LEONARDO RICCIOPPO GARCEZ

**Investigação de Métodos Geradores de Malhas Aplicados a
Geometrias Típicas das Seções Transversais de Cabos Umbilicais e
Tubos Flexíveis**

São Paulo

2017

LEONARDO RICCIOPPO GARCEZ

**Investigação de Métodos Geradores de Malhas Aplicados a
Geometrias Típicas das Seções Transversais de Cabos Umbilicais e
Tubos Flexíveis**

**Dissertação apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção do título
de Mestre em Ciências**

**Área de Concentração:
Engenharia Mecânica - Controle e
Automação**

**Orientador: Prof. Dr. Clóvis de Arruda
Martins**

**São Paulo
2017**

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, _____ de _____ de _____

Assinatura do autor: _____

Assinatura do orientador: _____

Catálogo-na-publicação

Garcez, Leonardo Riccioppo

Investigação de Métodos Geradores de Malhas Aplicados a Geometrias Típicas das Seções Transversais de Cabos Umbilicais e Tubos Flexíveis / L. R. Garcez -- versão corr. -- São Paulo, 2017.
172 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecânica.

1.Computer Aided Design 2.Computer Aided Engineering 3.Método dos Elementos Finitos 4.Cabos Umbilicais 5.Tubos Flexíveis I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecânica II.t.

À minha família por todo apoio dado ao longo do período, ao meu orientador por toda sua experiência e sabedoria passadas em seus conselhos, à CAPES por todo apoio financeiro e à Universidade de São Paulo pelo acolhimento deste projeto em seu Programa de Pós-Graduação em Engenharia Mecânica da Escola Politécnica.

Digo: o real não está na saída nem na chegada: ele se dispõe para a gente é no meio da travessia.

(João Guimarães Rosa)

RESUMO

Os métodos de geração de malhas são essenciais para simulações numéricas de engenharia. Simulações estruturais pelo método dos elementos finitos (MEF) dependem diretamente das malhas que representam as geometrias às quais são impostos os carregamentos e parte importante da precisão dos cálculos é devida à qualidade da malha. Há uma vasta variedade de métodos e algoritmos para gerar malhas computacionais direcionadas à aplicação do MEF, mas nem todos abrangem geometrias genéricas ou irregulares. Figuras irregulares com regiões curvas e vazios internos requerem o uso de algoritmos específicos, ou genéricos o suficiente, de modo que sejam capazes de processar suas particularidades geométricas, gerando elementos de boa qualidade. Este trabalho propõe uma investigação dos métodos geradores de malhas que tratem as geometrias frequentemente apresentadas nas seções transversais dos componentes dos cabos umbilicais e tubos flexíveis, selecionando criteriosamente aquele que mais se adapte às necessidades geradas pela aplicação do MEF nestas geometrias. Foi realizada uma breve revisão do estado da arte dos métodos geradores de malhas bidimensionais partindo das revisões já publicadas e de artigos explicando versões de algoritmos, seguindo os trabalhos encontrados na literatura. São formulados os critérios de seleção de métodos geradores de malha e de trabalhos encontrados baseando-se nas características. Ao final, o Método de Pavimentação foi selecionado e, a partir de sua implementação computacional, foi desenvolvido o GreenMesh, um programa de computador gerador de malhas utilizando figuras dadas como entrada. Foram geradas várias malhas a partir das geometrias típicas dos equipamentos aqui focados e com isso foi verificado o uso do algoritmo para este fim.

Palavras-chave: Computer Aided Design. Computer Aided Engineering. Método dos Elementos Finitos. Cabos Umbilicais. Tubos Flexíveis.

ABSTRACT

Computational mesh generation methods are essential for engineering numerical simulation. Structural simulations by the finite element method (FEM) rely straightly of meshes that represent the geometries on which the loads are imposed. They are very important for the calculus precision because the mesh quality influences the results. There is a vast variety of computational mesh generation methods and algorithms aimed to FEM analysis, but not all handles generic and irregular geometries. Irregular figures formed by curved regions and internal void requires the use of specific algorithms, or sufficiently generic, in a way that they can process geometric particularities, generating good quality elements. This work proposes an investigation of mesh generation methods that handle frequently presented geometries in cross sections of umbilical cable and flexible pipes internal components, selecting judiciously the method that best suits the necessities generated by the application of FEM analysis on theses geometries. A brief survey about the state of the art of bidimensional mesh generation methods was performed, starting from already published reviews and from articles explaining versions of algorithms, accordingly to the material found in the literature. Finally, the Paving Method has been selected and, starting from its computational implementation, it has been developed GreenMesh, a computer program to generate meshes using given figures as input. It has been generated many meshes from typical geometries of the aimed equipments and that verified the use of the algorithm to this purpose.

Keywords: Computer Aided Design. Computer Aided Engineering. Finite Element Method. Umbilical cables. Flexible pipes.

LISTA DE FIGURAS

Figura 1.1 - a) Cabo umbilical conectado a uma FPS e aos equipamentos submersos.....	12
Figura 1.2 - a) Exemplo de umbilical de controle de injeção química. b) Exemplo de tubo flexível....	14
Figura 1.3 - a) Seção da carcaça intertravada. b) Modelo tridimensional em corte da carcaça intertravada.	16
Figura 1.4 - Perfis da armadura de pressão.	17
Figura 1.5 - a) Um modelo de um cabo umbilical de controle hidráulico. b) Um cabo umbilical de potência da Nexans™. c) Cabo submarino de alta voltagem da empresa Akker™.	18
Figura 1.6 - Tela de uma das primeiras versões do CableCAD.	19
Figura 1.7 - Exemplo de tela do programa desenvolvido.....	20
Figura 1.8 - Exemplo de malha dos componentes do cabo umbilical e resultado de simulação.	21
Figura 1.9 - Exemplo de malha hexaédrica.	23
Figura 1.10 - Exemplo de malha 2D de triângulos.	24
Figura 1.11 - Modelo de malha 2D de quadriláteros.	24
Figura 5.1 - Exemplo de decomposição de topologia.	34
Figura 5.2 - Inconsistência na triangulação.	34
Figura 5.3 - a) Figura com os nós gerados. b) Figura após a conexão dos nós	35
Figura 5.4 - Exemplo de malha adaptada.....	37
Figura 5.5 - a) Figura plana dividida em 2 macroelementos; b) Projeção da grade de um quadrado unitário para um dos macroelementos; c) Forma final da malha após o método ser aplicado às 2 divisões da figura geométrica.	38
Figura 5.6 - Exemplo esquemático do mapeamento conformado.	39
Figura 5.7 - Exemplo da abordagem de decomposição geométrica.....	40
Figura 5.8 - Modelo de uma geometria discretizada pelo método <i>Quad-Tree</i>	41
Figura 5.9 - Diagrama de Voronoi.	43
Figura 5.10 - Diagrama de Voronoi degenerado.....	43
Figura 5.11 - Diagrama de Delaunay.	44
Figura 5.12 - Sequência de triangulação por frente de avanço.	47
Figura 5.13 - a) Normais das arestas da fronteira. b) Normais dos nós da fronteira. c) Normais dos nós após a suavização de Laplace.....	49
Figura 5.14 - a) Malha refinada com apenas quadriláteros. b) Malha grosseira do método "quadtree".	50
Figura 5.15 - Malha de quadriláteros da seção de uma chave inglesa.	51
Figura 5.16 - Exemplo de conexão de fronteiras interna e externa.....	52
Figura 5.17 - a) Geometria com os eixos médios. b) Remoção das concavidades. c) Divisão das sub-regiões. d) Geração das malhas por sub-região.....	53
Figura 5.18 - a) Geometria dividida em sub-regiões e com as bordas segmentadas. b) Malha de quadriláteros criada a partir das sub-regiões.	54
Figura 5.19 - Modelo do processo de preenchimento das frentes de avanço por elementos.	55
Figura 5.20 - Serie de a até i de estados da malha durante a geração dos quadriláteros.	56
Figura 5.21 - a) Duas frentes de avanço se intersectando devido a elementos muito grandes. b) Duas frentes de avanço opostas e com elementos de tamanhos muito diferentes.....	59
Figura 5.22 - Exemplo de malha de quadriláteros gerada pelo método de centralização.	60

Figura 6.1 - Exemplo trabalhado na artigo de Choi.....	68
Figura 6.2 - Exemplo de malha bastante genérica.	70
Figura 6.3 - Caso com pouca complexidade para geração de malha.	71
Figura 7.1 - Esquema da primeira linha de novos elementos gera.	75
Figura 7.2 - Pseudocódigo do algoritmo de Pavimentação.....	77
Figura 7.3 - Pseudocódigo representando a rotina modificada.....	79
Figura 7.4 - Pseudo-código da versão final do algoritmo de pavimentação desenvolvido.	80
Figura 7.5 - Diagrama de Classes representando as relações entre as principais entidades do programa.....	81
Figura 7.6 - Esquema de trecho de malha com as meia-arestas.....	83
Figura 7.7 - a) Malha de uma coroa com singularidade. b) Malha estruturada de uma coroa	84
Figura 7.8 - Exemplo de singularidade devido ao tamanho de elemento.	85
Figura 7.9 - Esquema de intersecção de segundo nível.	86
Figura 7.10 - Aproximação de elementos de tamanhos diferentes.	87
Figura 7.11 - Esquema representando os diferentes estados dos nós.	88
Figura 7.12 - Faixas de ângulos para cada estado dos nós.....	89
Figura 7.13 - Modelo da projeção do nó tipo <i>side node</i>	90
Figura 7.14 - Modelo de projeção do nó tipo Corner Node.....	91
Figura 7.15 - Modelo de projeção do nó tipo Reversal Node.	92
Figura 7.16 - Exemplo de intersecção entre elementos vizinhos.....	93
Figura 7.17 - Tratamento de arestas com tamanhos diferentes.....	94
Figura 7.18 - Algoritmo de tratamento de intersecções.	95
Figura 7.19 - Modelo esquemático das etapas de tratamento da intersecção elemento-elemento... ..	96
Figura 7.20 - Modelo esquemático das etapas de tratamento da intersecção elemento-fronteira. ..	98
Figura 7.21 - Exemplo de intersecção de elemento de canto.....	99
Figura 7.22 - Exemplo de intersecção de elemento único.	100
Figura 7.23 - Pseudocódigo geral do algoritmo do módulo de costura (<i>Seam</i>).....	101
Figura 7.24 - Configurações da aplicação da costura (<i>Seam</i>) na ocorrência elemento-elemento	102
Figura 7.25 - Configurações da aplicação da costura (<i>Seam</i>) na ocorrência elemento-fronteira.	102
Figura 7.26 - Exemplo de aplicação de dobra. Processo esquematizado da esquerda para a direita.	103
Figura 7.27 – Exemplo de aplicação de dobras.....	104
Figura 7.28 - Esquema de identificação de arco formado por arestas qualificadas.	106
Figura 7.29 - Pseudocódigo do algoritmo de determinação da inserção de dobras.	107
Figura 7.30 - Esquema do processo de inserção de dobras.....	108
Figura 7.31 - Esquema mostrando a inserção de uma cunha na linha de elementos.	109
Figura 7.32 - Exemplo de cunhas dispostas a cada 90º.	110
Figura 7.33 - Esquema de determinação do ângulo para inserção das cunhas.	111
Figura 7.34 - Pseudocódigo para a inserção de cunhas.	112
Figura 7.35 - Esquema mostrando a inserção da cunha.	113
Figura 7.36 - Esquema indicando os parâmetros para o modelo isoparamétrico.	114
Figura 7.37 - Esquema do ajuste de comprimento na suavização.....	115
Figura 7.38 - Esquema do procedimento de ajuste angular da suavização.....	116
Figura 7.39 - Esquema mostrando os nós para o cálculo da suavização interna.....	117
Figura 7.40 - Módulo esquematizado de fechamento de fronteira.....	118
Figura 7.41 - Esquema para representação do colapso em fechamento de fronteira.	119

Figura 7.42 - Esquema da adição da faixa de elementos em uma fronteira com 5 nós.	125
Figura 7.43 - Ângulos internos do elemento.	127
Figura 8.1 - Pipeline da biblioteca OpenGL.	129
Figura 9.1 - Tela inicial do GreenMesh.	131
Figura 9.2 - Visualização do arquivo de geometrias importado no GreenMesh.	132
Figura 9.3 - Sequência de criação de elementos no retângulo.	134
Figura 9.4 - Malha estruturada de um retângulo.	135
Figura 9.5 - Malha não estruturada de um retângulo.	136
Figura 9.6 - Coroa pouco discretizada.	137
Figura 9.7 - Coroa mais discretizada.	137
Figura 9.8 - Malha de um círculo pouco refinada.	138
Figura 9.9 - Malha de círculo mais refinada.	138
Figura 9.10 - Malha do <i>filler</i> com aresta base de 20 mm.	139
Figura 9.11 - Malha do <i>filler</i> com aresta base de 24 mm.	139
Figura 9.12 - Malha do <i>filler</i> com aresta base de 30 mm.	140
Figura 9.13 - Malha do <i>filler</i> com aresta base de 36 mm.	140
Figura 9.14 - Malha do <i>filler</i> com aresta base de 40 mm.	141
Figura 9.15 - Gráfico com a qualidade média das malhas do <i>filler</i>	142
Figura 9.16 - Malha de uma placa com furo.	143
Figura 9.17 - Exemplo de <i>fillers</i> importados no GreenMesh.	144
Figura 9.18 - Malha de um <i>filler</i> genérico inspirado nos exemplos de umbilicais do catálogo da empresa Akker.	145
Figura 13.1 - Sistema Sketchpad de Ivan Sutherland.	157
Figura 13.2 - Elementos de nível hierárquico 0, 1 e 2.	158
Figura 13.3 - Elementos pertencentes ao <i>Loop</i> . Níveis de hierarquia de 2, 3, 4.	159
Figura 13.4 - Elementos contidos na entidade <i>Arc</i>	159
Figura 13.5 - Elementos contidos na entidade <i>Line</i>	160
Figura 13.6 - Exemplo de trecho do arquivo de entrada do GreenMesh.	161
Figura 13.7 - Exemplo de curva de Bézier.	164
Figura 13.8 - Modelo de peça representada por <i>wireframe</i>	166
Figura 13.9 - Representação de um modelo CSG com sua árvore de operações booleanas.	167
Figura 13.10 - Diagrama dos conjuntos de entidades da estrutura Brep.	168
Figura 13.11 - a) Exemplo de malha estruturada; b) Exemplo da malha irregular; c) Exemplo de malha semi-estruturada.	170

LISTA DE TABELAS

Tabela 6.1 - Análise preliminar dos métodos geradores de malhas encontrados na literatura (continua).	63
Tabela 6.2 - Lista de métodos pré-selecionados.	65
Tabela 6.3 - Avaliação final dos métodos pré-selecionados.	73
Tabela 7.1 - Tabela dos padrões para preenchimento da fronteira com 6 nós.	121
Tabela 7.2 - Ângulos internos dos padrões de fronteira no formato retangular	122
Tabela 7.3 - Ângulos internos dos padrões de fronteira no formato triangular	123
Tabela 7.4 - Ângulos internos dos padrões de fronteira no formato semi-circular	124
Tabela 7.5 - Ângulos internos dos padrões de fronteira no formato circular	124

SUMÁRIO

RESUMO	6
ABSTRACT	7
LISTA DE FIGURAS	8
LISTA DE TABELAS	11
SUMÁRIO	12
1 Introdução	12
1.1 Contexto da Indústria Offshore	12
1.2 Modelos de Cabos Umbilicais e Tubos Flexíveis	13
1.3 Programas Específicos de CAD/CAE	18
1.2. Aplicação do Método dos Elementos Finitos	22
1.2.1. Análise Estrutural	22
1.2.2. Precisão do Método	22
1.3. Motivação	25
2 Objetivos	27
3 Metodologia	28
3.1 Fluxo de Execução do trabalho	28
4 Definição dos Critérios Primários de Avaliação	30
5 Revisão Bibliográfica	33
6 Análise Detalhada dos Métodos	62
6.1 Pré-Seleção de Métodos	62
6.2 Definição dos Novos Critérios	65
6.3 Discussão das Versões dos Métodos	67
6.4 Seleção do Método	73
7 Método da Pavimentação	75
7.1 Estrutura de Dados	80
7.2 Discretização das Fronteiras	83
7.3 Etapas do Método	87
7.3.1 Adição de nova linha de elementos	87
7.3.2 Verificação de intersecções (<i>Intersection Module</i>)	92
7.3.3 Verificação de costura de elementos próximos (<i>Seam Module</i>)	100
7.3.4 Inserção de Dobras (<i>Tuck Module – Row Adjustment</i>)	103
7.3.5 Inserção de Cunhas (<i>Wedge Module – Row Adjustment</i>)	108

7.3.6	Suavização (<i>Smooth Module</i>)	113
7.3.7	Fechamento da fronteira (<i>Closing Boundary Module</i>).....	118
7.3.8	Medição da Qualidade de Elemento	126
8	Tecnologias Empregadas.....	128
8.1	Linguagem de Programação.....	128
8.2	Ferramenta de Exibição Gráfica	128
8.3	Interface de Usuário.....	130
9	Resultados da Implementação	131
10	Conclusões.....	147
11	Comentários Finais e Trabalhos Futuros.....	150
12	Bibliografia	152
13	Apêndice.....	156
A.	História da Geometria Computacional.....	156
B.	Formato do Arquivo de Geometria de Entrada do GreenMesh	158
C.	Formas Geométricas e Modelos Representativos	161
D.	Tipos de Malhas (classificação dos elementos e classificação das malhas).....	168

1 Introdução

1.1 Contexto da Indústria Offshore

Tubos flexíveis e cabos umbilicais são equipamentos de vital importância na Indústria Offshore. É através deles que várias atividades deste negócio são feitas como o transporte de petróleo e gás natural de poços de petróleo submersos no mar, bem como a transmissão de potências elétrica e hidráulica para alimentação e controle de maquinário. Eles são empregados conectando as unidades produtoras de petróleo (*Floating Production System*), ou FPS, aos poços e aparelhos submersos, o que pode exigir alguns quilômetros de tubo ou cabo. Desta forma, estas estruturas ficam expostas a ambientes extremos e precisam resistir a altas tensões provenientes dos carregamentos estáticos e dinâmicos devidos ao próprio peso, às variações de correnteza marítima, às variações de temperaturas, ao fluxo de seus fluidos internos, entre a outros fenômenos. A Figura 1.1 mostra exemplos esquemáticos das unidades de processamento e produção de petróleo e gás conectados por estes dispositivos.

Figura 1.1 - a) Cabo umbilical conectado a uma FPS e aos equipamentos submersos.

Fonte: (AKKER solutions - power cable catalog, 2015)

O projeto destes equipamentos não é uma tarefa trivial. No caso dos cabos umbilicais, cada aplicação exige um projeto diferente, o que torna difícil transformá-los em produtos

catalogados. Assim, eles são dimensionados para as situações específicas de cada aplicação, com relação ao modo de fabricação, às propriedades dos materiais e aos carregamentos a que serão submetidos. As análises da influência das cargas nestes equipamentos têm grande contribuição das simulações pelo método dos elementos finitos (MEF), geralmente adotando ferramentas específicas para esta finalidade. Programas como ANSYS™, NASTRAN™ e ABAQUS™ são globalmente conhecidos e utilizados em simulações mecânicas, no entanto são muito genéricos e a grande quantidade de ferramentas que apresentam, pode os tornar sistemas pouco adequados para o dia-a-dia de projeto de cabos umbilicais e tubos flexíveis. Este fato cria espaço na Indústria para o desenvolvimento de ferramentas computacionais específicas e otimizadas para estes produtos. Assim, se faz necessário o conhecimento aprofundado dos componentes internos típicos destes dispositivos, para que se possibilite a modelagem representativa bidimensional de suas seções transversais de maneira adequada.

