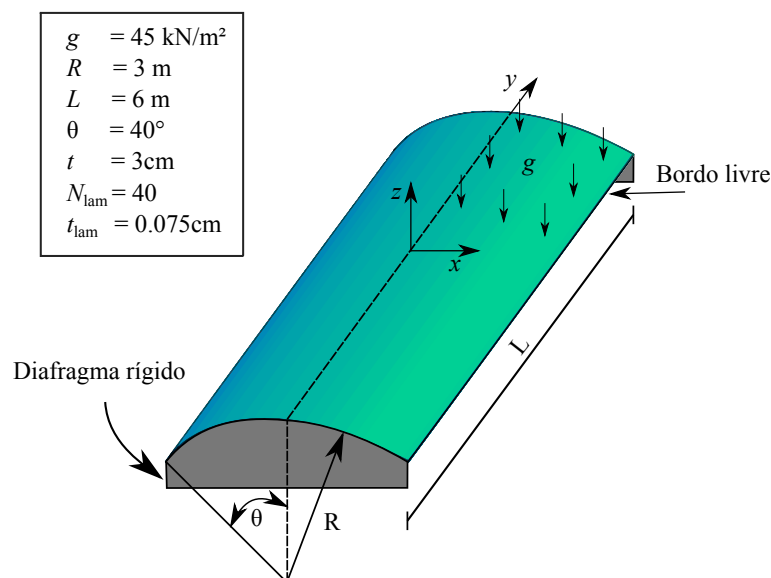
Fonte: Elaborada pelo autor.

Todas as variantes obtiveram a solução ótima do problema e apresentaram desempenho muito melhor que o do PSO padrão. É importante notar que o uso dos operadores de Adição de Lâmina e Deleção de Lâmina não levaram a melhoria de desempenho. Tais operadores não se mostram necessários, uma vez que o processo de decodificação apresentado na Seção 6.4.4 possibilita que a adição e eliminação de lâminas ocorram de forma natural pelos operadores de cruzamento no GA e atualização da posição da partícula no PSO.

6.7.3 Maximização de rigidez de uma casca cilíndrica

Este exemplo trata da otimização de uma casca laminada similar à estrutura abordada no Item 5.3.4. A geometria e carregamento da casca cilíndrica estão apresentados na Figura 97. O objetivo da otimização é maximizar a rigidez da estrutura, considerando uma espessura total de 30mm e um esquema de laminação simétrico e balanceado com 40 camadas.

Figura 97 – Descrição do Exemplo 6.7.3.



Fonte: Elaborada pelo autor.

A maximização da resistência da estrutura é obtida indiretamente pela minimização do deslocamento w da estrutura. Foi considerado um *grid* de 16×16 pontos paramétricos para avaliação do deslocamento em diferentes pontos do modelo. A resistência da estrutura é avaliada considerando o critério de falha de Tsai-Wu como uma restrição. Outra restrição limita o número máximo de camadas adjacentes de mesma orientação.

O problema de otimização é definido com:

Determinar $\mathbf{x} = [\theta_1, \theta_2, \dots, \theta_{N_v}]$ que

Minimize w_b

Sujeito a

$$S_{TW}/S_{req} - 1 \leq 0$$

$$N_{cp}/cp_{max} - 1 \leq 0$$

(159)

onde cp_{max} é o número de camadas adjacentes permitido, S_{TW} é fator de segurança obtido pelo critério de Tsai-Wu, calculado pela Equação (17) e S_{req} é o fator de segurança requerido. Neste exemplo, $S_{req} = 1.5$ e $cp_{max} = 4$.

O material compósito utilizado é o carbono-epóxi. As propriedades elásticas e as resistências do material estão respectivamente apresentadas nas Tabelas 41 e 42.

Tabela 41 – Propriedades elásticas do compósito carbono-epoxi²².

E_1 (GPa)	E_2 (GPa)	E_3 (GPa)	ν_{12}	ν_{13}	ν_{23}	G_{12} (GPa)	G_{13} (GPa)	G_{23} (GPa)
147	10.3	10.3	0.27	0.27	0.54	7.0	7.0	3.7

Fonte: Elaborada pelo autor.

Tabela 42 – Resistências do compósito carbono-epoxi²².

X_T (MPa)	X_C (MPa)	Y_T (MPa)	Y_C (MPa)	Z_T (MPa)	Z_C (MPa)	S_{12} (MPa)	S_{13} (MPa)	S_{23} (MPa)
2288	1725	57	228	57	228	76	114	114

Fonte: Elaborada pelo autor.