1.2 Modelos de Cabos Umbilicais e Tubos Flexíveis

Estes dispositivos são compostos por várias subestruturas que formam os componentes internos e camadas, conforme pode ser observado na Figura 1.2. Existe uma série de funcionalidades próprias de cada tipo de componente como resistência a esforços, isolamento, transporte de fluidos e transmissão de sinal e potência. Na Figura 1.2, observa-se um exemplo de um cabo umbilical e de um tubo flexível.

Figura 1.2 - a) Exemplo de umbilical de controle de injeção química. b) Exemplo de tubo flexível.

Fonte: a) (PRYSMIAN deep water umbilicals catalog, 2015). b) (TECHNIP flexible pipe catalog, 2015)

Os componentes e camadas típicos destes equipamentos podem ser generalizados nos seguintes tipos:

Carcaça Intertravada: é a camada mais interna de tubos flexíveis e de alguns tubos hidráulicos de cabos umbilicais. É formada por um perfil *extrudado* em hélice que, por seu formato típico, garante um autotravamento. Tem a função de resistir a carregamentos externos de pressão e esmagamento.

Capa Plástica: é uma camada composta por um polímero específico que pode ser utilizada na estanqueidade no caso de camadas mais internas, ou no isolamento dos componentes internos do meio externo, no caso de ser uma camada mais externa.

Armadura de Pressão: armadura com perfil característico, utilizada na sustentação de esforços resultantes da pressão interna da tubulação, da pressão externa e de esmagamento (*crushing*).

Armadura Helicoidal: é uma malha helicoidal formada por tendões de seção próxima à circular ou à retangular. Utilizadas frequentemente em pares com ângulos em sentidos opostos, atua na resistência a carregamentos axiais e, algumas vezes, radiais.

Capa Antidesgaste: é uma capa utilizada entre camadas metálicas, como a armadura helicoidal, para reduzir o atrito entre elas.

Fillers: são componentes geralmente de material polimérico, com função de preencher vazios nos espaços próximos dos componentes. Podem apresentar geometrias irregulares nas seções transversais.

Cabos Elétricos e Ópticos: são cabos de potência e de dados utilizados no envio de potência elétrica e informações digitais ao maquinário submerso, respectivamente.

Mangueira: estruturas de composição semelhante aos tubos flexíveis utilizadas para a injeção de fluidos e de potência hidráulica aos equipamentos submersos.

Fitas de Reforço: são fitas enroladas em torno de alguma camada ou componente para aumentar resistência a esforços radiais. Geralmente é usado o *kevlar* como material de reforço.

Pode-se notar que a seção transversal de alguns componentes segue uma forma cilíndrica, mas também apresentam configurações helicoidais no espaço tridimensional. A carcaça intertravada é um exemplo, pois é formada por uma seção em “S”, como mostra a Figura 1.3a, e é projetada no espaço 3D em hélice de modo que tal perfil se auto-entrelaça, formando um tubo cilíndrico. A Figura 1.3b expõe um modelo computacional de uma fatia referente ao corte axial deste componente.

Figura 1.3 - a) Seção da carcaça intertravada. b) Modelo tridimensional em corte da carcaça intertravada.

Fonte: a) (API_RP_17B - recommended practice for flexible pipe, 2002). b) (Provasi, Meirelles, & Martins, 2010).

Outra geometria complexa no projeto de cabos umbilicais e tubos flexíveis é a armadura de pressão, que segue o mesmo princípio de extrusão da carcaça intertravada, mas com a seção transversal de geometria diferente, devido a sua atuação na resistência à pressão interna. A seção pode ser composta de alguns tipos de perfis, conforme mostra a Figura 1.4.

Figura 1.4 - Perfis da armadura de pressão.

Fonte: (API_RP_17B - recommended practice for flexible pipe, 2002).

Também se tem na armadura helicoidal certa dificuldade de modelagem por seus arames em hélices com seções próximas das circulares ou retangulares, não apresentando uma seção exatamente regular em alguns modelos destes equipamentos. Seguindo com outros exemplos de geometrias complexas destes equipamentos, a Figura 1.5 exhibe outros modelos de cabos umbilicais que possuem *fillers* de geometria variada, sugerindo a grande multiplicidade de configurações que são possíveis, principalmente, nos cabos umbilicais.

Figura 1.5 - a) Um modelo de um cabo umbilical de controle hidráulico. b) Um cabo umbilical de potência da Nexans™. c) Cabo submarino de alta voltagem da empresa Akker™.

Fonte: a) (Norway Patente N° 5,902,958, 1999). b) (Nexans Official Website, 2015). c) (AKKER solutions - power cable catalog, 2015).

Tanto a carcaça intertravada quanto a armadura de pressão, por mais complexas que sejam, juntamente com os outros componentes destes equipamentos, podem ser representados em um modelo tridimensional a partir da *extrusão* de suas seções transversais ao longo de uma curva em hélice no espaço. Assim, caso se deseje construir modelos tridimensionais de cabos umbilicais e tubos flexíveis, a obtenção da malha bidimensional de suas seções transversais se torna a tarefa mais complexa do processo.

1.3 Programas Específicos de CAD/CAE

São conhecidas algumas ferramentas e metodologias para o projeto e para a simulação estrutural específica de cabos umbilicais e tubos flexíveis, a fim de se aprimorar a etapa de projeto. Knapp, Le, e Cruickshank (1991) propõem uma metodologia implementada em um programa computacional para auxiliar estes projetos, mas que apresenta limitações de geometrias possíveis para os componentes, permitindo apenas o design da seção transversal do cabo ou tubo a partir de círculos, arcos e coroas. Entretanto, é uma ferramenta que executa simulações com o método dos elementos finitos (MEF) para o cálculo de tensões, deformações e contatos entre os elementos internos destas estruturas. Nas últimas décadas, o programa apresentado neste artigo tem sido constantemente refinado e comercializado com o nome de CableCAD. A Figura 1.6 mostra uma tela das primeiras versões desta ferramenta.

Figura 1.6 - Tela de uma das primeiras versões do CableCAD.

Fonte: (Knapp, Le, & Cruickshank, 1991).

Provasi, Meirelles, e Martins (2010) desenvolveram uma proposta de ferramenta de design para projeto utilizando simulações de MEF com elementos de revolução, que otimizam a modelagem dos componentes típicos destes dispositivos. Entretanto, como ocorre também no programa de Knapp, Le, e Cruickshank (1991), não foi desenvolvido um gerador genérico de malhas, o tornando dependente das suas malhas simplificadas e o limitando a apenas geometrias regulares como arcos, círculos, coroas e retângulos. A Figura 1.7 mostra uma tela do protótipo da ferramenta.

Figura 1.7 - Exemplo de tela do programa desenvolvido.

Fonte: (Provasi, Meirelles, & Martins, 2010).

Outra ferramenta comercial que possibilita do projeto e simulação estrutural de cabos e tubos é o UmbiliCAD®. Desenvolvido pela empresa Ultra Deep™, ele permite a criação de várias formas de componentes internos, no entanto é limitado para geometrias pré-definidas. O usuário tem a possibilidade de criar e editar apenas componentes de um conjunto disponibilizado pelo programa como tubos, cabos elétricos, armaduras e outros. Nele não se disponibiliza a edição livre de formas geométricas para, por exemplo, customizar as figuras de tendões de armaduras ou de enchimentos, como seria possível em um módulo de desenho em programas de CAD (BAKKE, 2009). Há outra ferramenta largamente utilizada na indústria offshore, que possibilita o cálculo de carregamentos nas estruturas de cabos umbilicais e tubos flexíveis. O programa é chamado UFlex e foi desenvolvido pela empresa Marintek. Mesmo sendo muito utilizado, não possui interface de usuário “amigável” e para ser executado, é necessária a criação de arquivos de texto de entrada com todas as informações para a simulação. É necessário fornecer a geometria descrita por meio de um código próprio, descrito em um arquivo de texto, o que torna a tarefa laboriosa. A ferramenta possui um módulo semiautomático de geração de malhas, que trata todas as formas geométricas, dentro das limitações de entrada, criando as malhas rapidamente. Ocorre, entretanto, a mistura de elementos triangulares e

quadriláteros em alguns casos, o que pode afetar a precisão do cálculo. A Figura 1.8 mostra um exemplo de simulação estrutural com as malhas dos componentes internos de um cabo umbilical.

Figura 1.8 - Exemplo de malha dos componentes do cabo umbilical e resultado de simulação.

Fonte: (Marintek, 2010).

Desenvolver um algoritmo gerador automático de malhas para a aplicação do método dos elementos finitos não é uma tarefa trivial. É necessário controlar a qualidade das malhas para garantir um bom desempenho da simulação. A fim de se garantir melhor controle dos resultados das simulações, é necessário que o programa disponibilize certo controle da malha criada, permitindo alterações da subdivisão do domínio conforme a necessidade do caso a ser calculado, como, por exemplo, regiões de concentração de tensão que exigem maior refinamento para melhorar a qualidade dos resultados. Vários métodos geradores de malhas são encontrados na literatura, mas apenas uma parte deste conjunto pode ser aplicada a geometrias genéricas, com fronteiras curvas e espaços internos vazios. Neste trabalho serão analisados alguns métodos encontrados na literatura, comparando-os, a fim de se selecionar o mais indicado para uma implementação específica, conforme será explicado adiante.

1.2. Aplicação do Método dos Elementos Finitos

1.2.1. Análise Estrutural

O Método dos Elementos Finitos (MEF) é uma ferramenta de análise numérica para simular fenômenos físicos representados por equações diferenciais parciais. Seu uso é fundamental em todas as comunidades de engenharia e sua aplicação abrange áreas variadas da ciência como medicina, geologia, análise de estruturas, análise de fluidos, simulação de fenômenos térmicos, químicos, eletromagnéticos, entre outros. As simulações executadas com o auxílio do método dos elementos finitos são implementadas em um computador quando as geometrias se tornam mais complexas, como ocorre na maioria dos casos. Considerando-se uma explicação sucinta, este método se baseia na discretização de um domínio gerando uma malha de elementos (geometria de uma peça ou de uma região de um escoamento de fluido, por exemplo) e na aproximação da solução das equações diferenciais parciais do fenômeno físico para cada divisão do domínio, obtendo, por fim, a solução aproximada global da malha.

1.2.2. Precisão do Método

As soluções numéricas do Método dos Elementos Finitos (MEF) obtidas são aproximações da solução exata, a qual, na maioria dos casos, demanda muito trabalho para ser obtida ou é de obtenção inviável por métodos analíticos ou por verificações experimentais. Assim, devido ao MEF ser uma tecnologia de menor custo, houve, nas últimas décadas, um investimento crescente no desenvolvimento de ferramentas do tipo tanto do meio acadêmico quanto na Indústria. Programas que aplicam o MEF também apresentam, no entanto, problemas críticos de funcionamento como a disponibilidade de memória das máquinas, a velocidade de processamento de malhas muito complexas e os erros numéricos. Uma das etapas cruciais na aplicação do MEF é a geração de uma malha a partir do domínio do problema. Seu refinamento é de grande influência na precisão dos cálculos.

Ao se mirar na aplicação do MEF em estruturas de cabos umbilicais e de tubos flexíveis, depara-se com uma etapa crucial do processo: a geração das malhas. Análises destas estruturas podem abranger modelos geométricos dos mais variados dependendo do tipo de análise que se deseja realizar: modelos de barras unidimensionais representando o dispositivo como um todo de maneira simplificada; representação de geometrias locais através da discretização da seção transversal do equipamento (análise de seus componentes internos, como o exemplo da Figura

1.8); ou até mesmo a necessidade de se modelar um trecho 3D de alguns metros do dispositivo, discretizando todos os seus componentes no espaço tridimensional e isso exige muita memória e processamento computacional da máquina utilizada. Geralmente as malhas bi e tridimensionais são formadas por elementos isoparamétricos e por elementos de casca, representados geometricamente por hexaedros e tetraedros no caso 3D e por quadriláteros e triângulos no caso 2D e de superfícies, conforme mostram os exemplos de malhas 2D geradas pelo programa UFlex na Figura 1.8. Os exemplos Figura 1.9, Figura 1.10, Figura 1.11, mostram outros casos de malha 3D de hexaedros, de uma malha 2D de triângulos e de uma malha 2D de quadriláteros respectivamente.

Figura 1.9 - Exemplo de malha hexaedrica.

Fonte: (Navarro, 2009).

Figura 1.10 - Exemplo de malha 2D de triângulos.

Fonte: (Cavendish, 1974).

Figura 1.11 - Modelo de malha 2D de quadriláteros.

Fonte: (Blacker & Stephenson, 1991).

A malha 3D da Figura 1.9 foi criada com base na malha 2D gerada nas superfícies bidimensionais do modelo, possibilitando a sua construção por um processo de extrusão. Malhas 3D de cabos umbilicais e de tubos flexíveis geralmente são construídas por processo análogo. Nota-se a diferença entre as malhas de algumas superfícies da Figura 1.9, com

diferenças acentuadas nos formatos dos quadriláteros. Na Figura 1.11, também se observam grandes diferenças em seus quadriláteros e o mesmo pode se dizer dos triângulos na Figura 1.10. Esta diferença de distorção entre elementos implica na qualidade da malha. Classifica-se a malha em três tipos quanto à sua homogeneidade e à qualidade de seus elementos: Estruturada, Semiestruturada e Não-estruturada (definidos do Apêndice D). A qualidade da malha influencia os cálculos do MEF, de modo que melhores resultados são obtidos com malhas de melhor qualidade, ou seja, mais próximas de malhas estruturadas. Torna-se necessário, portanto, mensurar a qualidade dos elementos gerados através de algum procedimento de comparação, sendo alguns deles discutidos no Apêndice D.

Na geração de malhas faz-se necessário, portanto, dividir a geometria da estrutura em elementos menores, discretizando adequadamente as curvas e linhas das figuras. O auxílio computacional se torna essencial para o armazenamento, transmissão, processamento e visualização destas informações, exigindo o emprego de modelos de armazenamento de dados geométricos. Destes, os mais frequentemente incorporados em ferramentas computacionais são o *wireframe*, o CSG (*Constructive Solid Geometry*) e o BRep (*Boundary Representation*), explicados no Apêndice C no final deste texto.

Os modelos de representação de geometria computacional foram desenvolvidos com base na ciência da Geometria Computacional, ou CAGD (*Computer Aided Geometry Design*), que foi criada devido à necessidade de se automatizar o projeto de máquinas e dispositivos após o advento dos computadores nas indústrias. Assim, para se representar malhas para a aplicação do Método dos Elementos Finitos em estruturas reais, é essencial que se manipulem as formas geométricas dos equipamentos e dispositivos projetados, fazendo uso das ferramentas disponibilizadas pela CAGD.

1.3. Motivação

As figuras das seções transversais que podem ser inseridas no projeto de um tubo flexível ou cabo umbilical trazem geometrias genéricas além das coroas, círculos e dos retângulos de alguns componentes típicos, como visto nos exemplos da Figura 1.5. Muitas vezes os enchimentos (*fillers*) assumem geometrias complexas e, na criação de suas malhas, se faz necessário utilizar um método de geração de malhas que atenda tanto as geometrias simples quanto às mais genéricas. Com o intuito de se desenvolver, no futuro, um programa CAE específico para a implementação do MEF (Método dos Elementos Finitos) na simulação

estrutural de cabos umbilicais e tubos flexíveis, a criação de um algoritmo gerador automático de malhas se torna uma tarefa essencial.

Este trabalho discute uma solução para questão da criação de um módulo gerador de malhas específico no tratamento das geometrias das seções transversais dos componentes destes dispositivos, selecionando e testando o algoritmo mais adequado à tarefa. No Capítulo 5 deste trabalho, são apresentados os algoritmos encontrados na literatura que serão avaliados quanto à possibilidade de serem empregados no gerador de malhas para MEF e são devidamente avaliados com base em critérios definidos nos Capítulos 4 e 6. A execução das avaliações detalhadas destes métodos exige o conhecimento de fundamentos de geometria computacional, que são apresentados tanto no capítulo da revisão bibliográfica, como no Apêndice A.

2 Objetivos

Este trabalho visa a selecionar e implementar o método de geração automática de malhas mais adequado para a aplicação nas geometrias das seções transversais dos componentes de cabos umbilicais e tubos flexíveis. Deve-se selecionar ou adaptar o algoritmo automático de geração de malhas mais adequado para ser incorporado em um potencial futuro programa de cálculo estrutural de cabos umbilicais e tubos flexíveis que utilizaria o Método dos Elementos Finitos. A seleção de algoritmos se iniciará após uma busca na literatura por métodos geradores de malhas 2D, sendo encaminhada com base nos requisitos característicos da simulação estrutural local destes equipamentos e de suas geometrias típicas. Deve-se desenvolver, por fim, um protótipo de um programa computacional simples para a implementação e a verificação do algoritmo selecionado.

3 Metodologia

3.1 Fluxo de Execução do trabalho

Este trabalho é composto de duas etapas: a etapa de busca por um método gerador de malhas que seja adequado às particularidades das geometrias e das estruturas aqui focadas e a etapa de aplicação do método em um programa computacional próprio.

Na primeira etapa, buscaram-se métodos geradores de malhas bidimensionais na literatura e, em seguida, foram definidas as hipóteses do problema e uma série de critérios referentes aos métodos genéricos de geração de malha e às especificidades dos equipamentos aqui visados. É realizada uma pré-seleção de todos os métodos encontrados baseados nos critérios e nas hipóteses primárias. As hipóteses primárias são as características básicas que um método automático de geração de malha deve apresentar para ser considerado neste trabalho. Em seguida, os métodos nos textos encontrados são avaliados com uma série de critérios. Estes são divididos em três tipos:

Critérios excludentes gerais: Estas características são requerimentos gerais para um texto que apresenta um método automático de geração de malhas, independentemente da aplicação final da malha. Caso não as sigam, o texto é desconsiderado.

Critérios excludentes específicos: Estas características são requerimentos específicos devidos à aplicação do método gerador de malha nas geometrias das estruturas em foco neste trabalho. Caso não as sigam, o texto é desconsiderado.

Critérios graduais: São características que devem ser atendidas em maior ou menor grau pelos textos, quantificando a avaliação por notas. São utilizados após aplicado o filtro inicial dos trabalhos pelas hipóteses primárias e pelos critérios excludentes. O método que obtiver o maior somatório de notas será o escolhido.

A avaliação dos métodos e dos textos encontrados é discutida na Seção 6 deste trabalho, iniciando-se pela exclusão de textos que não atenderam às hipóteses primárias ou aos critérios excludentes. Por fim, já com um grupo filtrado de métodos, estes são avaliados com referência aos critérios graduais, permitindo mensurar qual é o método mais adequado para as especificações deste trabalho.

A segunda etapa é composta pela codificação e avaliação do método selecionado na etapa anterior em um programa simples e próprio para testes. O algoritmo em questão é revisado e estruturado de forma que seja passível de implementação computacional. A ferramenta que

incorpora o método selecionado deve exercer algumas funcionalidades mínimas como os módulos de geração de malha de programas comerciais ou abertos:

- Apresentar a malha mediante uma janela gráfica
- Criar geometrias ou carregá-las a partir de um arquivo de entrada com as informações geométricas provenientes de outra fonte (p. ex. informações de imagens ou programa de desenho).
- Possibilitar a análise detalhada das propriedades da malha como qualidade dos elementos, número de nós e de elementos.

Muitas ferramentas CAE comerciais ou abertas possuem um módulo de CAD incorporado. Entretanto, no programa de teste desenvolvido para este trabalho, o desenvolvimento de um CAD não está previsto no escopo. Desenvolver um módulo de CAD suficientemente genérico exige um longo tempo de trabalho. Ponderou-se, portanto, que devido à diferença de foco deste trabalho com a criação de um módulo CAD, optou-se pelo não desenvolvimento de um módulo para geração interativa de geometrias.

A fim de se atender às tarefas básicas citadas, pesquisou-se a linguagem computacional mais adequada ao trabalho, bem como a existência de *frameworks* e bibliotecas que auxiliem na construção de um programa de computador incorporando todas as funcionalidades de maneira mais eficiente para um usuário. A discussão é apresentada na Seção 8 mais a frente neste texto.

4 Definição dos Critérios Primários de Avaliação

A qualidade de malha é uma característica de grande importância na obtenção de resultados satisfatórios pelo Método dos Elementos Finitos. Melhores resultados são alcançados com malhas homogêneas e mais próximas da estruturada, contendo maior quantidade de elementos de alta qualidade. A qualidade dos elementos é definida com base na comparação com o elemento ideal, ou seja, o elemento base na formulação do método numérico. No caso de uma malha de triângulos, utiliza-se como referência um triângulo equilátero ou um triângulo retângulo isósceles. Já em malhas de quadriláteros, o critério de comparação de um elemento é a semelhança a um quadrado, identificando a diferença entre as laterais do elemento e de seus ângulos internos. No Apêndice D, explicam-se os tipos de malhas e alguns critérios para a medição da qualidade dos elementos.

Neste trabalho, os vários artigos encontrados na literatura devem ser avaliados a fim de se encontrar os métodos mais genéricos e mais adequados para o problema aqui apresentado. Definiu-se que os métodos geradores de malhas caracterizados como genéricos devem atender às seguintes hipóteses:

- A. Não haver limitação na qualidade da malha devido à orientação das fronteiras do domínio.

Certos domínios podem apresentar fronteiras próximas a retângulos ou outras geometrias regulares e, por isso, alguns métodos geradores de malhas, que criam nós internos ao domínio por meio de grades de pontos, geram elementos de boa qualidade quando sua grade é orientada adequadamente ao domínio. Entretanto, caso a grade seja desorientada em relação ao domínio, os elementos resultantes terão sua qualidade comprometida.

- B. Não deve ser necessária a busca por parametrizações matemáticas para relacionar as fronteiras entre si.

Alguns métodos geradores de malhas se baseiam nas equações que formam a fronteira do domínio para gerar gradualmente os elementos. Esta abordagem limita a aplicação do método para as geometrias específicas em cada caso.

Em seguida, avaliaram-se todos os métodos encontrados na literatura com o intuito de se obter, ao final, um conjunto de métodos pré-selecionados para uma segunda avaliação mais

detalhada. Assim, definiram-se critérios que direcionam a escolha de artigos mediante a adequação para o problema aqui proposto. Os critérios de seleção de método foram separados em dois tipos: os critérios gerais e os critérios específicos. Os gerais são listados como:

1. *Os elementos gerados devem ser próximos dos elementos de referência.*
2. *Quadriláteros gerados devem ser convexos.*
3. *Malhas de quadriláteros devem apresentar a maioria dos nós com valência 2 e 4 (O conceito de valência é apresentado no Apêndice D).*
4. *O artigo deve apresentar o método e não apenas citar sua utilização.*

Já os critérios específicos foram definidos com base nas características geométricas e estruturais próprias dos cabos umbilicais e tubos flexíveis, conforme é explicado abaixo:

1. *As malhas devem ser formadas apenas por quadriláteros ou apenas por triângulos a fim de manter sua homogeneidade*

Devido ao objetivo final deste trabalho focar em uma futura aplicação do método dos elementos finitos à malha criada, algumas medidas devem ser tomadas na etapa de criação da malha para favorecer melhores resultados numéricos. Malhas homogêneas permitem melhor controle sobre os cálculos, enquanto manipular uma malha heterogênea exige cuidados especiais em seu manuseio. Na maioria dos casos, a rotina de criação de malhas heterogêneas é interativa com o usuário, pois este indica quais tipos de elementos são mais indicados para cada região do domínio. Utilizam-se estas variações com o propósito de também reduzir o consumo de memória. Costuma-se adequar elementos de tipos e tamanhos diferentes de acordo com o que o usuário espera como resultado dos cálculos numéricos. Como este tipo de manipulação é específico para cada caso estudado, perde-se a aplicação genérica proposta aqui.

2. *Geometrias regulares como círculos, coroas e retângulos devem gerar malhas próximas de uma estruturada.*

A seção transversal destes equipamentos também costuma apresentar geometrias regulares. É usual que mangueiras e capas plásticas sejam representadas por coroas; algumas armaduras helicoidais e cabos elétricos são representados por círculos; enquanto, outras armaduras helicoidais apresentam formas retangulares. Este critério permite a identificação de métodos que apenas geram malhas sem reconsiderar o

posicionamento dos pontos disponíveis internamente ao domínio para gerar elementos de melhor qualidade.

3. *Regiões do domínio com pequenas distâncias entre as fronteiras não devem gerar elementos distorcidos.*

O método deve permitir a criação de elementos pequenos e de alta qualidade em regiões mais estreitas que outras no domínio. A criação de elementos de má qualidade proporciona uma menor precisão nos cálculos.

4. *O método deve tratar figuras com curvas, furos e vazios internos.*

Conforme observado na Figura 1.5, geometrias de mangueiras, capas plásticas e de *fillers* apresentam furos e vazios, devendo ser tratadas automaticamente.

5 Revisão Bibliográfica

Encontra-se disponível na literatura uma vasta variedade de métodos geradores de malhas, seja no âmbito 2D como no 3D, tornando não trivial a tarefa de classificá-los em diferentes gêneros. Foram encontrados, durante a pesquisa bibliográfica, alguns estudos que propõem o levantamento de características comuns entre eles para ordenar os muitos algoritmos de construção de malhas disponíveis, sendo esses estudos publicados com diferenças da ordem de 10 a 20 anos e sugerindo uma possível atualização dos algoritmos ao longo dos anos. Este campo é constantemente atualizado e revisto, pois muitos são os métodos e algoritmos novos criados ao longo dos anos, ou são algoritmos que propõem alterações de algum procedimento frequentemente utilizado na área de geração de malhas. Uma característica observada na maioria das publicações sobre algoritmos e sobre programas disponíveis é que cada revisão classifica os conjuntos de algoritmos encontrados baseando-se nos modelos e algoritmos vigentes na sua época, não fazendo, no entanto, menção às revisões anteriores sobre o assunto.

A seguir são apresentadas algumas revisões de algoritmos geradores de malhas sucedidas pela análise de métodos avulsos selecionados na literatura, que, em grande parte, são modificações de métodos já citados nas revisões. Aqui, as análises foram guiadas de forma que se verifique a viabilidade do uso dos métodos apresentados nas geometrias bidimensionais típicas dos cabos umbilicais e dos tubos flexíveis.

O estudo de Ho-Le (1988) propõe uma classificação dos métodos disponíveis na sua época de acordo com o tipo de abordagem na manipulação da geometria e com o tipo de entrada de dados (uma nuvem de pontos, um grupo de retas e pontos ou um conjunto de geometrias mais complexas), como descrito a seguir:

1. **Determinação inicial da topologia da malha (*Mesh topology first*):** Define-se que esta abordagem procura a aproximação de uma malha pré-montada para a geometria exata por meio do refinamento. No entanto, o autor não desenvolveu este tópico por alegar que este tipo de método é dependente de outros tipos que geram a malha a partir de geometrias ou de informações de pontos. Não se encaixa, por tanto, na classificação de métodos que criem malhas somente a partir de suas fronteiras.
2. **Criação inicial dos nós (*Nodes first*):** Esta abordagem se divide em 2 modelos:

Decomposição de topologia: A figura, já com suas fronteiras discretizadas em segmentos de reta, tem seus vértices conectados para formar triângulos. São inseridos mais nós nas arestas destes novos triângulos e ligam-se os novos nós aos antigos formando outros triângulos, e assim sucessivamente, refinando a malha. A Figura 5.1 mostra um esquema deste procedimento

Figura 5.1 - Exemplo de decomposição de topologia.

Fondeste: (Ho-Le, 1988).

Uma versão do método é apresentada por Wördenweber (1983), mas ao ser implementada em uma rotina automática, por ser uma versão menos desenvolvida desta abordagem, pode gerar inconsistências como segmentos passantes por furos ou por regiões côncavas ou fora do domínio da geometria. A Figura 5.2 exemplifica um caso de uma possível inconsistência e sua devida correção.

Figura 5.2 - Inconsistência na triangulação.

Fondeste: (Wördenweber, 1983).

Tal ocorrência exige algumas modificações e adaptações no método, resultando em particularidades não aplicáveis em um método genérico para

geração de malhas. Este algoritmo, no entanto, foi base para outros procedimentos mais atuais.

Abordagem de conexão de nós: É fornecida uma nuvem de pontos inicialmente e estes pontos são conectados formando triângulos ou retângulos. Aqui são incluídos procedimentos como a técnica de Cavendish (1974). Este método é próximo de ser considerado automático, pois o usuário deve fornecer como entrada apenas as fronteiras da figura já aproximada por nós e segmentos de reta. Segue-se com a criação de pontos no interior da figura, mas sem se definir um tipo de abordagem genérica. Ao final, os pontos são todos interligados, formando triângulos ou quadriláteros, conforme mostra a Figura 5.3. No caso de figuras mais complexas, a determinação de regiões mais refinadas ou não fica pouco automática, tornando o processo muito penoso ao usuário e sem uma clareza de eficiência garantida para geometrias genéricas.

Figura 5.3 - a) Figura com os nós gerados. b) Figura após a conexão dos nós

Fonte: próprio autor do texto.

Aqui fica evidente a possibilidade de criação de nós internos nos furos e espaços vazios no interior da forma geométrica. Isso se diz tratado no método proposto por Cavendish, no entanto, é apenas explicado um caso simples de um círculo com um furo circular de pequena dimensão comparada com o resto da figura, não sendo uma abordagem genérica. Faz-se necessário um tratamento da ocorrência de vazios internos de maneira análoga ao tratamento das inconsistências citado no método anterior, o que torna cada caso um problema específico, inviabilizando sua aplicação em processos automáticos genéricos. O método de Cavendish também não trata a entrada

de informações de geometrias parametrizadas, como é o caso de figuras curvilíneas complexas, sendo mostradas em seu trabalho apenas as geometrias com fronteiras já discretizadas como ponto de partida. Uma possível solução para generalizar as particularidades de cada geometria é o fornecimento destas informações qualitativas em união à entrada dos dados representativos de segmentos e curvas. Este modelo de estruturação de informações geométricas é chamado BRep (*Boundary Representation*) - verificar sua breve explanação no Apêndice C - que contém informações a mais sobre as particularidades da geometria analisada, comparativamente com o modelo CSG ou o *wireframe* (modelos de armazenamento de dados de geometria explicados no Apêndice C). O modelo BRep providencia melhor armazenamento das informações sobre as curvas e geometrias formadoras das fronteiras.

Neste grupo de abordagens por conexão de nós, também é incluída a triangulação de Delaunay, que propõe critérios na geração de triângulos em uma nuvem de pontos. É o procedimento mais adotado na conexão de pontos para a formação de triângulos. O procedimento de Delaunay será mais bem explicado adiante.

Varias outras adaptações destas triangulações e métodos semelhantes são apresentados na literatura da época, mas suas noções gerais e eventuais limitações na abordagem de geometrias irregulares já são englobadas pelos trabalhos aqui citados.

3. Modelo de malha adaptada: Esta abordagem se divide em 3 modelos:

Abordagem baseada em grade: uma grade de triângulos ou retângulos, maior que as dimensões da figura, é sobreposta a ela e suas fronteiras concatenadas com as bordas da figura. A malha resultante deste procedimento se torna bem definida no centro, mas com elementos mal formados nas periferias, com grande probabilidade de resultar em malhas não homogêneas. A Figura 5.4 mostra um exemplo de malha gerada por este procedimento.

Figura 5.4 - Exemplo de malha adaptada.

Fonte: (Ho-Le, 1988).

Na revisão de Ho-Le são listados vários casos que adotam tal abordagem, no entanto, a aplicação desta nas formas geométricas de cabos umbilicais e tubos flexíveis gerará uma variação considerável entre a qualidade dos elementos da fronteira e a qualidade dos elementos do interior da figura, além de gerar elementos triangulares e retangulares em uma mesma malha. Esta abordagem seria aplicável neste trabalho caso se adotasse outro método de manipulação de malha, promovendo um pós-processamento para homogeneizar os tipos de elementos. No entanto, isto foge da proposta determinada na Seção 2, pois o pós-processamento de malhas é um tema a parte e bastante vasto. Deve-se selecionar apenas um método que atenda ao máximo às características típicas dos equipamentos citados, pois a aplicação de vários métodos em uma mesma malha envolverá o processo de otimização de geração de malha. Pós-processamento de malhas é um assunto de vasta abrangência e exige um estudo direcionado.

Abordagem do elemento mapeado: cria-se uma grade de triângulos em um triângulo unitário ou uma grade de quadriláteros em um quadrado unitário. O usuário deve dividir a figura em regiões de 3 lados ou 4 lados, respectivamente, como se fossem macroelementos, e em seguida a grade triangular ou quadrangular, dependendo do caso, é mapeada nestas sub-

regiões da figura original. Este método não costuma ser utilizado em ferramentas de CAD/CAE/CAM, pois a divisão da figura em subregiões não é automático, se tornando um processo muito laborioso para potenciais usuários. Sendo assim, este trabalho não se focará em implementar algoritmos que não sejam automáticos. A Figura 5.5 mostra um exemplo representativo desta abordagem.

Figura 5.5 - a) Figura plana dividida em 2 macroelementos; b) Projeção da grade de um quadrado unitário para um dos macroelementos; c) Forma final da malha após o método ser aplicado às 2 divisões da figura geométrica.

Fonte: (Ho-Le, 1988)

Por outro lado, devido às particularidades das geometrias encontradas no objeto deste trabalho dependerem do tipo de componente analisado, este método pode ser adotado para o caso do surgimento de geometrias regulares o suficiente a ponto de possibilitarem o mapeamento da grade sem relevantes distorções nos elementos. Seções bidimensionais de mangueiras, arames, capas plásticas e cabos elétricos costumam apresentar composições de

geometrias regulares como círculos, coroas e retângulos, podendo elas serem submetidas a este tipo de método de maneira automática.

Abordagem do mapeamento conformado: um polígono padrão é utilizado como base para a criação de uma malha regular. Tal polígono de controle deve conter a mesma quantidade de vértices que a região da figura analisada. Esta malha de controle é projetada na região alvo. É possível utilizar esta abordagem com os macroelementos da abordagem apresentada anteriormente, de tal maneira que em vez de se utilizar uma malha triangular ou quadrada unitária, é utilizado um polígono de controle com n lados. A Figura 5.6 exemplifica uma aplicação do método.

Figura 5.6 - Exemplo esquemático do mapeamento conformado.

Fonte: próprio autor do texto.

Esta abordagem também não se caracteriza como automática e, por isso, este gênero de método não será tomado como possibilidade de algoritmos de geração de malha a ser adotado.

4. **Nós e elementos criados simultaneamente (Decomposição de geometria):** A figura é subsequentemente dividida em regiões, com os nós e os elementos criados simultaneamente às divisões, resultando em novos elementos a cada iteração de subdivisão da geometria. Este tipo de método é conhecido por

considerar as características geométricas e topológicas da figura simultaneamente. Os métodos citados a seguir são considerados os mais automáticos de todos os apresentados na revisão de Ho-Le. O algoritmo apresentado por Bykat (1976) propõe um procedimento automático de subdivisão da figura em regiões convexas e a construção de uma malha triangular próxima à estruturada, mas com algumas regiões mal condicionadas, principalmente regiões de fronteira com ângulos agudos (por exemplo figuras côncavas). O método proposto por Tracy (1977) também garante a subdivisão da figura por regiões automaticamente, no entanto, o procedimento não fica claro no artigo. Já o procedimento proposto por (Sadek, 1980) se direciona para a criação de malhas triangulares apenas, seguindo uma série de critérios para a criação dos triângulos e dos novos nós da figura geométrica. É citado também o método de Lindholm (1983) que é outro procedimento semelhante ao anterior para a criação de malhas totalmente triangulares. A Figura 5.7 mostra um esquema do processo de uma iteração de subdivisão da geometria. Geralmente, neste gênero de método, os elementos são criados a partir das periferias e das fronteiras da figura em direção ao seu centro, respeitando a densidade de nós definida no início do processo.

Figura 5.7 - Exemplo da abordagem de decomposição geométrica.

Fonte: próprio autor do texto.

Os métodos citados por Ho-Le sugerem como se encontrava o estado da arte no final da década de 70 e em meados da década de 80, de acordo com a abrangência de sua pesquisa, quando os atualmente reconhecidos programas de simulação pelo método dos elementos finitos estavam começando a ser desenvolvidos.

Em um período mais recente, tem-se o estudo de Owen (1998) que propõe outra revisão dos métodos de geração de malha desenvolvidos na época e ainda analisa rapidamente os módulos geradores de malhas dos pacotes computacionais mais utilizados na indústria e na academia. No seu trabalho, são citados algoritmos geradores de malhas de triângulos, quadriláteros, tetraedros e hexaedros. A seguir são mencionados apenas os processos de geração de malhas 2D presentes no artigo, ou seja, os algoritmos referentes a malhas de triângulos e de quadriláteros.

Owen classifica os tipos de abordagem dos métodos para cada tipo de elemento criado, triângulo ou quadrilátero. No caso da geração de malhas de elemento triangular, têm-se as seguintes abordagens:

Método *Quad-Tree*: Uma malha de quadrados de tamanhos variados é “ajustada” à figura geométrica, de maneira que os quadrados intersectem suas fronteiras. Esses pontos de intersecção são conectados por um segmento de reta em cada quadrado passante pela fronteira. Feito isso, os polígonos são todos divididos em triângulos. O trabalho de Yerry e Shephard (1984) é um exemplo representativo destes tipos de métodos. A Figura 5.8 mostra uma geometria discretizada pelo método *Quad-Tree*.

Figura 5.8 - Modelo de uma geometria discretizada pelo método *Quad-Tree*.

Fonte: (Yerry & Shephard, 1984).

Nota-se que nas proximidades das regiões curvas são criados triângulos mal formados, ou seja, que apresentam grande distorção, em comparação com os elementos mais próximos da fronteira externa. A ocorrência de elementos mal formados prejudica a qualidade da malha e, conseqüentemente, resulta em piores resultados dos cálculos pelo Método dos Elementos Finitos. Por isso, o uso desta abordagem se torna pouco interessante neste trabalho.

Triangulação de Delaunay: Os métodos que se baseiam na triangulação de Delaunay determinam um tipo classificatório, como definido por Owen, enquanto na classificação de Ho-Le eles pertencem ao tipo Abordagem por Conexão de Nós. Apesar do tipo de classificação, é consensual que a triangulação de Delaunay é tida como um conjunto de critérios de conexão de nós de uma nuvem de pontos e não como um método gerador de malha, se fazendo necessário o associar a outro procedimento que gere pontos internos à geometria. A explicação da Triangulação de Delaunay pode ser realizada a partir de algumas propriedades:

- Diagrama Voronoi: Dada a nuvem de pontos $S = \{p_1, p_2, \dots, p_n\}$ de n pontos no espaço R^2 , para cada p_i , define-se uma região $R(S; p_i)$ no plano por

$$R(S; p_i) = \{p \in R^2 \mid \|p - p_i\| \leq \|p - p_j\|, \quad i \neq j\} \quad (5.1)$$

que é chamada Região de Voronoi de p_i , de modo que $\|p - q\|$ representa a distância Euclidiana entre dois pontos p e q . O que significa que a região $R(S; p_i)$ é formada pelo conjunto de pontos cuja distância até p_i é menor ou igual à distância até qualquer outro ponto da nuvem. Ao se delimitar todas as Regiões de Voronoi para cada ponto da nuvem, obtém-se o Diagrama de Voronoi de S , expresso por $VD(S)$. A Figura 5.9 mostra um exemplo de um Diagrama de Voronoi e nela se podem notar segmentos de reta que formam as fronteiras destas regiões, denominados como Arestas de Voronoi.

Figura 5.9 - Diagrama de Voronoi.

Fonte: (Farin, Hoschek, & Kim, 2002).