Os projetos foram analisados utilizando uma malha sólida isogeométrica de grau 2 em todas as direções e discretização 16x16, considerando uma análise linear do modelo completo (sem simetria). Foi considerado integração reduzida (2x2x2 pontos por lâmina) com quadratura de Gauss, resultando em 320 pontos de integração por elemento isogeométrico. O critério de falha foi calculado nos pontos de integração do modelo e o cálculo da restrição utiliza o menor fator de segurança entre todos os pontos de integração.

O problema foi otimizado considerando laminados tradicionais ($\Delta\theta = 45^\circ$) e laminados dispersos ($\Delta\theta = 15^\circ$) e ($\Delta\theta = 05^\circ$). As otimizações foram realizadas pelos métodos *GA - S*, *PSO - GMS*, *PSO - RS*, *PSO - SS* e *PSO-STD*. Foram realizadas 5 otimizações de cada método considerando 10000 avaliações. Os parâmetros utilizados para cada método foram os obtidos da calibração dos problemas de maximização, no caso dos laminados tradicionais, foi utilizado os parâmetros apresentados nas Tabelas 26 e 27. Já no caso dos laminados dispersos, foram utilizados os parâmetros apresentados nas Tabelas 29 e 30.

Todos os algoritmos obtiveram o mesmo projeto ótimo em todos os casos. A Tabela 43 apresenta os projetos ótimos, como também o deslocamento mínimo obtido e o fator de segurança S_{TW} . Pode-se observar que o fator de segurança aumentou 13% e 14% considerando laminados dispersos.

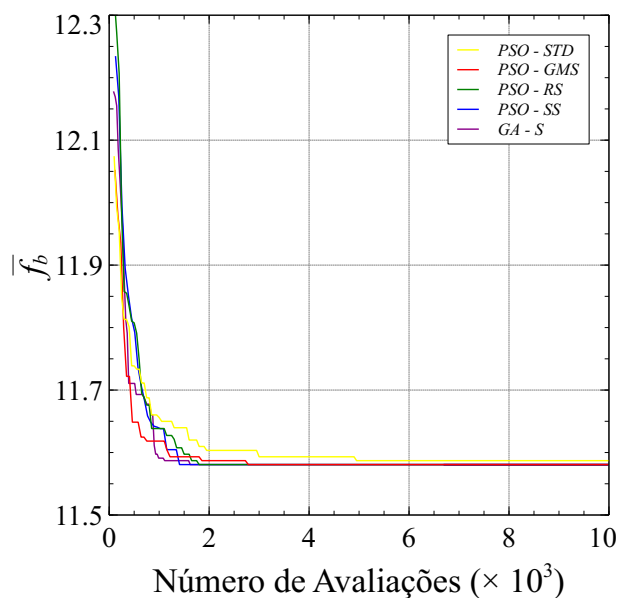
Tabela 43 – Resultados de projeto ótimo, menor deslocamento w e critério de falha obtidos para cada caso $\Delta\theta$ considerado.

$\Delta\theta$	Projeto ótimo	$w_b(cm)$	S_{TW}
45°	$[(90_4^\circ \pm 45^\circ)_2 0_4^\circ \pm 45^\circ 0_2^\circ]_s$	11.5807	2.12297
15°	$[90_4^\circ \pm 75_2^\circ \pm 60^\circ \pm 15^\circ 0_4^\circ \pm 15^\circ 0_2^\circ]_s$	11.0027	2.38941
5°	$[90_4^\circ \pm 80^\circ \pm 65^\circ \pm 55^\circ \pm 10^\circ 0_4^\circ \pm 5^\circ 0_2^\circ]_s$	10.9603	2.42405

Fonte: Elaborada pelo autor.

A Figura 98 mostra o valor médio da melhor função objetivo obtida ($\overline{f_b}$) em função do número de avaliações realizada durante a otimização, para $\Delta\theta = 45^\circ$. O método *PSO - SS* obteve melhor convergência que os outros métodos considerados.