É possível que ocorram situações específicas na disposição dos pontos da nuvem. Diz-se que o diagrama está degenerado quando 4 ou mais pontos estão posicionados no formato de um círculo sem nenhum outro ponto no seu interior, levando as Regiões de Voronoi a convergirem em um único ponto central. A Figura 5.10 expõe um exemplo desta ocorrência.

Figura 5.10 - Diagrama de Voronoi degenerado.

Fonte: (Farin, Hoschek, & Kim, 2002).

O Diagrama de Delaunay de S , escrito como $DD(S)$, é obtido a partir do $VD(S)$ apenas unindo cada 2 pontos da nuvem S por segmentos de reta e passando pelas Arestas de Voronoi. Dados 2 pontos p_i e p_j da nuvem de pontos, com $i \neq j$, esses pontos só serão conectados caso compartilhem uma Aresta de Voronoi de $VD(S)$. O $DD(S)$ resultante do $VD(S)$ da Figura 5.10 é representado na Figura 5.11. É importante salientar que no caso de um Diagrama de Voronoi degenerado, ao serem conectados os pontos da nuvem, não serão obtidos apenas triângulos, mas também polígonos de mais de 3 lados, o que pode ser resolvido dividindo estes polígonos em outros triângulos, que é parte do processo de métodos geradores de malha que adotam a Triangulação de Delaunay.

Figura 5.11 - Diagrama de Delaunay.

Fonte: (Farin, Hoschek, & Kim, 2002).

- Propriedade dos círculos circunscritos: esta propriedade é obtida a partir do Diagrama de Voronoi. Define-se que cada triângulo formado pelo Diagrama de Delaunay é circunscrito por um círculo que não contém nenhum outro ponto de controle da nuvem de pontos dada em seu interior que não sejam

os 3 formadores do triângulo. Algebricamente, tomando-se uma função $F(p_i, p_j, p_k, p_l)$ como:

$$F(p_i, p_j, p_k, p_l) = \begin{vmatrix} 1 & x_i & y_i & x_i^2 + y_i^2 \\ 1 & x_j & y_j & x_j^2 + y_j^2 \\ 1 & x_k & y_k & x_k^2 + y_k^2 \\ 1 & x_l & y_l & x_l^2 + y_l^2 \end{vmatrix} \quad (5.2)$$

onde (p_i, p_j, p_k) são os pontos formadores de um triângulo de Delaunay e $F(p_i, p_j, p_k, p_l)$ é a função que verifica se o ponto p_l está contido no círculo que circunscreve o triângulo $\Delta p_i p_j p_k$. Caso os pontos (p_i, p_j, p_k) sejam listados no sentido anti-horário, a função F assume os seguintes valores:

$$\begin{cases} F(p_i, p_j, p_k, p_l) > 0, & \text{se } p_l \text{ é um ponto fora da circunferência} \\ & \text{circunscrita em torno de } \Delta p_i p_j p_k. \\ F(p_i, p_j, p_k, p_l) < 0, & \text{se } p_l \text{ é um ponto interno à circunferência} \\ & \text{circunscrita em torno de } \Delta p_i p_j p_k. \\ F(p_i, p_j, p_k, p_l) = 0, & \text{se } p_l \text{ é um ponto pertencente à circunferência} \\ & \text{circunscrita em torno de } \Delta p_i p_j p_k. \end{cases}$$

- Inserção de pontos: Esta etapa precede a aplicação dos critérios de triangulação de Delaunay e se caracteriza pela criação de nós nas fronteiras da figura e em seu interior. A revisão de Owen cita alguns trabalhos anteriores que apresentam algoritmos para criar estes nós adicionais. Ruppert (1992) propõe um método de inserção de nós na geometria para a triangulação de Delaunay: os novos nós internos são criados preferencialmente próximos aos triângulos de menor ângulo, por meio da inserção do centro do círculo circunscrito a um triângulo. Além disso, os triângulos são formados considerando a casca convexa, ou *convex hull*, formada pelos nós mais externos da fronteira da figura. Esta abordagem resulta em triângulos sendo formados em espaços vazios, se houver. Ao final do processo, os triângulos nas áreas de vazio podem ser removidos por algum algoritmo específico. Uma desvantagem deste procedimento é a possibilidade de o algoritmo exigir muito tempo de processamento ao considerar espaços vazios na triangulação.

O tipo de método proposto por Ruppert é classificado como de “Qualidade Garantida dos Elementos”, pois, junto de vários outros métodos baseados na triangulação de Delaunay qualificados desta maneira na literatura, propõe a criação de triângulos de tamanho minimamente controlado.

Outra forma de gerar nós na figura de maneira similar à adição do nó central do círculo circunscrito a um elemento triangular, foi apresentada por Rebay (1993), que sugere 2 metodologias considerando o Diagrama de Voronoi. Ambas são iniciadas mediante uma triangulação de Delaunay entre os pontos da(s) fronteira(s), gerando triângulos em regiões fora da malha ou internos a ela. Diferentemente do algoritmo de Ruppert, estes métodos não investem tempo de processamento na criação de triângulos de boa qualidade nas regiões fora da figura. O primeiro método, mais simples, adiciona pontos no interior de triângulos já criados, no centro da circunferência que circunscreve o triângulo trabalhado. Já o segundo método é mais criterioso, inserindo pontos nas regiões de triângulos já criados, ao longo das Arestas de Voronoi. Foi apresentado outro tipo de abordagem de criação de nós com a Triangulação de Delaunay em que estes são criados baseados em uma frente de avanço interna partindo das fronteiras. Marcum & Weatherill (1995) propõem que os pontos sejam inseridos conforme a fronteira avança para o interior da figura em passadas acumulativas. O artigo, no entanto, apresenta uma explicação superficial do método, que é proposto tanto para os casos 2D quanto para os 3D, de modo genérico, dificultando seu entendimento.

As maneiras de inserir pontos na Triangulação de Delaunay citadas por Owen mostram como este tipo de abordagem de geração de malhas triangulares foi um dos mais utilizados e desenvolvidos desde a revisão de Ho-Le.

Frente de Avanço: Além da tão reconhecida Triangulação de Delaunay, Owen também caracteriza essa outra vertente de geração de malhas triangulares importante de ser destacada. Versões de métodos baseados nesta abordagem foram desenvolvidas por vários estudos, e alguns foram incorporados em pacotes de geradores de malhas de programas comerciais, como o caso da ferramenta ANSYS™. Lo (1985) propõe o uso da frente de avanço como um método automático de geração de malhas triangulares a partir, a princípio, de qualquer

domínio planar. Os novos nós são inseridos inicialmente nas fronteiras, as discretizando para um tamanho de aresta de elemento médio (pré-estabelecido pelo usuário) para esta malha. Em seguida, uma grade de pontos é inserida no interior da figura, verificando se os pontos são adicionados nas regiões de vazio por meio de critérios apresentados no artigo. A abordagem final baseia-se principalmente na triangulação de pontos internos em frentes de arestas formadas com a sucessiva conexão dos nós internos à figura. A Figura 5.12 mostra um esquema da sequência de execução deste algoritmo.

Figura 5.12 - Sequência de triangulação por frente de avanço.

Fonte: (Lo, 1985).

Este procedimento se caracteriza por os elementos serem criados inicialmente próximos às fronteiras e seguindo para a expansão destas frentes para o interior da figura, tanto no caso da frente proveniente da fronteira externa, como no caso das frentes das fronteiras dos furos e dos espaços vazios internos.

Um dos artigos que apresentaram pela primeira vez a frente de avanço é o algoritmo proposto por Peraire, *et al.* (1987). O seu processo de geração de novos elementos foi seguido no trabalho de Lo em se tratando de gerar os novos elementos a partir das fronteiras que avançam ao longo do interior da figura, mas se difere na etapa de criação de nós. Na versão de Peraire, *et al.* (1987), os novos nós não são criados por uma grade de pontos, adiciona-se cada ponto conforme os elementos são criados pela projeção dos nós de fronteiras. Vale salientar que este tipo de procedimento se assemelha àquele classificado por Ho-Le como Decomposição de Geometria e que o algoritmo proposto por Peraire, *et al.* (1987) também exige a divisão do domínio em sub-regiões, na malha de fundo, devendo ser feita de maneira manual e tornando o método não automático.

Verificou-se a possibilidade de tornar este método mais automático realizando-se algumas adaptações.

É citado também o trabalho de Pirzadeh (1993) que propõe uma versão da frente de avanço em que os nós criados no interior da figura podem ser submetidos a uma variação do módulo de avanço para o interior para cada região, resultando em regiões com triângulos menores e outras com triângulos maiores, dependendo da taxa de variação do tamanho do elemento em cada iteração da frente de avanço ao longo do interior da figura. O método difere do proposto por Lo e por Peraire, *et al.* por quantificar os vetores de avanço para cada nó da fronteira atual através de uma função de suavização de Laplace, definida por Pirzadeh como:

$$v_p^{i+1} = (1 - \omega)v_p^i + \frac{\omega}{N_p} \sum_{k=1}^{N_p} v_k \quad (5.3)$$

onde v_p é o vetor resultante da soma dos vetores normais das arestas que se conectam ao nó p , v_k é a resultante da soma dos vetores normais das arestas conectadas ao nó k , sendo os nós k aqueles que estão conectados a p , ω é um fator de relaxação ($0 < \omega < 2$), i é estágio da iteração e N_p é o número de nós que se ligam ao nó p . A Figura 5.13 mostra um esquema de uma região da fronteira com os vetores de avanço nos nós.

Figura 5.13 - a) Normais das arestas da fronteira. b) Normais dos nós da fronteira. c) Normais dos nós após a suavização de Laplace.

Fonte: (Pirzadeh, 1993).

O artigo da versão da frente de avanço proposta por Pirzadeh não explica, no entanto, como devem ser tratadas as eventuais intersecções entre elementos criados.

Owen também cita alguns métodos geradores de malhas retangulares em sua revisão, dividindo-os em dois tipos, os indiretos e os diretos. Os métodos indiretos são algoritmos de manipulação de malhas pré-criadas e que contêm elementos triangulares para serem transformados em quadriláteros. Por serem métodos de pós-processamento de malhas, o presente trabalho não fará menção a estes tipos de algoritmos.

Até o momento foram apresentadas as versões de métodos geradores de malhas com uma abordagem mais genérica, envolvendo tanto malhas triangulares, quanto retangulares e mistas. Assim, a partir deste trecho, o foco será direcionado nos métodos diretos de geração de malhas retangulares, que são divididos em algumas vertentes por Owen:

- Geração de Malha Retangular por Decomposição: Neste tipo de método, a figura é dividida em uma "quad-tree" ou árvore de quadriláteros, onde suas áreas são

ajustadas em regiões retangulares e, por fim, os nós da fronteira são ajustados para tornar suas curvas e linhas aproximadas por elementos retangulares. Baehmann, *et al.* (1987) propõem um método que cria inicialmente uma malha grosseira de retângulos de variados tamanhos preenchendo o espaço da figura e se ajustando às fronteiras que os intersectam por segmentos de reta. Em seguida são feitas várias manipulações nesta malha grosseira para formar elementos triangulares e retangulares. Cada elemento é verificado quanto ao posicionamento de seus nós, aos ângulos internos e até com quais elementos vizinhos são compartilhadas arestas. Após certa etapa do algoritmo é possível tornar a malha composta apenas por triângulos ou por quadriláteros, sendo possível até se desenvolver uma malha mista com os dois tipos de elementos. A Figura 5.14 mostra uma malha grosseira e uma no estágio refinado apenas com quadriláteros.

Figura 5.14 - a) Malha refinada com apenas quadriláteros. b) Malha grosseira do método "quadtree".

Fonte: (Baehmann, *et al.*, 1987).

Verifica-se que este tipo de método é pouco interessante de ser adotado neste trabalho, pois não cumpre a primeira hipótese na seleção do algoritmo. A qualidade da malha gerada fica diretamente dependente da orientação da grade sobreposta à figura e da orientação de suas próprias fronteiras.

Outro trabalho, apresentado por Talbert e Parkinson (1990), propõe um método dito prático de geração de malha totalmente composta por quadriláteros e que pode ser aplicado a qualquer geometria, considerando que se defina na figura o conjunto de loops fechados de curvas e segmentos de reta. O algoritmo foi desenvolvido primariamente para gerar malhas em figuras de formas curvas, que

no artigo são representadas por curvas de Bézier, mas pode ser aplicado a geometrias regulares. O método é considerado automático, pois exige poucas informações do usuário para o processo: as curvas que representam as fronteiras, o tamanho do elemento quadrado (medida do seu lado) e o fator de refinamento de malha. O algoritmo tem como base a sucessiva divisão da figura em partes menores, considerando a configuração que gerará os elementos de maior qualidade. No entanto, é apresentada uma limitação na automatização de entrada de informação que é a necessidade de o usuário definir linhas que conectam fronteiras internas a externas no caso de a figura apresentar furos e espaços vazios. Estas linhas são chamadas de linhas de corte e são utilizadas para delimitar as sub-regiões do domínio (Outros métodos aqui apresentados propõem procedimentos para que esta etapa seja também automática, mas não ficaram claros na explicação). A Figura 5.15 expõe um exemplo de malha gerada a partir deste método.

Figura 5.15 - Malha de quadriláteros da seção de uma chave inglesa.

Fonte: (Talbert & Parkinson, 1990).

Owen cita o trabalho de Chae e Jeong (1997) que propõe um aperfeiçoamento do trabalho de Talbert e Parkinson, adaptando a geração de malhas retangulares em figuras bidimensionais para gerar malhas em planos e em superfícies, realizando algumas alterações nos procedimentos de montagem da malha. Talbert e Parkinson se utilizam de operadores de 6 nós, que são maneiras de

gerar elementos quadriláteros em configurações típicas de figuras fechadas (ou loops) de 6 nós formadas durante a geração da malha. Já Chae e Jeong propõem o uso de um operador de 8 nós, alegando melhores resultados. O ponto principal do seu trabalho é a capacidade de gerar e projetar as malhas de quadriláteros nas superfícies do espaço 3D.

Outro trabalho semelhante ao de Talbert e Parkinson é o proposto por Nowotny (1997), que apresenta um processo bem semelhante ao de divisão da figura em sub-regiões, apresentando uma função de otimização do posicionamento da linha de corte de regiões. Um ponto importante que difere os dois algoritmos é o tratamento de furos e espaços vazios. Nowotny propõe uma otimização de posicionamento da linha de conexão entre as fronteiras (nó-figura ou nó-nó) e o encontro da melhor configuração para a conexão destas fronteiras inicialmente isoladas, enquanto Talbert e Parkinson direcionam esta tarefa para o usuário, não especificando maneiras de realizar esta tarefa automaticamente. A Figura 5.16 mostra um exemplo de linha de união entre a fronteira externa e um furo pelo algoritmo de Nowotny.

Figura 5.16 - Exemplo de conexão de fronteiras interna e externa.

Fonte: (Nowotny, 1997).

- Geração de Malha Retangular por Subdivisão por Eixo Médio: Foi citada também outra vertente de métodos de subdivisão de domínio conhecida como subdivisão por eixo médio, introduzido inicialmente por Tam e Armstrong (1991). O procedimento desenvolvido por Tam e Armstrong para a obtenção dos eixos médios é baseado na aplicação inicial da Triangulação de Delaunay na

figura, utilizando apenas os nós das fronteiras, mas no artigo é omitida a forma com que deve ser feito este processo. Após gerar os triângulos de Delaunay, são mapeadas todas as circunferências circunscritas a eles a fim de determinar seus pontos centrais, pois pertencem aos eixos médios da figura. Tendo determinadas as subdivisões limitadas pelos eixos médios e as fronteiras, divide-se cada uma em malhas de quadriláteros, procedimento que também não é explicado no artigo. A Figura 5.17 mostra um esquema das etapas da geração de malha por este método.

Figura 5.17 - a) Geometria com os eixos médios. b) Remoção das concavidades. c) Divisão das sub-regiões. d) Geração das malhas por sub-região.

Fonte: (Tam & Armstrong, 1991).

Seguindo a vertente de geração de malhas de quadriláteros por decomposição da geometria, outro trabalho citado foi o de Joe (1995), que propõe um método de geração de malha baseado na divisão da figura em vários polígonos. Estas sub-regiões têm suas arestas segmentadas adicionando-se nós extras conforme critérios de tamanho de elementos e densidade de malha. Por fim, grades de quadriláteros são acopladas a cada sub-região e seus elementos ajustados. Este tipo de abordagem se assemelha à Abordagem por Elemento Mapeado proposta por Ho-Le, no entanto sendo agora apresentada de maneira automática, sugerindo o avanço destes métodos ao longo dos anos. A Figura 5.18

exemplifica as etapas de divisão da área da geometria e da malha gerada ao final. Joe, no entanto, não explica detalhadamente o método, apenas mostrando parte de seu código, na forma de pseudo-código, sem explicar qual é o significado das variáveis e das funções apresentadas.

Figura 5.18 - a) Geometria dividida em sub-regiões e com as bordas segmentadas. b) Malha de quadriláteros criada a partir das sub-regiões.

Fonte: (Joe, 1995).

- Geração de Malha Retangular por Frente de Avanço: O último tipo de abordagem dentre os métodos diretos de geração de malha citados por Owen em sua revisão é a frente de avanço para elementos quadráticos. Zhu, *et al.* (1991) propõem uma adaptação do método de frente de avanço para criação de triângulos, para os unir e para formar quadriláteros. Desenvolve-se o método a partir do algoritmo proposto por Peraire, *et al.* (1987) com algumas alterações para ajustar os triângulos a ponto de favorecer suas uniões em quadriláteros. Inicia-se o processo da mesma maneira que a proposta por Peraire, *et al.*: Figuras com furos ou vazios internos devem ter as já citadas linhas de corte adicionadas, mas Zhu, *et al.* não deixam claro se esta etapa é automática ou se deve ser fornecida como entrada pelo usuário, nem como realizá-la. As etapas para a formação dos elementos e das frentes de avanços são:
 - a. Discretização das fronteiras por critérios de adição de nós;
 - b. Projeção de nós da fronteira para o interior da figura com critérios de afastamento;
 - c. Montagem dos triângulos pela associação dos nós das fronteiras com os nós criados no interior da figura;
 - d. União dos triângulos para gerar quadriláteros;

- e. Verificação da malha para corrigir determinadas configurações de elementos ou otimizar regiões.

Os resultados apresentados neste artigo são interessantes, principalmente com relação aos exemplos com furos e vazios, pois são geometrias semelhantes às encontradas em alguns *fillers* dos cabos umbilicais.

Outro tipo de método que segue os padrões da frente de avanço é o da pavimentação (*paving*). Blacker e Stephenson (1991) têm o primeiro trabalho registrado que apresenta a aplicação da pavimentação em figuras planas. O algoritmo é baseado no preenchimento das fronteiras, dadas como entrada pelo usuário, com elementos sendo adicionados sucessivamente ao longo do seu interior. Na Figura 5.19 é apresentado um esquema do processo de inserção de elementos nas fronteiras de avanço e pode-se observar que os elementos são adicionados em sentido horário nas fronteiras referentes aos furos e vazios internos, enquanto seguem um sentido anti-horário de adição na fronteira externa.

Figura 5.19 - Modelo do processo de preenchimento das frentes de avanço por elementos.

Fonte: (Blacker & Stephenson, 1991).

A propagação das frentes de avanço se dá por uma série de verificações dos elementos criados. Conforme é montada, a malha sofre alterações sucessivas como adição de fila de elementos, ajustes de nós, remoção de quadriláteros, suavização, verificação de intersecção de frentes de avanço, entre outras. Os exemplos apresentados no artigo e a descrição do algoritmo são razoavelmente

detalhados. A Figura 5.20 apresenta uma série de estados da malha ao longo do processo de geração dos quadriláteros.

Figura 5.20 - Serie de a até i de estados da malha durante a geração dos quadriláteros.

Fonte: (Blacker & Stephenson, 1991).