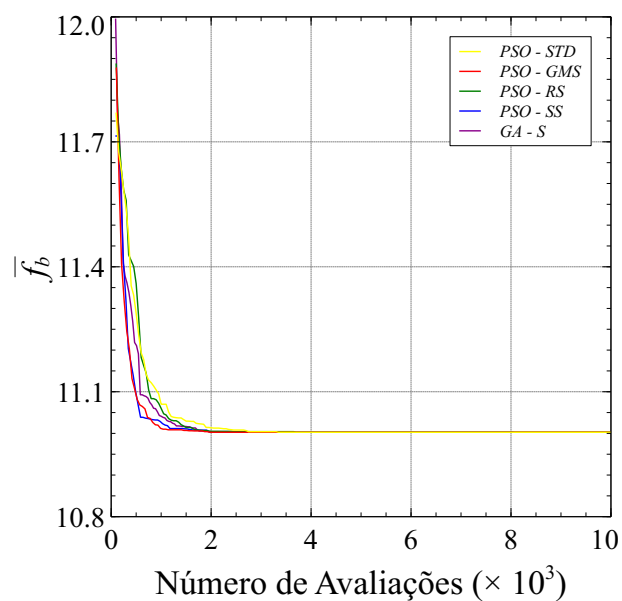
Figura 98 – Resultados de $\overline{f_b}$ por número de avaliações para o caso $\Delta\theta = 45^\circ$.



Fonte: Elaborada pelo autor.

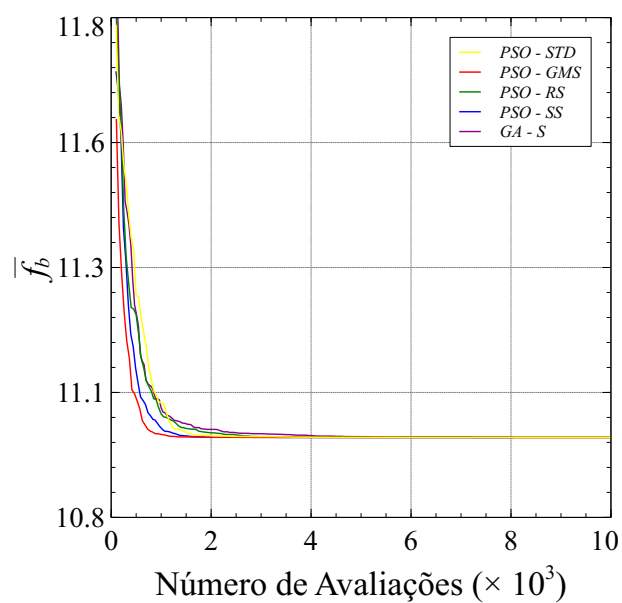
No caso de laminados dispersos, o método *PSO - GMS* obteve maior velocidade de convergência do que o *PSO - SS*, como mostram as Figuras 99 e 100. Em todos os casos, os algoritmos *PSO - SS* e *PSO - GMS* convergiram para resposta ótima mais rápido do que o *GA - S* e o *PSO* padrão. É importante ressaltar que no caso $\Delta\theta = 5^\circ$ o *PSO* padrão obteve desempenho superior ao *GA - S*.

Figura 99 – Resultados de $\overline{f_b}$ por número de avaliações para o caso $\Delta\theta = 15^\circ$.



Fonte: Elaborada pelo autor.

Figura 100 – Resultados de $\overline{f_b}$ por número de avaliações para o caso $\Delta\theta = 5^\circ$.



Fonte: Elaborada pelo autor.

7 CONCLUSÃO

Este trabalho tratou da análise isogeométrica e otimização de estruturas laminadas. A Análise Isogeométrica foi formulada utilizando Lagrangeana Total, de maneira a permitir a análise de estruturas com grandes deslocamentos. A formulação foi desenvolvida inicialmente para sólidos isotrópicos e depois estendida para o caso de sólidos laminados.

A formulação foi implementada em um programa de elementos finitos acadêmico utilizando o paradigma de Programação Orientada a Objetos. A implementação considerou o procedimento de Extração de Bézier. Este procedimento permite que a estrutura básica utilizada em programas de elementos finitos, como a montagem da matriz de rigidez e vetor de forças externas, possa ser aproveitada na análise isogeométrica. As principais modificações realizadas no programa se concentraram na criação de rotinas para cálculo das funções de forma e suas derivadas.

Apesar do foco deste trabalho ter sido na implementação de elementos sólidos, também foram implementados elementos bidimensionais, como placas e cascas abatidas laminadas. Estes elementos foram facilmente desenvolvidos devido ao uso da Programação Orientada a Objetos utilizada no programa de elementos finitos.

A análise isogeométrica de sólidos laminados requer um alto custo computacional para ser processada. Este alto custo está relacionado com a quadratura utilizada para realizar a integração numérica na espessura do modelo. Uma estratégia de paralelização da montagem da matriz de rigidez global foi implementada, reduzindo de forma significativa o tempo necessário para o processamento dos exemplos tratados neste trabalho.

É importante notar que não foram encontrados trabalhos na literatura utilizando a AIG para sólidos laminados, apesar de existirem alguns trabalhos abordando placas laminadas^{16,96}. Todos os exemplos analisados obtiveram resultados em excelente concordância com a literatura, indicando que uma correta formulação e implementação foram realizadas. Os exemplos de cascas modelados como sólido apresentaram bons resultados mesmo sendo utilizado apenas um elemento quadrático por espessura. Assim, a AIG se mostrou uma ferramenta adequada para a análise linear e não linear de estruturas laminadas.

Um algoritmo híbrido baseado no método da Nuvem de Partículas para otimização de estruturas laminadas foi desenvolvido e aplicado com sucesso. O método combina operadores genéticos e operadores específicos para otimização de estruturas laminadas com a heurística clássica da Nuvem de Partículas. Um procedimento de calibração de parâmetros (meta-otimização) foi utilizado com o intuito de comparar de forma justa as variantes do método de otimização proposto.

É importante ressaltar que não foram encontrados trabalhos abordando a meta-otimização aplicada a problemas de estruturas laminadas. A meta-otimização é um procedimento de alto custo computacional, que depende do uso de computação paralela para ser realizado. Neste trabalho, a meta-otimização foi implementada em um sistema de otimização

acadêmico. Tal sistema já possuía procedimentos de computação paralela implementados, o que facilitou a realização deste trabalho. É importante notar que foi essencial para este trabalho ter a disponibilidade de um cluster para execução das otimizações e meta-otimizações.

Os resultados obtidos nos exemplos de otimização mostraram que as melhores variantes do método proposto são a que usa topologia global e mutação (*PSO - GMS*) e a que usa topologia quadrada (*PSO - SS*). O algoritmo proposto obteve desempenho comparável ao AG em problemas de laminados tradicionais, onde as orientações possíveis são limitadas a poucos ângulos. Por outro lado, o algoritmo proposto apresentou desempenho superior ao do AG na otimização de laminados dispersos. Isto mostra que o método proposto é capaz de otimizar tanto laminados tradicionais quanto laminados dispersos de forma adequada.

Apesar dos métodos de otimização heurísticos utilizados neste trabalho não garantirem que a melhor solução obtida é de fato o mínimo do problema, dois algoritmos distintos, PSO e AG, obtiveram os mesmos projetos ótimos em todos os exemplos abordados neste trabalho. A probabilidade que ambos os algoritmos tenham obtido as mesmas soluções sem que estas sejam ótimas é bastante pequena.

Algumas sugestões de trabalho futuros são apresentadas a seguir:

- a) Implementar a Análise Isogeométrica utilizando T-Splines. Estas representações geométricas são uma generalização de NURBS e podem aplicar refinamentos de forma localizada.
- b) Implementar os algoritmos de refinamentos AIG dentro do programa de análise estrutural.
- c) Estudar o efeito que o grau da NURBS ao longo da direção da espessura do laminado têm sobre o campo de deformações e tensões ao longo da espessura do laminado, considerando que apenas um elemento sólido laminado é utilizado na espessura da estrutura.
- d) Implementar elementos isogeométricos de casca para análise de laminados.
- e) Estudar o efeito que a variação do número de lâminas consideradas na otimização causa no desempenho do método de otimização proposto.
- f) Aplicar a meta-otimização considerando mais exemplos de otimização de laminados para vários números de avaliações diferentes.
- g) Aplicar otimização de forma utilizando análise isogeométrica como ferramenta de análise.