O método de pavimentação foi muito aceito pela comunidade científica e também pela Indústria, sendo implementado em ferramentas reconhecidas como Fluent e MSC Patran.

Ao longo da revisão do trabalho de Owen, notou-se que a classificação dos tipos de métodos geradores de malhas sofreu adaptações desde o trabalho de Ho-Le, mesmo este não

sendo citado por Owen. Houve uma evolução das abordagens com a tendência de tornar os algoritmos mais automáticos, reduzindo as entradas do usuário para apenas a geometria, o tipo de elemento desejado e a densidade de elementos. Isso consolidou o uso de alguns métodos em ferramentas comerciais e acadêmicas, tornando possível a geração de malhas, sejam 2D ou 3D, de geometrias complexas das peças e componentes projetados atualmente de maneira eficiente.

A tese de doutorado de Fenwick (2005) apresenta uma breve revisão de métodos de geração de malhas triangulares para modelos 2D e de malhas de tetraedros para modelos 3D. Os métodos apresentados, em sua maioria, utilizam o critério de Delaunay para formar os elementos, mas têm naturezas variadas na criação de novos nós. A autora divide os métodos que, durante uma etapa de processamento, utilizam o critério de Delaunay em dois tipos: Os geradores de malha por casca convexa (*Convex Hull*) e os geradores de malha por BRep (*Boundary Representation*). No caso do primeiro, a malha é criada apenas formando os elementos com base nos vértices de entrada. Com relação aos métodos que utilizam o modelo BRep, os vértices são criados no interior da figura de maneira mais criteriosa e são consideradas as geometrias das fronteiras internas e externas. Nota-se que os tipos de métodos classificados como geradores por BRep se assemelham aos tipos citados por Owen, o qual foi citado por Fenwick. Seu trabalho propõe, por fim, um modelo de algoritmo para geração de malha triangular não estruturada considerando o critério de Delaunay.

Outra revisão de métodos de geração de malhas encontrada na literatura foi a de Bommers, *et al.* (2012) que é direcionada a malhas de quadriláteros. Esta revisão, no entanto, é muito genérica e engloba majoritariamente algoritmos aplicados em modelos 3D e de otimização de malhas pré-processadas, não sendo, por isso, desenvolvida uma leitura mais detalhada deste artigo. Nota-se que este estudo não cita as revisões de Ho-Le e de Owen, que analisam o estado da arte de métodos 2D e 3D de suas épocas, já que alguns dos algoritmos apresentados nestes estudos foram consolidados em programas comerciais geradores de malhas que vem sendo desenvolvidos desde a década de 90 até os dias de hoje.

Além dos trabalhos de revisão encontrados na literatura, existe uma vasta gama de diferentes versões e ajustes das vertentes apresentadas aqui. Se a maioria fosse tomada para análise, o tempo demandado de trabalho se tornaria impraticável. Foram selecionados alguns trabalhos avulsos de interessante citação por caracterizar mais uma parte importante do estado da arte ou por apresentarem métodos interessantes considerando o foco deste trabalho. Adiante são citados alguns outros métodos encontrados na pesquisa bibliográfica e que possuem alguma relação com as vertentes citadas.

O método proposto por Azarenok (2012) segue uma abordagem de mapeamento de malha no domínio. No entanto, utiliza para isso o cálculo variacional e não é especificada a abrangência desta abordagem para domínios de geometria genérica. As revisões citadas anteriormente também não enfatizaram este tipo de abordagem paramétrica. O artigo direciona os exemplos para malhas de mecânica dos fluidos computacional, apresentando apenas exemplos de malhas com padrões típicos deste tipo de simulação. Azarenok não menciona a usabilidade do algoritmo em simulações de estruturas pelo Método dos Elementos Finitos, nem a sua eficiência com figuras genéricas, tornando difícil uma avaliação mais criteriosa da sua aplicabilidade no foco do trabalho aqui apresentado.

Entre os vários métodos apresentados nas revisões, a vertente da frente de avanço e seus correlatos é uma das que se mostraram mais promissoras para este trabalho e tem sido largamente adotada em programas geradores de malhas ao longo dos anos. A eficiência do método de pavimentação, por exemplo, fomentou novas versões do método e novos estudos visando sua melhoria até os anos atuais. Ma e Zhao (2014) apresentam um estudo nesta linha, propondo um algoritmo que corrige uma dita falha de robustez da pavimentação. Alega-se que em casos de duas frentes de avanço muito próximas e com elementos relativamente grandes ocorrem interferências de difícil tratamento (Figura 5.21a), e também são apontados problemas no caso de duas frentes de avanço opostas e com tamanhos de elementos muito diferentes, gerando problemas de acoplamento entre elementos (Figura 5.21b). É apresentado um algoritmo que trata previamente as fronteiras inserindo linhas de corte, sendo esta etapa bem explicada. O restante do algoritmo não é suficientemente apresentado e são mostrados resultados de figuras tridimensionais, o que se destoa do algoritmo apresentado no artigo. Promover a ampliação da abordagem do método de pavimentação do plano 2D para um ambiente 3D não é uma tarefa trivial.

Figura 5.21 - a) Duas frentes de avanço se intersectando devido a elementos muito grandes. b) Duas frentes de avanço opostas e com elementos de tamanhos muito diferentes

Fonte: (Ma & Zhao, 2014).

Choi (2011) também cita estes eventuais problemas do algoritmo de pavimentação e propõe um método novo de geração de malhas de quadriláteros chamado método de centralização. O algoritmo baseia-se na vertente de subdivisão do domínio em regiões menores por meio das linhas de corte criadas, de acordo com o autor, automaticamente. Em seguida, são criados nós centrais em cada sub-região, tomando-os como pivôs das subseqüentes divisões para a geração dos elementos quadriláteros. A Figura 5.22 mostra um exemplo esquemático de malha gerada por este método e nota-se que os elementos próximos a algumas bordas estão pouco próximos de uma malha estruturada, o que destoa parcialmente da pavimentação, que inicia a formação dos elementos a partir das fronteiras e que tende a os deixar com melhor qualidade.

Figura 5.22 - Exemplo de malha de quadriláteros gerada pelo método de centralização.

Fonte: (Choi, 2011).

Encontra-se, na literatura mais recente, outra vertente pouco referenciada nas revisões citadas aqui, mas que possui considerável nicho de aplicação. Foram encontrados alguns estudos referentes a essa linha de pesquisa definida como geração de malha auxiliada por processamento paralelo. Chernikov e Chrisochoides (2008), por exemplo, publicaram um estudo de um método gerador de malhas triangulares com base nos critérios de Delaunay. O método impõe que as fronteiras iniciais são comuns a todos os processadores, sendo todos atualizados continuamente caso elas sejam alteradas durante a geração dos elementos. O domínio é dividido em sub-regiões que são distribuídas entre os processadores, os elementos são criados separadamente em cada processador e toda a malha é unificada ao final do processo.

Rodriguez, *et al.* (2013) apresentam outra proposta de método de geração de malha por processamento paralelo. Propõem construir uma malha de triângulos baseando-se no método da frente de avanço, mantendo as fronteiras iniciais comuns a todos os processadores. A figura é dividida em regiões distribuídas entre os processadores, e depois que os elementos são gerados, são interligadas novamente, seguindo o mesmo processo do método de Chernikov e Chrisochoides. A geração de malhas por processamento paralelo é um procedimento cada vez mais adotado atualmente devido à complexidade dos modelos computacionais e à tendência

dos computadores aumentarem seu poder de processamento com o aumento do número de núcleos. Este estudo pode se tornar uma linha a ser seguida em trabalhos futuros na criação de malhas tridimensionais de cabos umbilicais e tubos flexíveis, pois também se costuma analisar casos complexos de malhas com centenas de milhares de elementos, quantidades essas que costumam exigir muito processamento computacional, sendo excessiva até para *Workstations* dedicadas.

Após a análise generalizada das revisões e dos artigos encontrados na literatura, inicia-se a avaliação dos critérios e a execução da etapa de seleção dos métodos geradores de malhas 2D encontrados, avaliados como possíveis candidatos para a solução do problema proposto. O método deverá ser apresentado de maneira própria e convincente, com o máximo de detalhamento das etapas, para a implementação do método escolhido em um programa que permita o tratamento das geometrias típicas das seções transversais dos componentes de cabos umbilicais e tubos flexíveis. Assim, foi dedicada maior atenção aos artigos mais claros e que se mostraram como a opção menos laboriosa, mantendo a qualidade da malha a maior possível.

6 Análise Detalhada dos Métodos

6.1 Pré-Seleção de Métodos

A qualidade de malha é uma característica de grande importância na obtenção de resultados satisfatórios através do Método dos Elementos Finitos. Melhores resultados são alcançados com malhas homogêneas e mais próximas da estruturada, contendo maior quantidade de elementos de alta qualidade. A qualidade dos elementos é definida com base na comparação com o elemento ideal, ou seja, o elemento base na formulação do método. No caso de uma malha de triângulos, utiliza-se como referência um triângulo equilátero ou um triângulo retângulo isósceles. Já em malhas de quadriláteros, o critério de comparação de um elemento é a semelhança a um quadrado, identificando a diferença entre as laterais do elemento e de seus ângulos internos. Retomando as hipóteses e os critérios apresentados na Seção 4:

Hipóteses:

- A. Não haver limitação na qualidade da malha devido à orientação das fronteiras do domínio;
- B. Não deve ser necessária a busca por parametrizações matemáticas para relacionar as fronteiras entre si.

Critérios gerais:

- 1. Os elementos gerados devem ser próximos dos elementos de referência.
- 2. Quadriláteros gerados devem ser convexos.
- 3. Malhas de quadriláteros devem apresentar a maioria dos nós com valência 2 e 4 (O conceito de valência é apresentado no Apêndice D).
- 4. O artigo deve apresentar o método e não apenas citar sua utilização.

Critérios específicos:

- 1. As malhas devem ser formadas apenas por quadriláteros ou apenas por triângulos a fim de se manter sua homogeneidade.
- 2. Geometrias regulares como círculos, coroas e retângulos devem gerar malhas próximas de uma estruturada.

Tabela 6.1 - Análise preliminar dos métodos geradores de malhas encontrados na literatura (conclusão).

Artigo	Hipóteses		Critérios Gerais				Critérios Específicos			
	A	B	1	2	3	4	1	2	3	4
(Lohner, Parikh, & Gumbert, 1988)	X	X	X	X	X		X	X	X	X
(Ma & Zhao, 2014)	X	X	X	X	X	X	X	X	X	X
(Marcum & Weatherill, 1995)	X	X	X	X	X		X	X	X	X
(Nowotny, 1997)	X	X	X	X	X	X	X	X	X	X
(Miranda & Martha, 2014)	X		X	X	X	X	X	X	X	X
(Lee, Pennington, & Shaw, 1984)		X	X	X	X	X	X		X	X
(Peraire, <i>et al.</i> , 1987)	X	X	X	X	X	X	X	X	X	X
(Phillips, Mitchiner, Blacker, & Lin, 1988)	X	X	X	X	X		X	X	X	X
(Pirzadeh, 1993)	X	X	X	X	X	X	X	X	X	X
(Rebay, 1993)	X	X	X	X	X	X	X	X	X	X
(Ruppert, 1992)	X	X	X	X	X	X	X			
(Sadek, 1980)	X	X		X	X	X	X	X		X
(Schöberl, 1997)	X	X		X	X	X	X	X		X
(Sheffer & Üngör, 2001)	X	X	X	X	X		X	X	X	X
(Shewchuk, 2000)	X	X	X	X	X		X	X	X	X
(Shewchuk, 1996)	X	X	X	X	X		X	X	X	X
(Takayama, Panozzo, Sorkine-Hornung, & Sorkine-Hornung, 2013)	X	X	X	X	X		X	X	X	X
(Talbert & Parkinson, 1990)	X	X	X	X	X	X	X	X	X	X
(Tam & Armstrong, 1991)	X	X	X	X	X	X	X	X	X	X
(Tracy, 1977)	X	X	X	X	X		X	X	X	X
(Unruh & Anderson, 1992)	X	X	X	X	X		X	X	X	X
(Wördenweber, 1983)	X	X	X	X	X		X	X	X	X
(Atalay, Xu, & Ramaswami, 2009)		X	X	X	X	X	X			X
(Yerry & Shephard, 1984)		X	X	X	X	X	X			X
(Zhu, Zienkiewicz, Hinton, & Wu, 1991)	X	X	X	X	X	X	X	X	X	X

Trabalhos de revisão foram inicialmente analisados e serviram como guia para o encontro de artigos com diferentes versões de métodos, sendo estes submetidos a uma análise mais detalhada. É interessante mencionar que alguns artigos não passaram para a segunda etapa por apresentarem particularidades que resultam no descumprimento de alguns critérios. Bykat (1976), por exemplo, apresenta um método de subdivisão do domínio, mas o procedimento proposto gera focos de elementos distorcidos em regiões que não deveriam apresentar esta ocorrência, como pode ser observado no artigo. No caso dos artigos de Cavendish (1974) e de Cavendish e Wixom (1975), a apresentação dos métodos é feita com pouca profundidade, levando à conclusão do não cumprimento de vários critérios.

Há outros casos em que os artigos explicam os métodos superficialmente como Johnen, Earnst e Geuzaine (2014), Haber, *et al.* (1978), Jones (1975), entre outros, e isso pode ser verificado nas versões de métodos que não cumpriram o quarto critério genérico na Tabela 6.1. Assim, após esta análise preliminar, foram selecionados 10 artigos que atenderam aos primeiros critérios, sendo eles listados na Tabela 6.2.

Tabela 6.2 - Lista de métodos pré-selecionados.

Tipo de Elem.	Método	Artigo
Elemento Triangular	Delaunay	(REBAY, 1993)
	Frente de Avanço	(LO, 1985)
	Frente de Avanço	(PERAIRE, <i>et al.</i> , 1987)
	Frente de Avanço	(PIRZADEH, 1993)
Elemento Retangular	Subdivisão do Domínio	(TALBERT; PARKINSON, 1990)
	Eixo Médio	(TAM; ARMSTRONG, 1991)
	Pavimentação	(BLACKER; STEPHENSON, 1991)
	Frente de Avanço	(ZHU <i>et al.</i> , 1991)
	Subdivisão do Domínio	(CHOI, 2011)
	Subdivisão do Domínio	(JOE, 1995)
	Frente de Avanço	(MA; ZHAO, 2014)
	Subdivisão do Domínio	(NOWOTTNY, 1997)

6.2 Definição dos Novos Critérios

Tendo a lista dos métodos selecionados para esta segunda etapa de análise, é desenvolvida uma nova filtragem. Estas versões de métodos são avaliadas por notas de acordo com outros seis critérios, agora em uma nova fase mais refinada de avaliação. Estes critérios são definidos como:

A. Versatilidade possível na aplicação do método em geometrias variadas.

O método apresentado deve englobar as geometrias bidimensionais mais genéricas possíveis. Exemplos contidos no artigo auxiliam na determinação da nota.

B. Figuras com geometrias regulares resultam em malhas próximas da estruturada.

Este critério é relativo à proximidade das malhas geradas às estruturadas quando as geometrias são formadoras de figuras regulares (por exemplo coroas, círculos, retângulos).

C. O artigo é claro e bem explicado.

O critério propõe avaliar o nível de clareza na apresentação do método pelo artigo. Verificam-se as explicações de todas as etapas de construção da malha. Um artigo mal explicado pode acarretar em menores notas nos outros critérios em decorrência das dúvidas geradas.

D. A codificação do algoritmo apresentado se mostra de simples implementação computacional.

Deve-se avaliar o nível de complexidade no desenvolvimento de um programa que incorpore a versão do método.

E. A versão do método controla a geração de novos nós a fim de priorizar a configuração com os elementos de melhor qualidade automaticamente.

Neste critério, verifica-se o grau de tratamento dado aos nós internos gerados a fim de se criar elementos de maior qualidade. Considera-se que o tratamento da qualidade dos elementos no próprio método é mais significativo do que apenas ajustar a posição dos nós após os novos elementos serem conectados à malha.

F. O método tende a gerar melhores resultados numéricos.

Este critério se relaciona com o anterior, pois malhas de maior qualidade geram melhores resultados na simulação pelo método dos elementos finitos (MEF). Outro fator importante nos resultados é o ganho de precisão, marginal ou expressivo dependendo do caso, do cálculo aplicando o Método dos Elementos Finitos (MEF) aos elementos quadriláteros em comparação aos triangulares. Assim, este critério adicionará a nota 0 ou 1 ao total de notas para o método, com 0 referente a elementos triangulares e 1 referente a elementos quadriláteros. É importante ressaltar que grande parte dos algoritmos de geração de malhas, em especial as triangulares, adota procedimentos menos rigorosos de controle

da qualidade dos elementos, pois tem a abordagem focada na construção de modelos de computação gráfica. No caso do campo da computação gráfica, elementos distorcidos, em certo grau, afetam menos os resultados desejados do que nas aplicações do MEF.

Não foi considerada como um critério a explicação da etapa de discretização das curvas das fronteiras das figuras. Embora este seja um procedimento imprescindível no algoritmo para geração de malha, nenhum dos artigos pré-selecionados se atenta neste assunto, com exceção de Talbert e Parkinson (1990) que menciona tangencialmente o tratamento de curvas de Bézier como entrada geométrica para o seu método. Os novos critérios foram formulados para promover os trabalhos que são mais claros e as metodologias com aplicação mais simples, o que condicionará a uma implementação computacional mais direta com a linguagem de programação escolhida (definida na Seção 8.1). As notas distribuídas para cada critério seguem a seguinte ordem crescente:

- 0 – Não atende;
- 1 – Atende tangencialmente ao critério;
- 2 – Atende de maneira fraca, exigindo grandes adaptações;
- 3 – Atende exigindo pequenas adaptações;
- 4 – Atende satisfatoriamente;

6.3 Discussão das Versões dos Métodos

Seguiu-se esta segunda etapa com a avaliação dos algoritmos a partir dos novos critérios. Conforme mostrado na Tabela 6.3, vários artigos tiveram a apresentação dos respectivos métodos de maneira superficial e não satisfatória. É notável como a descrição dos métodos, na maioria dos trabalhos, omite a explicação de algumas etapas essenciais dos algoritmos. Isso ocorre quando se indica algum processo contido em outra referência ou por simplesmente ocultar um detalhe da etapa. O trabalho de Talbert e Parkinson (1990) explica superficialmente a etapa da visibilidade dos nós, que é essencial para a subdivisão do domínio em regiões convexas, levando a diferentes interpretações e dando razão para uma nota mediana no critério C. Tam e Armstrong (1991) também citam, de maneira superficial, a etapa de subdivisão do domínio em polígonos, etapa essencial para a reprodução do algoritmo proposto. O texto de Zhu, *et al.* (1991) propõe a criação de quadriláteros a partir da frente de avanço de triângulos, baseando-se no método de Peraire. *et al.* (1987). No entanto, Zhu, *et al.* detalham

pouco o processo de desenvolvimento da malha de fundo, elemento essencial para o funcionamento do algoritmo, pois é partir dela que se definem alguns parâmetros de medidas utilizadas ao longo do processo de criação da malha final. Também é pouco explicado o funcionamento da etapa de junção dos triângulos em quadriláteros. O problema de criação da malha de fundo é apresentado por Peraire, *et al.* (1987), mas é sugerida a intervenção do usuário nesta etapa, tornando sua versão do método de frente de avanço pouco automática. Seria necessário desenvolver alguma alteração no método que automatize a criação da malha de fundo.

Os artigos que apresentam versões do método de subdivisão de domínio em regiões convexas possuem algumas diferenças significativas, seja na explicação como na implementação do algoritmo. Na versão apresentada por Choi (2011) não é explicada a maneira com que são criadas as linhas de corte para gerar as sub-regiões. Também não é explicada uma metodologia genérica para lidar com as fronteiras dos vazios e furos no processo de divisão do domínio. O exemplo de geometria desenvolvido no artigo também é bastante simplificado, conforme mostra a Figura 6.1.