REFERÊNCIAS

- 1 LE RICHE, R.; HAFTKA, R. T. Optimization of laminate stacking sequence for buckling load maximization by genetic algorithm. **AIAA Journal**, v. 31, n. 5, p. 951–956, May 1993.
- 2 LIU, B.; HAFTKA, R. T.; AKGÜN, M. A.; TODOROKI, A. Permutation genetic algorithm for stacking sequence design of composite laminates. **Computer Methods in Applied Mechanics and Engineering**, v. 186, n. 2–4, p. 357 – 372, 2000.
- 3 SOREMEKUN, G.; GÜRDAL, Z.; HAFTKA, R.; WATSON, L. Composite laminate design optimization by genetic algorithm with generalized elitist selection. **Computers & Structures**, v. 79, n. 2, p. 131 – 143, 2001.
- 4 SILVA, R. F.; TEÓFILO, F. A. F.; Parente Junior, E.; MELO, A. M. C.; HOLANDA, Á. S. Optimization of composite catenary risers. **Marine Structures**, v. 33, p. 1 – 20, 2013.
- 5 ROCHA, I. B. C. M.; Parente Junior, E.; MELO, A. M. C. A hybrid shared/distributed memory parallel genetic algorithm for optimization of laminate composites. **Composite Structures**, v. 107, n. 0, p. 288 – 297, 2014.
- 6 AYMERICH, F.; SERRA, M. Optimization of laminate stacking sequence for maximum buckling load using the ant colony optimization (aco) metaheuristic. **Composites Part A: Applied Science and Manufacturing**, v. 39, n. 2, p. 262 – 272, 2008.
- 7 WANG, W.; GUO, S.; CHANG, N.; ZHAO, F.; YANG, W. A modified ant colony algorithm for the stacking sequence optimisation of a rectangular laminate. **Structural and Multidisciplinary Optimization**, v. 41, n. 5, p. 711–720, 2010.
- 8 KOIDE, R. M.; FRANÇA, G. V. Z. D.; LUERSEN, M. A. An ant colony algorithm applied to lay-up optimization of laminated composite plates. **Latin American Journal of Solids and Structures**, v. 10, p. 491 – 504, 05 2013.
- 9 KATHIRAVAN, R.; GANGULI, R. Strength design of composite beam using gradient and particle swarm optimization. **Composite Structures**, v. 81, n. 4, p. 471 – 479, 2007.
- 10 CHO, H. Maximizing structure performances of a sandwich panel with hybrid composite skins using particle swarm optimization algorithm. **Journal of Mechanical Science and Technology**, v. 23, n. 12, p. 3143–3152, 2009.
- 11 GHASHOCHI BARGH, H.; SADR, M. Stacking sequence optimization of composite plates for maximum fundamental frequency using particle swarm optimization algorithm. **Meccanica**, v. 47, n. 3, p. 719–730, 2012.
- 12 LAKSHMI, K.; MOHAN RAO, A. R. Optimal design of laminate composite isogrid with dynamically reconfigurable quantum pso. **Struct. Multidiscip. Optim.**, Secaucus, NJ, USA, v. 48, n. 5, p. 1001–1021, Nov. 2013.
- 13 COOK, R. D.; MALKUS, D. S.; PLESHA, M. E.; WITT, R. J. **Concepts and applications of finite element analysis**. 4nd. ed. John Wiley & Sons, 2002.

- 14 HUGHES, T.; COTTRELL, J.; BAZILEVS, Y. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. **Computer Methods in Applied Mechanics and Engineering**, v. 194, n. 39–41, p. 4135 – 4195, 2005.
- 15 COTTRELL, J. A.; HUGHES, T. J.; BAZILEVS, Y. **Isogeometric analysis: Toward integration of cad and fea**. John Wiley & Sons Ltd, 2009.
- 16 KAPOOR, H.; KAPANIA, R. Geometrically nonlinear NURBS isogeometric finite element analysis of laminated composite plates. **Composite Structures**, v. 94, n. 12, p. 3434 – 3447, 2012.
- 17 LE-MANH, T.; LEE, J. Postbuckling of laminated composite plates using nurbs-based isogeometric analysis. **Composite Structures**, v. 109, n. 0, p. 286 – 293, 2014.
- 18 ROCHA, I. B. C. M. **Análise e otimização de cascas laminadas considerando não linearidade geométrica e falha progressiva**. 2013. Dissertação (Mestrado em Engenharia Civil) – Universidade Federal do Ceará, 2013.
- 19 PEDERSEN, M.; CHIPPERFIELD, A. Simplifying particle swarm optimization. **Applied Soft Computing**, v. 10, n. 2, p. 618 – 628, 2010.
- 20 Dantas Junior, E. M. **Análise não linear de compósitos laminados utilizando método dos elementos finitos**. 2014. Dissertação (Mestrado em Estruturas) – Universidade Federal do Ceará, 2014.
- 21 JONES, R. M. **Mechanics of composite materials**. 2nd. ed. Taylor & Francis, 1999.
- 22 DANIEL, I. M.; ISHAI, O. **Engineering mechanics of composite materials**. 2nd. ed. Oxford University Press, 2006.
- 23 BARBERO, E. J. **Introduction to composite materials design**. 2nd. ed. 2011.
- 24 REDDY, J. N. **Mechanics of laminated composite plates and shells: Theory and analysis**. 2nd. ed. CRC Press, 2004.
- 25 TSAI, S. W.; WU, E. M. A General Theory of Strength for Anisotropic Materials. **Journal of Composite Materials**, v. 5, p. 58–80, 1971.
- 26 CARRERA, E. Theories and finite elements for multilayered, anisotropic, composite plates and shells. **Archives of Computational Methods in Engineering**, v. 9, n. 2, p. 87–140, 2002.
- 27 CARRERA, E. Theories and finite elements for multilayered plates and shells:a unified compact formulation with numerical assessment and benchmarking. **Archives of Computational Methods in Engineering**, v. 10, n. 3, p. 215–296, 2003.
- 28 LI, K.; QIAN, X. Isogeometric analysis and shape optimization via boundary integral. **Computer-Aided Design**, v. 43, n. 11, p. 1427 – 1437, 2011. Solid and Physical Modeling 2011.
- 29 MANTYLA, M. **Introduction to solid modeling**. New York, NY, USA: W. H. Freeman &

- Co., 1988.
- 30 PIEGL, L.; TILLER, W. **The nurbs book (monographs in visual communication)**. 2. ed. Springer-Verlag Berlin Heidelberg, 1997.
 - 31 FARIN, G. **Curves and surfaces for cagd: A practical guide**. 5th. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002.
 - 32 DE CASTELJAU, P. Outillages méthodes calcul. Technical report, A. Citroen, Paris, 1959.
 - 33 DE CASTELJAU, P. Courbes et surfaces á poles. Technical report, A. Citroen, Paris, 1963.
 - 34 HARFORD, A. **GIMP essential reference**. New Riders professional library. Indianapolis, IN: New Riders, 2000.
 - 35 HURTER, B. **The Best of Adobe Photoshop: Techniques and Images from Professional Photographers**. Chicago: Amherst Media Inc., 2006.
 - 36 BAH, T. **Inkscape: Guide to a vector drawing program**. First. ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2007.
 - 37 HARREL, W.; STEWARD, W. **Coreldraw 7 secrets**. 1st. ed. Foster City, CA, USA: IDG Books Worldwide, Inc., 1997.
 - 38 LORENTZ, G. G. **Bernstein polynomials**. second. ed. New York, N.Y.: Chelsea Publishing Company, 1986.
 - 39 SEDERBERG, T. W. Computer aided geometric design course notes, October 2014.
 - 40 BARTELS, R. H.; BEATTY, J. C.; BARSKY, B. A. **An introduction to splines for use in computer graphics & geometric modeling**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1987.
 - 41 Programa Rhinoceros. <https://www.rhino3d.com/>. Acessado em 10/09/2015.
 - 42 Programa Maya. <http://www.autodesk.com/products/maya/overview>. Acessado em 10/09/2015.
 - 43 Programa Autocad. <http://www.autodesk.com/products/autocad/overview>. Acessado em 10/09/2015.
 - 44 PIEGL, L. On nurbs: A survey. **IEEE Comput. Graph. Appl.**, Los Alamitos, CA, USA, v. 11, n. 1, p. 55–71, Jan. 1991.
 - 45 KIENDL, J.; BAZILEVS, Y.; HSU, M.; WÜCHNER, R.; BLETZINGER, K. The bending strip method for isogeometric analysis of kirchhoff–love shell structures comprised of multiple patches. **Computer Methods in Applied Mechanics and Engineering**, v. 199, n. 37–40, p. 2403 – 2416, 2010.
 - 46 BENSON, D.; BAZILEVS, Y.; HSU, M.; HUGHES, T. Isogeometric shell analysis: The reissner–mindlin shell. **Computer Methods in Applied Mechanics and Engineering**, v.