Figura 6.1 - Exemplo trabalhado no artigo de Choi.

Fonte: (Choi, 2011).

Já Talbert e Parkinson (1990) desenvolvem com mais detalhes o método de subdivisão do domínio. No entanto, também apresentam um obstáculo para o entendimento pleno do método. A etapa de verificação da visibilidade dos nós fica pouco clara, mas é imprescindível

ao algoritmo. Na versão de Joe (1995) são apresentados alguns exemplos genéricos satisfatórios, mas o algoritmo não é explicado detalhadamente. Nesse artigo são mostradas várias linhas de pseudocódigo da rotina de implementação de seu algoritmo, entretanto várias funções e variáveis não tem suas características definidas, dificultando o entendimento do método e tornando sua interpretação um processo laborioso. Estas barreiras para uma melhor interpretação do método promoveram as baixas notas nos critérios. São mostrados exemplos de aplicação no algoritmo de Joe (1995), mas figuras mais genéricas do artigo apresentam elementos bastante distorcidos conforme o exemplo da Figura 6.2.

Figura 6.2 - Exemplo de malha bastante genérica.

Fonte: (Joe, 1995).

Nowotny (1997) apresenta sua versão do método de subdivisão sem detalhar algumas etapas do algoritmo. São apresentados alguns exemplos que sugerem a flexibilidade do método em atender geometrias genéricas e em gerar malhas estruturadas no caso de figuras circulares e retangulares. É importante ressaltar que o método de subdivisão apresenta limitações quanto ao

ajuste da posição dos nós criados, resultando em notas baixas no critério E. O fato de se dividir sucessivamente as sub-regiões com linhas de corte fixadas e se criar apenas nelas os novos nós, enrijece a disposição dos nós e o tamanho dos elementos.

Outro fator de grande relevância na escolha de um método para resolver o problema proposto no escopo deste trabalho é sua aplicabilidade em geometrias genéricas, principalmente exemplos de geometrias próximas das conferidas nas seções de cabos umbilicais e tubos flexíveis, como os mostrados na introdução deste trabalho. Os exemplos ideais para representar geometrias satisfatórias são figuras com formas curvilíneas, com vazios e furos internos, com fronteiras também curvilíneas, formando figuras com regiões com convexidade e outras com concavidade. Rebay (1993) apresenta 2 métodos geradores de triângulos com diferenças na abordagem e na velocidade. O autor garante que a aplicação dos dois métodos é válida para geometrias genéricas, no entanto, os exemplos apresentados no artigo são simplificados, com as fronteiras externas se limitando a curvas elípticas e vazios internos circulares. A Figura 6.3 mostra um exemplo de caso.

Figura 6.3 - Caso com pouca complexidade para geração de malha.

Fonte: (Rebay, 1993).

Já Lo (1985) apresenta um método com exemplos satisfatoriamente genéricos e detalha a maioria das etapas. Entretanto, a abordagem, não tão genérica, propõe criar uma grade de nós sobreposta à geometria, excluindo os pontos que são criados fora do domínio, mas sem explicar a maneira de o fazer. O autor sugere modificações na abordagem da sua versão do método de frente de avanço dependendo da geometria de entrada, exigindo ao usuário adequar a grade de nós gerada no interior do domínio às curvas da fronteira, o que torna o método não automático. O artigo de Peraire, *et al.* (1987) não apresenta geometrias genéricas satisfatórias e seu algoritmo é explicado com lacunas. É citada a malha de fundo (*background mesh*) que é utilizada como referência de parâmetros durante toda a criação da malha final, mas não é explicado como é possível a gerar de maneira automática. Zhu, *et al.* (1991) propõem utilizar os triângulos criados pela frente de avanço de Peraire, *et al.*, unindo-os em quadriláteros. O ponto de partida do algoritmo torna o processo não automático, pois depende da malha triangular de fundo, que também não é explicada detalhadamente. A versão do método de frente de avanço proposto por Ma e Zhao (2014) possui uma particularidade: os vazios internos são conectados à fronteira externa a fim de se criar apenas um loop em toda a geometria. Esta fronteira é projetada para o interior do domínio, criando elementos novos. O autor, no entanto, não detalha os próximos passos no tratamento de intersecção de elementos e fechamento de fronteiras, etapas essenciais para garantir elementos de melhor qualidade. Devido a isso, este artigo obteve nota mediana no critério C.

Tam e Armstrong (1991) propõem um método gerador de malha de quadriláteros que se baseia na definição dos eixos médios da figura por meio de uma triangulação de Delaunay inicial bem refinada. Mediante o conceito de que estes eixos médios são passantes pelos pontos centrais dos círculos que circunscrevem os triângulos da triangulação de Delaunay, o algoritmo propõe a construção e o armazenamento destes eixos de forma aproximada, tarefas visualmente simples ao olho humano, mas não triviais para serem armazenadas e processadas por computador. O artigo não desenvolve claramente a etapa de adição das curvas de corte para a divisão do domínio em sub-regiões. Também não são explicados os passos de delimitação das sub-regiões e nem o procedimento de adição de novos elementos, discretizando estas sub-regiões.

Na Tabela 6.3 listaram-se os artigos pré-selecionados e suas novas avaliações. Ressaltou-se, na segunda coluna, o tipo de método apresentado em cada artigo, de modo que artigos com o mesmo método significam apresentações diferentes ou versões diferentes do algoritmo, seja em sua completude ou em alguma de suas etapas de execução.

Tabela 6.3 - Avaliação final dos métodos pré-selecionados.

Tipo de Elemento	Método	Artigo	A	B	C	D	E	F	Total
Elemento Triangular	Delaunay	(REBAY, 1993)	3	1	2	4	1	0	11
	Frente de Avanço	(LO, 1985)	1	3	3	3	3	0	13
	Frente de Avanço	(PERAIRE, <i>et al.</i> , 1987)	3	2	2	3	3	0	13
	Frente de Avanço	(PIRZADEH, 1993)	3	2	2	2	3	0	12
Elemento Retangular	Frente de Avanço	(ZHU; <i>et al.</i> , 1991)	3	3	1	2	2	1	12
	Frente de Avanço	(MA; ZHAO, 2014)	3	4	2	3	3	1	16
	Eixo Médio	(TAM; ARMSTRONG, 1991)	4	3	2	1	3	1	14
	Pavimentação	(BLACKER; STEPHENSON, 1991)	4	4	3	3	4	1	19
	Subdivisão do Domínio	(TALBERT; PARKINSON, 1990)	3	3	2	2	2	1	13
	Subdivisão do Domínio	(CHOI, 2011)	3	2	2	2	2	1	12
	Subdivisão do Domínio	(JOE, 1995)	3	2	2	2	2	1	12
	Subdivisão do Domínio	(NOWOTTNY, 1997)	4	4	2	2	2	1	15

6.4 Seleção do Método

De todos os métodos analisados qualitativamente, o de pavimentação se sobressaiu em perspectiva de eficiência e no atendimento de geometrias genéricas semelhantes às propostas no início deste texto. Outro fator importante para a escolha do método de pavimentação apresentado por Blacker e Stephenson é a maior quantidade de detalhes na descrição das etapas do algoritmo com relação aos outros trabalhos, o que garante uma melhor base para sua reprodução em uma rotina própria. Não é apenas este tipo de algoritmo que tem seu uso consolidado atualmente, pois reconhecidos programas comerciais de simulação por elementos finitos fazem uso deste e de outros métodos em seus próprios geradores de malhas. As ferramentas *ANSYS™* e *ABAQUS™* implementam o método da frente de avanço, enquanto a aplicação *PATRAN* faz uso do método de pavimentação e de eixo médio, sendo todos os pacotes de software referências no mercado de ferramentas *CAE (Computer Aided Engineering)*. O método de frente de avanço (*advancing front*), no caso de uma malha triangular, é apresentado por Peraire, *et al.*, Pirzadeh e Lo. Mesmo se sabendo que é um método que vem sendo desenvolvido desde a década de 80 por estes e outros pesquisadores, estes artigos encontrados

na pesquisa bibliográfica apresentaram o algoritmo de maneira pouco clara, se tornando necessária uma busca bem mais direcionada na literatura para encontrar mais estudos sobre este método. O algoritmo utilizando a técnica de eixo médio também omitiu detalhes de várias etapas, além de se mostrar de complexa programação. O algoritmo de pavimentação (*Paving*), apresentado por Blacker e Stephenson (1991), é uma modificação do algoritmo de frente de avanço com adaptações para a geração de malhas de quadriláteros. O artigo que explica o método da pavimentação apresenta as etapas do algoritmo mais claramente, bem como apresenta exemplos de geometrias satisfatoriamente genéricas, dando motivo para notas altas nos critérios. A geração de malhas de quadriláteros através deste algoritmo produz elementos de boa qualidade tanto em geometrias genéricas como em geometrias regulares. Em geometrias regulares como uma coroa ou um retângulo, o método de pavimentação tende a criar uma malha estruturada, o que automatiza o programa de teste aqui proposto com a aplicação de apenas um método a todas as geometrias presentes dos componentes da seção transversal de um cabo umbilical ou tubo flexível.

Tendo o método selecionado, um estudo direcionado à estruturação de um programa computacional foi executado. Todas as etapas foram minuciosamente estudadas visando sua implementação. Na Seção 7 deste trabalho, foram explicados o método e as considerações feitas visando a sua implementação em um programa de computador que recebe como entrada as fronteiras das geometrias e os tamanhos de elementos médios e gera uma malha de quadriláteros a partir das fronteiras de entrada. Na Seção 9 são apresentados os resultados obtidos.

7 Método da Pavimentação

Conforme já citado na Seção 5, o método de pavimentação pode ser aplicado tanto em figuras regulares quanto em figuras genéricas, mediante a definição de suas fronteiras internas, que determinam furos e vazios, e externa, que determina o contorno mais externo da figura. Seu funcionamento é baseado na projeção consecutiva de fileiras de novos retângulos criados na direção do interior da figura, sendo tratadas, a cada linha de novos retângulos gerada, as eventuais intersecções entre apenas retângulos e entre retângulos e fronteiras. O tamanho dos novos elementos criados nas fileiras é determinado pelas arestas da fronteira que os originaram. Estas arestas estão contidas nas fronteiras de pavimentação, que são as fronteiras a partir das quais o algoritmo gera as novas linhas de retângulos. A primeira linha de novos elementos criada é construída a partir de uma fronteira fixa, interna ou externa, que é classificada momentaneamente como uma fronteira de pavimentação. A Figura 7.1 mostra um modelo esquemático representando a criação desta nova linha de elementos. No item 7.3.1 deste capítulo é apresentada a maneira como são dimensionados os vetores de projeção dos nós das fronteiras para a criação dos novos elementos.

Figura 7.1 - Esquema da primeira linha de novos elementos gera.

Fonte: próprio autor do texto.

Na Figura 7.1, após a criação da linha de novos elementos, a fronteira interna se torna a fronteira de pavimentação vigente, e a fronteira externa retorna ao seu status de fronteira fixa. Conforme será verificado mais adiante no texto, as fronteiras de pavimentação são geradas ao longo do avanço do processamento do método, ou seja, conforme são criados os novos elementos e as intersecções entre arestas são tratadas, novas fronteiras de pavimentação são formadas no interior da figura. O processo tende, assim, a reduzir para zero a quantidade de fronteiras de pavimentação. Blacker e Stephenson (1991) definem que, no momento que uma

fronteira de pavimentação atinge o número de 6 arestas ou menos, ela deve ser tratada de modo específico para cada quantidade de arestas. No caso de restarem seis arestas, o método trata a configuração seguindo alguns padrões definidos pelos autores. Eles alegam que acionando o fechamento da FP a partir de seis nós, obtêm-se malhas de melhor qualidade. Quando restam cinco arestas, se faz necessário criar uma aresta nova a partir da sua maior, criando uma faixa de novos elementos na malha. Caso restem quatro arestas, um novo elemento é gerado. Se restarem três, ocorreu um mal funcionamento do método e talvez algum erro tenha ocorrido. Por fim, se duas arestas sobrarem, estas devem ser sobrepostas. São definidos então dois tipos de fronteiras:

- Fronteiras Fixas (FF): São os *loops* de entrada. Seus segmentos e nós devem ser fixos, não podendo ser alterados, exceto no caso do fechamento de fronteiras interna de cinco nós.
- Fronteira de Pavimentação (FP): São as fronteiras base para a criação dos elementos no interior da figura.

As FFs podem assumir o status de fronteiras de pavimentação, momentaneamente, quando será criada a nova linha de elementos a partir delas. A criação de novos elementos segue uma sequência específica. Inicialmente cria-se a linha de novos elementos a partir da FF externa. Em seguida, caso existam FFs internas, deve-se criar as primeiras linhas de novos elementos a partir de cada FF interna, sucessivamente, até restarem apenas FP vigentes.

Após a criação de cada nova linha, inicia-se uma etapa de verificação de intersecção entre arestas. Caso ocorram intersecções, cada caso é tratado dentro dos padrões definidos no método. No item 7.3.2 são explicados os procedimentos para o tratamento das intersecções. Outra ocorrência que pode ocasionar a edição de elementos da malha é a aproximação excessiva entre arestas de fronteiras de pavimentação. Caso a distância entre as arestas adquira um valor crítico, elas são unidas, resultando em reduções no número de arestas na fronteira de pavimentação, bem como a possibilidade de se criar outras FPs. No item 7.3.3 é explicado como são definidos os critérios de distância entre arestas e o modo como é tratada esta ocorrência, chamada de costura, ou *seam*.

Após as verificações de intersecção e costura, deve-se ajustar a malha para a preparar para a inserção de uma nova linha de retângulos. Esta etapa de ajuste contempla a inserção de cunhas (*wedges*) e dobras (*tucks*), além da aplicação de uma suavização da malha. Cunhas são elementos extras adicionados entre dois quadriláteros de uma linha de novos elementos, em regiões convexas da linha, a fim de diminuir a taxa de crescimento dos elementos criados nesta região. No item 7.3.5 é explicado o método para geração de cunhas. As dobras são utilizadas

para o tratamento oposto, reduzir a taxa de decrescimento das arestas na região: são selecionados três quadriláteros em sequência e o central é colapsado, a fim de aumentar o comprimento das arestas dos quadriláteros vizinhos. No item 7.3.4 é explicado o procedimento para criação das dobras. O outro procedimento imprescindível nesta etapa de ajuste é a suavização. Conforme são realizadas modificações nos elementos da malha, se torna necessário ajustar a posição dos nós dos vizinhos para que eles não sejam excessivamente distorcidos após as edições dos novos elementos.

O método de pavimentação funciona, portanto, como um loop iterativo com a criação de novas linhas de quadriláteros e, ao final de cada iteração, a fronteira de pavimentação obtida é submetida a uma verificação da quantidade de elementos. Blacker e Stephenson (1991) resumem o algoritmo como um pseudocódigo mostrado na Figura 7.2.

Figura 7.2 - Pseudocódigo do algoritmo de Pavimentação.

```

repeat
  perform ROW CHOICE
  repeat
    ADD ROW portion
    SMOOTH row portion
    SEAM boundary
    if INTERSECTION occurs then
      connect overlaps
      SEAM boundary
    end if
  until complete row is added
  perform ROW ADJUSTMENT
  if INTERSECTION occurs then
    connect overlaps
    SEAM boundary
  end if
until CLOSURE CHECK is positive
CLEAN-UP MESH

```

Fonte: (Blacker & Stephenson, 1991)

As etapas listadas no pseudocódigo são explicadas abaixo:

- **Escolha da linha de elementos (*Row Choice*):** Definem-se quais serão os pontos selecionados e em qual fronteira de pavimentação eles estão. Busca-se um nó da FP que forme um ângulo mais próximo de 90°, e todos os outros nós da FP são selecionados sequencialmente, até ser encontrado outro nó, ou o próprio nó inicial, com arestas formando um ângulo próximo de 90°. A partir

destes pontos é construída a linha de novos elementos. O item 7.3.1 explica com mais detalhes esta etapa.

- **Verificação de fechamento (*Closure Check*):** Ao final do fechamento de toda linha de novos elementos e, por consequência, de toda nova FP, é verificado se ela apresenta mais de seis nós. No item 7.3.7 são mostrados os detalhes desta etapa.
- **Geração da linha de elementos (*Add Row*):** Após a seleção dos nós da FP, são criados os novos nós com a projeção deles para o interior da figura. Em seguida são criadas as arestas os conectando aos nós vizinhos. O item 7.3.1 explica mais detalhadamente a classificação dos nós e a forma como são criados os novos nós e as novas arestas que compõem os novos elementos.
- **Suavização (*Smooth*):** Após a adição de cada linha de elementos nova, os nós são ajustados para garantir elementos de melhor qualidade. No item 7.3.6 o algoritmo de suavização é apresentado.
- **Costura (*Seam*):** Arestas de uma mesma ou outra fronteira de pavimentação que estejam próximas até certo valor são unidas. O item 7.3.3 explica o procedimento de costura. Vale salientar que também pode ocorrer a costura em fronteiras de pavimentação formadas após a união de arestas na etapa de intersecção.
- **Ajuste de linhas de elementos (*Row Adjustment*):** As novas linhas de elementos têm seus quadriláteros verificados quanto ao tamanho das arestas. Caso as arestas opostas apresentem uma taxa de crescimento ou decréscimo que passe um valor limite específico, aplicam-se dobras (*tucks*) ou cunhas (*wedges*) na linha de novos elementos, dependendo do caso. No item 7.3.4 é explicado o procedimento de adição de dobras e, no item 7.3.5, explica-se o de adição de cunhas.
- **Intersecções (*Intersection*):** É verificada a existência de interferências entre as arestas de elementos e de fronteiras. O item 7.3.2 explica os procedimentos de detecção e de tratamento das intersecções.
- **Limpeza (*Clean Up*):** É realizada uma verificação geral na malha final, aplicando sucessivas suavizações para melhorar a qualidade da malha.

Esta sequência de procedimentos, no entanto, foi alterada no desenvolvimento do programa. Concluiu-se que a verificação de intersecções deveria ocorrer antes da verificação

de costuras, pois no desenvolvimento do código considerando a explicação de Blacker e Stephenson, notou-se que o módulo de *Seam* seria ativado com os casos de intersecção, já que sua ativação ocorre mediante a verificação da distância entre os pontos médios das arestas próximas na fronteira de pavimentação. Assim, a posição do módulo de *Intersection* e de *Seam* foram trocados no pseudocódigo. Outra alteração feita foi a adição de mais uma etapa de suavização ao final da atualização da nova fronteira de pavimentação. Identificou-se a necessidade de haver outra suavização de fronteira e elementos próximos antes do *loop* retornar ao início e criar a nova linha de elementos, o que resultou em melhores elementos criados a partir desta nova FP. O novo pseudocódigo é apresentado na Figura 7.3.

Figura 7.3 - Pseudocódigo representando a rotina modificada.

```

repeat
  perform ROW CHOICE
  repeat
    ADD ROW portion
    SMOOTH row portion
    if INTERSECTION occurs then
      connect overlaps
      SEAM boundary
    end if
    SEAM boundary
  until complete row is added
  perform ROW ADJUSTMENT
  if INTERSECTION occurs then
    connect overlaps
    SEAM boundary
  end if
  SMOOTH row portion and other elements
until CLOSURE CHECK is positive
CLEAN-UP MESH

```

Fonte: próprio autor do texto.

No decorrer do projeto do programa, notou-se que seria mais prático realizar as verificações de intersecção e de costura depois de criada a nova linha de elementos, pois manipular a malha global a cada novo elemento criado exigiria a alteração do estado de todas as estruturas de dados que formam o módulo de edição de malha para cada elemento. Estruturas como a fronteira de pavimentação e a linha de novos elementos deveriam ser atualizadas em uma frequência maior por iteração do *loop*, exigindo mais tempo de processamento. No código desenvolvido, tentou-se isolar ao máximo as modificações da malha restringindo as

verificações de intersecções e costuras à linha de novos elementos gerados por iteração do *loop*, já que a estrutura do código favoreceu esta abordagem.