- 199, n. 5–8, p. 276 – 289, 2010. Computational Geometry and Analysis.
- 47 HOSSEINI, S.; REMMERS, J. J.; VERHOOSSEL, C. V.; DE BORST, R. An isogeometric continuum shell element for non-linear analysis. **Computer Methods in Applied Mechanics and Engineering**, v. 271, n. 0, p. 1 – 22, 2014.
- 48 BOUCLIER, R.; ELGUEDJ, T.; COMBESURE, A. An isogeometric locking-free nurbs-based solid-shell element for geometrically nonlinear analysis. **International Journal for Numerical Methods in Engineering**, v. 101, n. 10, p. 774–808, 2015.
- 49 CRISFIELD, M. **Non-linear finite element analysis of solids and structures. volume 1. , essentials**. Chichester, New York: Wiley, 1991.
- 50 BATHE, K.-J. **Finite element procedures**. Englewood Cliffs, N.J. Prentice Hall, 1996. Éd. rev. de: Finite element procedures in engineering analysis. 1982.
- 51 MONTEIRO, F. A. C. **Uma formulação co-rotacional geral: Aplicação a pórticos espaciais**. 2004 – Instituto Tecnológico de Aeronáutica, 2004.
- 52 NGUYEN, T. N. **Isogeometric finite element analysis based on bézier extraction of nurbs and t-splines**. 2006. Dissertação (Mestrado em Computational Mechanics) – Norwegian University of Science and Technology, Trondheim, Noruega, 2006.
- 53 BORDEN, M. J.; SCOTT, M. A.; EVANS, J. A.; HUGHES, T. J. R. Isogeometric finite element data structures based on bézier extraction of nurbs. **International Journal for Numerical Methods in Engineering**, v. 87, n. 1-5, p. 15–47, 2011.
- 54 Parente Junior, E. **Fast version 1.0.0 user's manual**. Universidade Federal do Ceará, March 2015.
- 55 MARTHA, L. F.; Parente Junior, E. An object-oriented framework for finite element programming. In: . Editor H. A. MANG, F. G. RAMMERSTORFER, J. E. c2002.
- 56 NETO, M. M. **Estabilidade de edifícios de concreto pré-moldado com ligações semi-rígidas**. 2012. Dissertação (Mestrado em Estruturas) – Universidade Federal do Ceará, 2012.
- 57 FOWLER, M. **Uml distilled: A brief guide to the standard object modeling language**. 3. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2003.
- 58 DAGUM, L.; MENON, R. Openmp: An industry-standard api for shared-memory programming. **IEEE Comput. Sci. Eng.**, Los Alamitos, CA, USA, v. 5, n. 1, p. 46–55, Jan. 1998.
- 59 HUGHES, T.; REALI, A.; SANGALLI, G. Efficient quadrature for nurbs-based isogeometric analysis. **Computer Methods in Applied Mechanics and Engineering**, v. 199, n. 5–8, p. 301 – 313, 2010. Computational Geometry and Analysis.
- 60 TIMOSHENKO, S.; WOINOWSKY-KRIEGER, S. **Theory of plates and shells (engineering societies monograph)**. 2nd. ed. McGraw-Hill, 1959.

- 61 BATHE, K.-J.; BOLOURCHI, S. Large displacement analysis of three-dimensional beam structures. **International Journal for Numerical Methods in Engineering**, v. 14, n. 7, p. 961–986, 1979.
- 62 MORORÓ, L. A. T. **Análise não linear geométrica de vigas laminadas de parede fina**. 2013 – Universidade Federal do Ceará, 2013.
- 63 SZE, K. Y.; LIU, X. H.; LO, S. H. Popular benchmark problems for geometric nonlinear analysis of shells. **Finite Elem. Anal. Des.**, Amsterdam, The Netherlands, The Netherlands, v. 40, n. 11, p. 1551–1569, July 2004.
- 64 GRAHAM, S. L.; KESSLER, P. B.; MCKUSICK, M. K. Gprof: A Call Graph Execution Profiler. In: . SIGPLAN '82. New York, NY, USA: ACM, c1982. p. 120–126.
- 65 AWAD, Z. K.; ARAVINTHAN, T.; ZHUGE, Y.; GONZALEZ, F. A review of optimization techniques used in the design of fibre composite structures for civil engineering applications. **Materials & Design**, v. 33, n. 0, p. 534 – 544, 2012.
- 66 KOGISO, N.; WATSON, L.; GÜRDAL, Z.; HAFTKA, R. Genetic algorithms with local improvement for composite laminate design. **Structural optimization**, v. 7, n. 4, p. 207–218, 1994.
- 67 NASA Langley Research Center. Advanced composite structures. Technical report, Hampton, VA, USA, 1974.
- 68 GÜRDAL, Z.; HAFTKA, R. T.; HAJELA, P. **Design and optimization of laminated composite materials**. John Wiley & Sons, Ltd, 1999.
- 69 SURESH, S.; SUJIT, P.; RAO, A. Particle swarm optimization approach for multi-objective composite box-beam design. **Composite Structures**, v. 81, n. 4, p. 598 – 605, 2007.
- 70 SCHMIT, L. A.; FARSHI, B. Optimum design of laminated fibre composite plates. **International Journal for Numerical Methods in Engineering**, v. 11, n. 4, p. 623–640, 1977.
- 71 FARKAS, J.; JÁRMAI, K. Minimum material cost design of five-layer sandwich beams. **Structural optimization**, v. 15, n. 3-4, p. 215–220, 1998.
- 72 CARDOSO, J.; SOUSA, L.; CASTRO, J.; VALIDO, A. Optimal design of laminated composite beam structures. **Structural and Multidisciplinary Optimization**, v. 24, n. 3, p. 205–211, 2002.
- 73 CHANG, N.; WANG, W.; YANG, W.; WANG, J. Ply stacking sequence optimization of composite laminate by permutation discrete particle swarm optimization. **Structural and Multidisciplinary Optimization**, v. 41, n. 2, p. 179–187, 2010.
- 74 SEBAEY, T.; LOPES, C.; BLANCO, N.; COSTA, J. Ant colony optimization for dispersed laminated composite panels under biaxial loading. **Composite Structures**, v. 94, n. 1, p. 31 – 36, 2011.
- 75 ARORA, J. S. **Introduction to optimum design**. 3rd. ed. 2012.

- 76 TODOROKI, A.; SASAI, M. Improvement of design reliability for buckling load maximization of composite cylinder using genetic algorithm with recessive-gene-like repair. **JSME international journal. Series A, Solid mechanics and material engineering**, v. 42, n. 4, p. 530–536, oct 1999.
- 77 DEB, K. An efficient constraint handling method for genetic algorithms. **Computer Methods in Applied Mechanics and Engineering**, v. 186, n. 2–4, p. 311 – 338, 2000.
- 78 WOLPERT, D.; MACREADY, W. No free lunch theorems for optimization. **Evolutionary Computation, IEEE Transactions on**, v. 1, n. 1, p. 67–82, Apr 1997.
- 79 SÖRENSEN, K. Metaheuristics—the metaphor exposed. **International Transactions in Operational Research**, p. n/a–n/a, 2013.
- 80 RICHE, R. L.; HAFTKA, R. Improved genetic algorithm for minimum thickness composite laminate design. **Composites Engineering**, v. 5, n. 2, p. 143 – 161, 1995.
- 81 KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: . c1995. v. 4. p. 1942–1948 vol.4.
- 82 KENNEDY, J.; MENDES, R. Population structure and particle swarm performance. In: . c2002. v. 2. p. 1671–1676.
- 83 KENNEDY, J.; MENDES, R. Neighborhood topologies in fully informed and best-of-neighborhood particle swarms. **Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on**, v. 36, n. 4, p. 515–519, July 2006.
- 84 RATNAWEERA, A.; HALGAMUGE, S. K.; WATSON, H. C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. **Trans. Evol. Comp**, Piscataway, NJ, USA, v. 8, n. 3, p. 240–255, June 2004.
- 85 RIGET, J.; VESTERSTRØM, J. A diversity-guided particle swarm optimizer - the arps. Technical report, 2002.
- 86 ARANI, B. O.; MIRZABEYGI, P.; PANAHI, M. S. An improved pso algorithm with a territorial diversity-preserving scheme and enhanced exploration–exploitation balance. **Swarm and Evolutionary Computation**, v. 11, n. 0, p. 1 – 15, 2013.
- 87 BRATTON, D.; KENNEDY, J. Defining a standard for particle swarm optimization. In: . c2007. p. 120–127.
- 88 CLERC, M. Standard Particle Swarm Optimisation. 15 pages, Sept. 2012.
- 89 SHI, Y.; EBERHART, R. C. Parameter selection in particle swarm optimization. In: . EP '98. London, UK, UK: Springer-Verlag, c1998. p. 591–600.
- 90 TRELEA, I. C. The particle swarm optimization algorithm: convergence analysis and parameter selection. **Information Processing Letters**, v. 85, n. 6, p. 317 – 325, 2003.
- 91 GREFENSTETTE, J. Optimization of control parameters for genetic algorithms. **Systems, Man and Cybernetics, IEEE Transactions on**, v. 16, n. 1, p. 122–128, Jan 1986.

- 92 MEISSNER, M.; SCHMUKER, M.; SCHNEIDER, G. Optimized particle swarm optimization (opso) and its application to artificial neural network training. **BMC Bioinformatics**, v. 7, n. 1, p. 125, 2006.
- 93 SMIT, S. K.; EIBEN, A. E. Comparing parameter tuning methods for evolutionary algorithms. In: . CEC'09. Piscataway, NJ, USA: IEEE Press, c2009. p. 399–406.
- 94 BARROSO, E. S.; Parente Junior, E.; MELO, A. M. C. Meta-otimização aplicada ao projeto de estruturas laminadas. In: . Editor Parente Junior, E. c2014.
- 95 LOPEZ, R.; LUERSEN, M.; CURSI, E. Optimization of laminated composites considering different failure criteria. **Composites Part B: Engineering**, v. 40, n. 8, p. 731 – 740, 2009.
- 96 THAI, C. H.; NGUYEN-XUAN, H.; BORDAS, S. P. A.; NGUYEN-THANH, N.; RABEZUK, T. Isogeometric analysis of laminated composite plates using the higher-order shear deformation theory. **Mechanics of Advanced Materials and Structures**, v. 22, n. 6, p. 451–469, 2015.