É importante ressaltar que a nova linha de elementos criada será adicionada na malha global apenas após todas as verificações de *Intersection* e de *Seam* serem efetuadas. Assim, adicionam-se os novos elementos à malha e, caso a malha não feche, forma-se a nova fronteira de pavimentação. Em seguida, retorna-se ao início do *loop* com a busca do novo nó da fronteira que será selecionado para a criação da nova linha de elementos. A Figura 7.4 mostra a rotina geral em pseudocódigo.

Figura 7.4 - Pseudo-código da versão final do algoritmo de pavimentação desenvolvido.

```

repeat
  perform ROW CHOICE
  repeat
    CREATE ROW portion
  until complete row is created
  SMOOTH row portion
  IF INTERSECTION occurs then
    connect overlaps
    SEAM boundary
  end if
  IF SEAM occurs then
    connect closed edges
    SEAM boundary
  end if
  perform ROW ADJUSTMENT
  IF INTERSECTION occurs then
    connect overlaps
    SEAM boundary
  end if
  SMOOTH row portion and other elements
  ADD ROW to the MESH
until CLOSURE CHECK is positive
CLEAN-UP MESH

```

Fonte: próprio autor do texto.

7.1 Estrutura de Dados

A estrutura do programa foi criada com o foco nas funções de manipulação e de criação de entidades básicas como nós (*Node*), arestas (*Edge*) e elementos (*Element*). Outras entidades como a linha de elementos (*RowElement*), fronteiras (*Boundary*) foram criados para organizar as etapas do método. Estas entidades foram traduzidas no formato de classes, sendo suas propriedades representadas na Figura 7.5 abaixo.

Figura 7.5 - Diagrama de Classes representando as relações entre as principais entidades do programa.

Fonte: próprio autor do texto.

A classe *Node* contém as coordenadas cartesianas X e Y do nó, um número inteiro de identificação *ID*, as listas de identificadores das arestas e dos elementos que contém este nó, uma propriedade *NodeType* o identificando baseado no seu ângulo interno de fronteira e, por fim, apresenta também um método para verificação de igualdade com outro nó. O objeto *Node* possui uma lista de referências de outros nós que se conectam a ele. No caso da classe *Edge*, é importante ressaltar que todas as arestas representadas possuem direção e sentido definidos e,

por isso, foi criado o método *invertDirection()* para a inversão do seu sentido. A classe *Boundary* representa as fronteiras, seja de pavimentação (temporárias) quanto fixas, e armazena os nós, arestas e o sentido em que estão definidas essas listas de entidades pela propriedade *isClockWise*, o que caracteriza também se a fronteira é derivada de um *loop* interno ou externo. A classe *RowElement* representa a linha de novos elementos criados a cada passo do Método de Pavimentação, possuindo o *ID* de referência da fronteira de sua base. Aqui é interessante indicar que o objeto *RowElement* tem seus nós, arestas e elementos adicionados à malha geral apenas no fim do processo de criação destes novos elementos. Todas as verificações e manipulações são feitas considerando o *RowElement* como um grupo de entidades à parte e conectado em parte ou em toda a sua fronteira de pavimentação originária. Entende-se por fronteira de pavimentação originária a fronteira que serviu de referência para os vetores de projeção dos nós na criação dos novos elementos. Além disso, a propriedade *isClosed* define se esta linha de novos elementos é fechada, com a linha de elementos se conectando ao final por uma aresta e fechando nela o ciclo, ou aberta, quando os elementos não se encontram em uma aresta do objeto *RowElement*. Estas são as propriedades das entidades base do programa, e todas as manipulações as utilizam e as modificam a fim de criar uma malha com a tendência de ter menos elementos distorcidos.

Outra opção de estrutura de dados para realizar a geração de malhas e todas suas eventuais manipulações é a proposta por Silva Filho, *et al.* (2014). Diz-se que o algoritmo de pavimentação se torna mais facilmente implementável caso seja utilizada a *halfedge*, ou meia-aresta, que é uma aresta composta por informações adicionais como vetores em ambos os sentidos e informações sobre quadriláteros, nós e outras meia-arestas com as quais ela se conecta. Alega-se que a praticidade gerada por seu uso é automatização na detecção de *loops* internos ou externos devidos aos vetores que as meia-arestas armazenam, e que caso seja considerada a sequência dos vetores, são identificados os *loops* internos ou externos. A Figura 7.6 mostra uma representação das meias-arestas e como são conectadas formando os *loops* que tem o sentido da orientação de seus nós dita pelos vetores.

Figura 7.6 - Esquema de trecho de malha com as meia-arestas.

Fonte: próprio autor do texto.

Esta estrutura de dados não foi adotada por não se apresentar muito intuitiva com relação à criação de novas arestas, no tratamento de costuras e de intersecções. A falta de referências e de exemplos com seu uso nos artigos com malhas de quadriláteros leva a dúvidas quanto ao efetivo ganho de implementação. Além disso, a inclusão de meia-arestas na estrutura de dados exigirá maior tempo de implementação.

7.2 Discretização das Fronteiras

Antes de se iniciar a inserção de elementos na figura é necessário discretizar suas bordas a fim de se obter as fronteiras fixas da malha. O modo de o fazer, no entanto, não é comentado por Blacker e Stephenson. A maneira como é feita a discretização das fronteiras, antes de se iniciar a criação da malha, pode influenciar qualitativamente o estado final da malha. Pode-se aumentar a discretização de trechos específicos da fronteira para refinar uma região em comparação com outras regiões da malha e, dependendo de como é feita a entrada de dados no algoritmo, isso pode ser um fator complicador no processamento inicial das fronteiras fixas. Dependendo de como são criadas as arestas, algumas situações singulares são atingidas. No caso de uma coroa, ou qualquer outra figura que contenha apenas uma fronteira interna igual à externa em escala reduzida, se a quantidade de arestas entre as fronteiras interna e externa for destoante, são criados alguns elementos distorcidos para fechar a malha, conforme é exposto na comparação entre malhas na Figura 7.7.

Figura 7.7 - a) Malha de uma coroa com singularidade. b) Malha estruturada de uma coroa

Fonte: próprio autor do texto.

Na malha a) são gerados 2 elementos muito mal condicionados e que são praticamente triângulos. A solução desta ocorrência pode ser realizada por meio de um pós-processamento da malha a fim de corrigir elementos de pior qualidade. Os dois triângulos podem ser unificados em um novo elemento, bem como o elemento losangular, próximo aos triângulos, pode ser colapsado. Conforme já exposto aqui, por se tratar de mudança de foco, foi concluído que a criação de um módulo de pós-processamento da malha será mais promissora com trabalhos futuros direcionados para esta tarefa.

Outro possível problema devido à discretização das fronteiras é a má representação de trechos que precisam de um maior refinamento. Figuras que possuem trechos pequenos, em comparação com as dimensões de outras regiões da figura, devem ter arestas menores para serem bem representados, enquanto os trechos maiores não precisam de arestas tão pequenas. Isso indica um problema caso as fronteiras sejam discretizadas com arestas de comprimentos com valores próximos, exigindo que o tamanho médio de elemento seja suficientemente pequeno em toda a fronteira para proporcionar condições de geração de elementos de boa qualidade nos trechos agudos ou com empensoamentos. Fronteiras discretizadas com tamanhos de arestas inadequados para determinadas regiões da figura podem acarretar em erros no método de pavimentação.

O problema da proximidade ou interferência entre elementos de tamanhos muito diferentes também é relatado por Choi (2011). São citados especificamente os casos de

elementos reduzidos drasticamente, em regiões côncavas, se encontrando com elementos aumentados drasticamente, em regiões convexas. Tal ocorrência pode apresentar dificuldades de tratamento, que foram observadas na implementação do algoritmo de pavimentação para este trabalho. White e Kinney (1997) também indicam este problema de elementos de dimensões muito diferentes se intersectando ou se aproximando, como esquematizado na Figura 7.8. Nela, o problema é explícito porque a fronteira foi discretizada de modo a conduzir ao erro.

Figura 7.8 - Exemplo de singularidade devido ao tamanho de elemento.

Fonte: (White & Kinney, 1997)

Também se explica, no artigo, o problema denominado “*break-out*”, que é a criação de elementos indefinidamente até o término da memória da máquina. Isto ocorre quando a linha de novos elementos não fecha ou não é encontrado o fim da fronteira de pavimentação de origem (FP utilizada como base da qual os nós são referência para a criação desta linha de novos elementos). Este problema foi corrigido neste trabalho com a constante verificação do caso de a linha de novos elementos encontrar um nó de fim de linha ou encontrar o primeiro nó gerado nesta sequência de elementos, fechando a linha. O programa, no entanto, terá problemas para processar o caso de linha de elementos que sofre interferência de elementos em segundo nível, isto é, elementos que estejam um nível para trás da fronteira de pavimentação, conforme ilustra a Figura 7.9. As arestas em negrito estão intersectando arestas pertencentes ao elemento anterior ao da fronteira de pavimentação. A fim de se evitar o problema, os critérios para a verificação e tratamento de intersecções e costuras foram calibrados com sucessivos exemplos.

Figura 7.9 - Esquema de intersecção de segundo nível.

Fonte: próprio autor do texto.

No trabalho de Ma e Zhao (2014) também é citado o problema de tratamento da proximidade entre elementos que possuem dimensões muito diferentes e mostra o exemplo da Figura 7.10. Conforme citado anteriormente, o código desenvolvido neste trabalho trata todas as fronteiras de modo homogêneo, discretizando todas elas em arestas de tamanhos próximos. Essa limitação de discretização de fronteira, entretanto, tem a vantagem de contribuir para evitar o problema da Figura 7.10.

Figura 7.10 - Aproximação de elementos de tamanhos diferentes.

Fonte: (Ma & Zhao, 2014)

7.3 Etapas do Método

O Método de Pavimentação contém uma série de verificações do estado da malha a cada grupo de elementos adicionado, identificando características que exigem a aplicação de fases específicas para cada situação. Nos itens a seguir são explicadas as etapas do método e também são feitas considerações relevantes quanto à implementação do algoritmo.

7.3.1 Adição de nova linha de elementos

Os nós são classificados com o intuito de guiar a criação de elementos, pois cada tipo exige um procedimento diferente de inserção de novos nós e elementos. Definem-se as tolerâncias angulares $\sigma_1, \sigma_2, \dots, \sigma_6$ que expressam pequenos valores de ângulos para diferenciar os estados dos nós. Blacker e Stephenson (1991) não definem valores para estas tolerâncias, cabendo aos leitores do artigo concluírem a melhor maneira de se definir valores para estas variáveis. Os estados de nós são classificados a seguir, seguidos de um esquema os representando na Figura 7.11.

- Fim de linha ou *row end* ($\alpha \leq \frac{\pi}{2} + \sigma_1$): é um nó onde a linha de elementos sendo criados deve terminar.
- Transição fim/lateral de linha ($\frac{\pi}{2} + \sigma_1 < \alpha \leq \pi - \sigma_2$): nó que pode representar um nó de fim de linha ou um nó de lateral de linha.
- Lateral de linha ou *row side* ($\pi - \sigma_2 < \alpha \leq \pi + \sigma_3$): um nó neste estado posiciona-se próximo do alinhamento entre dois elementos, formando 180° entre as arestas da FP.
- Transição lateral/canto de linha ($\pi + \sigma_3 < \alpha \leq \frac{3\pi}{2} - \sigma_4$): nó que pode representar um nó de lateral de linha ou um nó de canto de linha.
- Canto de linha ou *row corner* ($\frac{3\pi}{2} - \sigma_4 < \alpha \leq \frac{3\pi}{2} + \sigma_5$): durante a adição de elementos na linha de pavimentação, a presença deste estado de nó exibe um canto na FP com mais de 180° .
- Transição canto/reverso de linha ($\frac{3\pi}{2} + \sigma_5 < \alpha \leq 2\pi - \sigma_6$): nó que pode representar um nó de canto de linha ou um nó reverso de linha.
- Reverso de linha ou *row reversal* ($\alpha > 2\pi - \sigma_6$): um nó com esse estado pode ser encarado como um canto vivo, sendo o tipo de nó que gera mais elementos na FP.

Figura 7.11 - Esquema representando os diferentes estados dos nós.

Fonte: (Blacker & Stephenson, 1991)

Assim, pode-se formular um gráfico que represente as faixas de ângulos dos estados de nós, como mostra a Figura 7.12. Nela se observa regiões entre os intervalos σ_i e que delimitam os estados de transição dos nós, o permitindo assumir um ou outro estado dependendo da configuração da malha no momento.

Figura 7.12 - Faixas de ângulos para cada estado dos nós

Fonte: próprio autor do texto.

O algoritmo é iniciado criando-se camadas de elementos nas FFs, compostas por elementos adicionados lado a lado até preencher um ciclo ou até encontrar outro nó do tipo *end node*, formando assim uma nova FP. Conforme já citado no início deste capítulo, as FPs são modificadas com a criação de elementos no sentido anti-horário no caso de fronteira externa e, no sentido horário, no caso de eventuais fronteiras internas.

Deve-se definir o nó a partir do qual se iniciará a criação da nova linha de elementos e para isso, verificam-se todos os nós da fronteira a fim de se encontrar um nó do tipo fim de linha. Caso um nó deste tipo não seja achado, o programa seleciona os nós do tipo transição fim/lateral de linha e testa a qualidade do elemento que seria formado com cada nó sendo utilizado como o inicial. O nó selecionado será aquele que apresentar o elemento de maior qualidade. Caso não sejam encontrados nós do tipo transição fim/lateral de linha, o mesmo procedimento é realizado, mas agora testando todos os nós da fronteira independentemente do tipo. Depois de selecionado o nó inicial, a lógica implementada no programa cria a nova linha

de elementos, nó a nó, até encontrar, na fronteira, um nó do tipo fim de linha ou o nó onde se iniciou esta linha de novos elementos. Também é possível encontrar um nó do tipo transição fim/lateral de linha em um dado momento durante a criação dos novos elementos. Neste caso, o algoritmo verifica qual tipo de nó gerará um elemento de menor qualidade, testando o processamento deste nó como o tipo nó de fim de linha e como o tipo *side node*, adotando, por fim, o tipo que não tiver o elemento de menor qualidade.

O procedimento para adição de elementos novos é baseado na projeção dos nós da fronteira de pavimentação base, gerando um ou mais elementos novos para serem adicionados na linha de novos elementos. Cada tipo de nó define um modelo de projeção da fronteira, conforme mostrado abaixo:

Projeção do nó do tipo *side node*: Projeta-se o nó N_i para dentro da figura formada pelas fronteiras conforme mostra a Figura 7.13, considerando as arestas $\overline{N_{i-1}N_i}$ e $\overline{N_iN_{i+1}}$ como referência para o vetor V , criando o novo nó N_j , a aresta $\overline{N_{j-1}N_j}$ e o elemento $N_{j-1}N_jN_iN_{i-1}$. O vetor V é criado como um vetor bissetor do ângulo α e de módulo

$$|V| = \frac{(d_1 + d_2)/2}{\sin(\alpha/2)}$$

onde d_1 é o comprimento da aresta $\overline{N_{i-1}N_i}$ e d_2 é o comprimento da aresta $\overline{N_iN_{i+1}}$.

Figura 7.13 - Modelo da projeção do nó tipo *side node*.

Fonte: (Blacker & Stephenson, 1991)

Projeção do tipo *corner node*: Projeta-se o nó N_i para dentro da figura formada pelas fronteiras, criando os novos nós N_j , N_k e N_l , e formando os elementos $N_{j-1}N_jN_iN_{i-1}$ e $N_jN_kN_lN_i$, conforme mostrado na Figura 7.14. Os vetores V_j , V_k e V_l são posicionados com a origem no nó N_i e formam ângulos $\frac{\alpha}{3}$, $\frac{\alpha}{2}$ e $\frac{2\alpha}{3}$ respectivamente, no sentido horário com relação ao segmento de reta $\overline{N_lN_{i-1}}$. Seus módulos são dados por:

$$|V_j| = \frac{(d_1 + d_2)/2}{\sin(\alpha/3)}$$

$$|V_k| = \sqrt{2}|V_j|$$

$$|V_l| = |V_j|$$

onde d_1 é o comprimento da aresta $\overline{N_{i-1}N_i}$ e d_2 é o comprimento da aresta $\overline{N_iN_{i+1}}$.

Figura 7.14 - Modelo de projeção do nó tipo Corner Node.

Fonte: (Blacker & Stephenson, 1991)

Projeção do tipo *reversal node*: Projeta-se o nó N_i para dentro da figura formada pelas fronteiras, criando cinco novos nós N_j , N_k , N_l , N_m e N_n e formando os elementos $N_{j-1}N_jN_iN_{i-1}$ e $N_jN_kN_lN_i$, conforme mostrado na Figura 7.15. Os vetores V_j , V_k , V_l , V_m e V_n são situados com a origem no nó N_i e são dispostos no sentido horário com ângulos $\frac{\alpha}{4}$, $\frac{3\alpha}{8}$, $\frac{\alpha}{2}$, $\frac{5\alpha}{8}$ e $\frac{3\alpha}{4}$ respectivamente. Seus módulos são dados por:

$$|V_j| = \frac{(d_1 + d_2)/2}{\sin(\alpha/4)}$$

$$|V_k| = \sqrt{2}|V_j|$$

$$|V_l| = |V_j|$$

$$|V_m| = |V_k|$$

$$|V_n| = |V_j|$$

onde d_1 é o comprimento da aresta $\overline{N_{i-1}N_i}$ e d_2 é o comprimento da aresta $\overline{N_iN_{i+1}}$.

Figura 7.15 - Modelo de projeção do nó tipo Reversal Node.

Fonte: (Blacker & Stephenson, 1991).

Com a linha de novos elementos criada, são aplicadas sequencialmente as verificações de edições necessárias, tanto na linha quanto nas eventuais fronteiras envolvidas. Após todo o tratamento envolvendo a linha de novos elementos, ela é acoplada à malha, formando uma nova fronteira de pavimentação.

7.3.2 Verificação de intersecções (*Intersection Module*)

A metodologia apresentada por Blacker e Stephenson (1991) não é clara na forma de implementação, nem no tratamento das intersecções em nível de estruturas de dados. Não há nem um exemplo simplificado de implementação ou pseudocódigo, deixando a cargo do leitor

a conclusão sobre quais estruturas de dados adotar e como proceder com o tratamento destas intersecções. Devido a isso e seguindo a estrutura de dados proposta neste trabalho, foi criado um módulo de verificação de ocorrência e de tratamento de intersecções. Estas ocorrências foram divididas em dois tipos:

Intersecção elemento-elemento: representa a intersecção entre elementos de uma mesma linha de novos elementos.

Intersecção elemento-fronteira: representa a intersecção entre um elemento da linha de novos elementos e uma fronteira, não necessariamente a mesma fronteira base dos novos elementos.

No artigo base, o tratamento das intersecções é apresentado de maneira bastante genérica, exigindo que o desenvolvedor do programa defina as verificações e as lógicas necessárias para a sua aplicação.

Um tipo de intersecção, que foi tratado separadamente no programa, é aquele que ocorre entre elementos vizinhos em uma fronteira. Entende-se por elementos vizinhos aqueles que possuem ao menos uma aresta em uma fronteira comum e que compartilham pelo menos um nó que pertence a essa mesma fronteira. Um exemplo desta ocorrência é mostrado na Figura 7.16 com a visível interferência entre dois elementos vizinhos.

Figura 7.16 - Exemplo de intersecção entre elementos vizinhos.

Fonte: próprio autor do texto.

As intersecções genéricas, desconsiderando as intersecções entre elementos vizinhos, são tratadas mediante um algoritmo desenvolvido pelo autor deste trabalho. No artigo base do

método de pavimentação, são levantadas algumas ocorrências de intersecções que eventualmente podem surgir durante a criação da malha, mas não é especificado um tratamento mais detalhado destas ocorrências. São indicadas algumas considerações no texto sobre a união das arestas e sobre as novas fronteiras de pavimentação criadas. Cabe aqui realizar algumas observações sobre determinadas considerações:

Limitação da Paridade (*Evenness constraint*): exige-se que as novas fronteiras de pavimentação formadas contenham um número par de nós, o que não ocorre em casos como intersecção elemento-elemento em uma linha de novos elementos fechada. O tratamento desta ocorrência não está claro no texto.

Busca à frente e atrás: este ponto sugere a busca por possíveis uniões à frente e atrás na linha de novos elementos que gerem elementos de melhor qualidade. O programa desenvolvido para este trabalho possui esta verificação.

Diferenças de tamanho: Com relação a arestas de tamanhos diferentes na intersecção, foram tratados os casos especiais de elementos vizinhos: quando são formados ângulos agudos ou quando eles se intersectam, conforme mostrado na Figura 7.17.

Figura 7.17 - Tratamento de arestas com tamanhos diferentes.

Fonte: (Blacker & Stephenson, 1991).

Assim, para a realização deste trabalho, foi criada uma metodologia para a edição da malha, das fronteiras e das linhas de novos elementos envolvidas na intersecção sendo tratada,

com a tentativa de atender ao máximo as considerações expostas no artigo base. O algoritmo para o tratamento de intersecções pode ser representado pelo pseudocódigo na Figura 7.18.

Figura 7.18 - Algoritmo de tratamento de intersecções.

```

repeat
  if INTERSECTION List.size > 0
    measure INTERSECTIONS qualities
    find max quality INTERSECTION
    process INTERSECTION
    edit RowElement
    edit Boundary
    create new entities
    repeat
      if SEAM List.size > 0
        process INTERSECTION
        edit RowElement
        edit Boundary
        create new entities
      end if
    until occurs SEAM == false
  end if
until occurs INTERSECTION == false

```

Fonte: próprio autor do texto.

A etapa de verificação de intersecções exige a verificação de interferência entre elementos e isso pode ser realizado por diferentes abordagens. É possível verificar se um nó de algum dos elementos está contido no interior de outro, ou testando a intersecção entre as arestas destes elementos. O modelo de verificação de intersecção proposto por Blacker e Stephenson (1991) não foi adotado, pois já estava à disposição um algoritmo pronto e testado, oriundo de trabalhos anteriores do pesquisador, e que promove o mesmo resultado. O algoritmo calcula a solução de um sistema linear gerado pela tentativa de detecção do ponto de intersecção entre duas retas parametrizadas, representando duas arestas de elementos diferentes; ou uma aresta de um elemento e uma aresta de uma fronteira. Assim, para cada tipo de intersecção, foi estabelecido um procedimento específico para tratar as entidades envolvidas e que são explicados na sequência.

- **Intersecção elemento-elemento**

O tratamento deste tipo de intersecção é iniciado com o armazenamento das ocorrências de intersecções entre os elementos da linha de novos elementos. Em seguida, verifica-se a qualidade mínima dos novos elementos formados, simulando a situação de que cada ocorrência seja tratada independentemente. É escolhida a interferência que resultar em elementos com maior qualidade. No caso do surgimento de elementos de canto (*corner elements*), que possuem duas arestas não compartilhadas com outros elementos, são testadas as combinações de conexões possíveis entre suas arestas livres a fim de se terminar a combinação que gerará elementos de maior qualidade.

Escolhidos os dois elementos da linha de novos elementos que serão unidos, inicia-se o processo de edição da malha. Devido à estrutura de dados adotada, a edição da malha deverá quebrar a linha de novos elementos e a fronteira de pavimentação base. São determinadas as arestas livres que serão coladas como a aresta mestra e a aresta escrava, de modo que os nós da aresta mestra são removidos e a aresta escrava assume a conexão de todas as outras arestas que se conectavam à aresta mestra. Este procedimento pode ser observado no modelo esquemático da Figura 7.19 da configuração a) para a configuração b), onde também é enfatizada a criação de novas linhas de elementos e de novas fronteiras base. A união dos dois elementos resulta na criação de duas fronteiras de pavimentação base e, no mínimo, duas novas linhas de novos elementos.

Figura 7.19 - Modelo esquemático das etapas de tratamento da intersecção elemento-elemento.

Fonte: próprio autor do texto.

Depois de realizada esta junção inicial dos elementos, deve-se prosseguir com a junção dos elementos vizinhos, caso seja necessário. A alteração da configuração de b) para a c) no esquema da Figura 7.19 mostra a junção dos elementos vizinhos. O critério para a realização da união foi a formação de um ângulo arbitrário de 50° ou menos, ou a ocorrência de

intersecção. Novamente são definidas as arestas mestra e escrava dos elementos vizinhos, remove-se o nó não compartilhado da aresta mestra e substitui-se a aresta mestra pela aresta escrava, conectando à aresta escrava todas as arestas que estavam conectadas à mestra. Este procedimento é repetido sucessivamente até a formação de elementos vizinhos que passem pelo critério definido. Vale salientar que o processo de união dos nós vizinhos (configuração c) também promove alteração nas fronteiras e nas linhas de elementos, resultando em mudanças nestas entidades a cada união de elementos vizinhos.

As edições de linhas de novos elementos e de fronteiras envolvem a remoção e a criação de nós, arestas e elementos. A fim de se automatizar estas funções, foram desenvolvidos módulos de edição de malha baseados nos Operadores de Euler. Estes módulos ficaram responsáveis por tratar a malha, as fronteiras e as linhas de novos elementos, removendo ou adicionando entidades, tratando também pendências, como nós conectados a apenas uma aresta e arestas conectadas a apenas um nó. Ficaram responsáveis também pelas atualizações de dados sobre vizinhanças e conexões entre as entidades da malha.

- **Intersecção elemento-fronteira**

O tratamento deste tipo intersecção é iniciado com o armazenamento das ocorrências de intersecções entre os elementos da linha de novos elementos e as arestas das fronteiras de pavimentação. Em seguida, verifica-se a qualidade mínima dos novos elementos a serem formados pela união das arestas livres do elemento com a aresta da fronteira, simulando a situação de que cada ocorrência seja tratada independentemente. Assim, é escolhida a interferência que resultar em elementos com maior qualidade. No caso do surgimento de elementos de canto (*corner elements*), que possuem duas arestas não compartilhadas com outros elementos, são testadas as combinações de conexões possíveis entre suas arestas livres a fim de se terminar a combinação que gerará elementos de maior qualidade.

Escolhido o elemento da linha de novos elementos e a aresta da fronteira selecionada que serão unidos, inicia-se o processo de edição da malha. Devido à estrutura de dados adotada, a edição da malha deverá quebrar a linha de novos elementos, a fronteira de pavimentação base e outra fronteira de pavimentação, caso seja envolvida. São determinadas as arestas livres que serão unidas, classificando-as como aresta mestra e aresta escrava, de modo que os nós da aresta mestra são removidos e a aresta escrava assume a conexão de todas as outras arestas que se conectavam à aresta mestra. Este procedimento pode ser observado no modelo esquemático da Figura 7.20, da configuração a) para a configuração b), onde também é enfatizada a criação de novas linhas de elementos e de novas fronteiras base. A união do elemento com a fronteira

resulta na criação de no mínimo uma nova fronteira de pavimentação base e, no mínimo, uma nova linha de novos elementos.

Figura 7.20 - Modelo esquemático das etapas de tratamento da intersecção elemento-fronteira.

Fonte: próprio autor do texto.

Depois de realizada esta junção inicial do elemento com a antiga fronteira de pavimentação, deve-se prosseguir com a junção das arestas vizinhas, se for o caso. A alteração da configuração b) para a c) no esquema da Figura 7.20 mostra a junção dos elementos vizinhos com a fronteira. O critério para a realização união foi a formação de um ângulo arbitrário de 50° ou menos, ou a ocorrência de intersecção. Novamente são definidas as arestas mestra, no elemento selecionado, e escrava, na fronteira alvo. Remove-se o nó não compartilhado da aresta mestra e substitui-se a aresta mestra pela aresta escrava, conectando à aresta escrava todas as arestas que estavam conectadas à mestra. Este procedimento é repetido sucessivamente até a formação de elementos vizinhos que passem pelo critério definido. Vale salientar que o processo de união dos nós vizinhos (configuração c) também promove alteração nas fronteiras e nas linhas de novos elementos, resultando em mudanças nestas entidades a cada união de elementos vizinhos. Assim, também são utilizados os módulos de edição de malha citados no caso do processamento das intersecções elemento-elemento.

Tanto no caso de intersecções elemento-elemento, como no caso elemento-fronteira, o programa desenvolvido não trata de casos específicos:

Elementos de canto: No caso de elementos de canto se intersectando com outros elementos ou fronteiras, a ocorrência não foi tratada, pois os exemplos testados não apresentaram esta ocorrência. O código consegue tratar este problema caso o elemento de canto pertença à fronteira de pavimentação e não à linha de novos elementos. A Figura 7.21 mostra um exemplo da configuração com o elemento de canto indicado.

Figura 7.21 - Exemplo de intersecção de elemento de canto.

Fonte: próprio autor do texto.

Elemento único: No caso de um único elemento criado intersectando outro elemento ou fronteira, e que é caracterizado também como um elemento de canto nos dois sentidos da linha de novos elementos, o programa também não realiza o seu tratamento, pois a forma com que foi desenvolvida a lógica de processamento das intersecções e costuras não gerou esta ocorrência com os exemplos testados. Este caso, no entanto, dificilmente ocorrerá, pois não são verificadas intersecções e costuras a cada elemento criado na linha de novos elementos, que foi uma alteração no método de pavimentação sugerida por White e Kinney (1997). Alega-se de que a verificação de intersecção ou costura, quadrilátero a quadrilátero adicionado à linha de novos elementos, torna o método mais robusto. No método proposto no presente trabalho, é realizado um tratamento geral da lista de novos elementos após ela ser concluída. Como mencionado no caso anterior, o código desenvolvido conseguirá tratar esta ocorrência caso o elemento único em questão não esteja na linha de novos elementos, ou seja, devem haver uma fileira de novos quadriláteros. A Figura 7.22 mostra um exemplo desta configuração que não será tratada devido à não ocorrência na versão do método aqui apresentada.

Figura 7.22 - Exemplo de intersecção de elemento único.

Fonte: próprio autor do texto.

7.3.3 Verificação de costura de elementos próximos (*Seam Module*)

Este módulo foi desenvolvido baseado nos mesmos processos do módulo de intersecções, sendo que a única diferença é a verificação das distâncias das arestas e não de intersecções entre elas. As ocorrências para possibilidade de aplicação de costura (*Seam*) foram divididas nos mesmos dois tipos: elemento-elemento e elemento-fronteira. O critério para determinar a necessidade de se aplicar a costura (*Seam*) não é explicado por Blacker e Stephenson (1991), sendo citado apenas o tratamento no caso de ângulos agudos em fronteiras de pavimentação, que foi abordado separadamente no código desenvolvido. Assim, fica a cargo do desenvolvedor especificar qual o critério para se aplicar a costura. O pseudocódigo representando uma visão geral do algoritmo é mostrado na Figura 7.23.

Figura 7.23 - Pseudocódigo geral do algoritmo do módulo de costura (Seam).

```

repeat
  if SEAM List.size > 0
    measure SEAM qualities
    find max quality SEAM
    process SEAM
    edit RowElement
    edit Boundary
    create new entities
    repeat
      if neighbor SEAM List.size > 0
        process SEAM
        edit RowElement
        edit Boundary
        create new entities
      end if
    until occurs neighbor SEAM == false
  end if
until occurs SEAM == false

```

Fonte: próprio autor do texto.

No trabalho, as arestas presentes em fronteiras de pavimentação e em elementos da linha de novos elementos foram testadas duas a duas, seguindo os seguintes critérios:

O método occurs SEAM do pseudocódigo da Figura 7.23 retorna *true* quando verifica uma ocorrência de arestas próximas por uma determinada distância de ativação. O critério é acionado quando as arestas livres de elementos próximos ou de fronteiras tenham seus pontos médios distantes de um valor menor ou igual a D . Definem-se l_1 e l_2 as arestas mestra e escrava, respectivamente, e l'_1 e l'_2 as arestas opostas às arestas mestra e escrava no elemento específico de cada uma. Define-se D como

$$D = \frac{1}{2} \left(\frac{(l_1 + l'_1)}{2} + \frac{(l_2 + l'_2)}{2} \right) \quad (7.1)$$

Caso a avaliação seja feita entre arestas de elementos de uma mesma linha de pavimentação, ou seja, em uma ocorrência elemento-elemento, o valor D deve ser usado como distância máxima para a ativação da costura. Se a avaliação for feita entre arestas de um elemento da linha de novos elementos e de uma fronteira de pavimentação, uma ocorrência elemento-fronteira, deve-se adotar o valor $0,5D$ como distância máxima para a ativação da costura. Note-se que estas verificações e os parâmetros adotados na ativação do módulo de costura são arbitrários, aproximados empiricamente a partir de testes com exemplos de geometrias típicas de cabos umbilicais e tubos flexíveis.

Uma vez ativado o módulo de costura, um procedimento análogo ao da intersecção é iniciado. É determinada a ocorrência de aproximação entre arestas que gerará elementos de melhor qualidade. Processam-se as entidades envolvidas para a edição da malha ao unir as arestas, seguindo um procedimento análogo ao utilizado na intersecção. A Figura 7.24 e a Figura 7.25 exemplificam esquematicamente as etapas de edição da região em que se aplica a costura para os casos elemento-elemento e elemento-fronteira, respectivamente.

Figura 7.24 - Configurações da aplicação da costura (*Seam*) na ocorrência elemento-elemento

Fonte: próprio autor do texto.

Figura 7.25 - Configurações da aplicação da costura (*Seam*) na ocorrência elemento-fronteira.

Fonte: próprio autor do texto.

As etapas mostradas se assemelham muito às do módulo de intersecção, reforçando que o tratamento é basicamente o mesmo, com exceção dos critérios para sua ativação. Assim, após a primeira costura das arestas da linha de novos elementos com outra aresta de um elemento ou com uma fronteira de pavimentação, as arestas vizinhas têm os ângulos avaliados. Assim, o tratamento das arestas vizinhas segue de maneira idêntica ao do módulo de intersecção.

7.3.4 Inserção de Dobras (*Tuck Module – Row Adjustment*)

Esta etapa é responsável pela detecção da necessidade de se inserir dobras (ou *tucks*) na linha de novos elementos caso haja a redução excessiva do tamanho de suas arestas conforme são criados os novos quadriláteros no interior da figura. A inserção de uma dobra na linha de novos elementos é realizada pelo colapso de uma região com 3 elementos para uma com 2 elementos, conforme exemplifica a Figura 7.26.

Figura 7.26 - Exemplo de aplicação de dobra. Processo esquematizado da esquerda para a direita.

Fonte: próprio autor do texto.

Os critérios para criar uma dobra são apresentados no artigo de Blacker e Stephenson de modo a gerar margem para inconsistências. Determina-se que as arestas de uma fronteira de pavimentação que apresentarem uma taxa de diminuição de comprimento de 0,8 e que formarem ângulos internos menores que 177° em um nó desta mesma FP, devem ativar a formação de dobras. Estes critérios qualificam arestas da fronteira de pavimentação para formarem dobras, entretanto a maneira como foi explicado o processo, leva a algumas questões:

- A variação de 0,8 é referente à aresta oposta do próprio elemento?
- Adotar a razão 0,8 pode ativar a dobra em casos de elementos de boa qualidade, trocando-os por elementos de qualidade inferior.
- É possível gerar uma situação de singularidade em um círculo muito discretizado, com seus nós com ângulos internos maiores que 177° . Nele os

elementos serão criados indefinidamente, cada vez com tamanho mais reduzido, até um limite, no infinito, de tamanho de aresta igual a zero.

Outros problemas encontrados na explicação do artigo são referentes à frequência com que serão aplicadas as dobras. Não é definido um padrão de quantidade de elementos que deve haver em uma porção da linha de novos elementos para aplicação da dobra. Apenas se faz uma indicação de que as dobras devem estar espaçadas de ângulos múltiplos de 90° considerando o ângulo formado entre as dobras com o centro do arco da FP como referência, conforme exemplifica a Figura 7.27.

Figura 7.27 – Exemplo de aplicação de dobras.

Fonte: (Blacker & Stephenson, 1991)

Os exemplos apresentados no artigo se limitam a geometrias com arcos de circunferência bem definidos. No caso do tratamento de geometrias genéricas, pode ser necessária a criação de uma dobra em uma curva que não seja um arco bem definido, e por isso não se pode determinar ângulos múltiplos de $\pi/2$ entre eventuais dobras como o caso dos arcos de circunferência da Figura 7.27.

Considerando estes pontos que impossibilitam a implementação de maneira mais clara desta etapa do método, foi desenvolvido um algoritmo com algumas alterações nos critérios, direcionando as adaptações para as geometrias testadas neste trabalho. Os critérios foram alterados de modo que não são mais verificados os ângulos, mas apenas a alteração do tamanho das arestas. São feitas duas avaliações: a razão de decrescimento entre uma aresta da FP e sua aresta oposta r_e , a razão de decrescimento da aresta da fronteira de pavimentação e o tamanho médio das arestas da fronteira fixa discretizada r_g . Definem-se l como o comprimento da aresta da fronteira da pavimentação, l_{op} o comprimento da aresta oposta a l no mesmo elemento e

l_{ref} o comprimento médio da aresta das fronteiras fixas. Assim, a aresta será dita qualificada quando

$$\begin{cases} r_e = \frac{l}{l_{op}} \leq 0,7 \\ r_g = \frac{l}{l_{ref}} \leq 0,6 \end{cases}$$

Os valores de referência para estas razões foram obtidos arbitrariamente, a partir de ajustes com os exemplos testados. Assim, caso $r_e \leq 0,7$ ou $r_g \leq 0,6$ a aresta obterá o status de qualificada. Esta comparação da aresta da FP com o tamanho médio das arestas da fronteira fixa foi adotada para evitar a não ativação das dobras por conta da eventual ocorrência de baixa taxa de decrescimento entre arestas de um mesmo elemento.

A outra parte do algoritmo é a definição de quais arestas qualificadas serão quebradas para adicionar a dobra. O artigo propõe uma verificação de ângulos entre a eventual posição de dobras, mas não explica claramente qual é o procedimento e nem como tratar casos de linhas de elementos que formem curvas genéricas ou polinomiais, pois não têm um único centro bem definido. O único critério bem expresso sobre inserção de dobras foi a limitação de se inserir no máximo quatro dobras em uma linha de novos elementos fechada ou próxima de se fechar (formando um arco de ângulo próximo a 360°). Devido à ausência de informações mais detalhadas em nível de algoritmo, desenvolveu-se uma metodologia para determinar um ângulo de circunferência formado pela sequência de arestas qualificadas, adaptando a curva formada pelas arestas a um arco de circunferência. Definida a lista de arestas qualificadas consecutivas, são traçadas as medianas da primeira e da última aresta desta lista, determinando o ponto de intersecção. A Figura 7.28 mostra um esquema do procedimento.

Figura 7.28 - Esquema de identificação de arco formado por arestas qualificadas.

Fonte: próprio autor do texto.

As arestas de pontos médios M_1 e M_2 delimitam uma lista de arestas qualificadas, seguindo a orientação da fronteira de pavimentação, indicada pelas setas. A intersecção das medianas determina o ponto Q , e assim calcula-se o ângulo α definido pelos vetores $\overrightarrow{QM_1}$ e $\overrightarrow{QM_2}$, seguindo a orientação da fronteira de pavimentação. Aproxíma-se a razão $\alpha/90^\circ$ para um número inteiro a fim de definir quantas dobras serão adicionadas. Além do ângulo, é verificada a quantidade de arestas qualificadas na lista em questão. São necessárias 4 arestas qualificadas seguidas para cada dobra a ser adicionada. Assim, tenta-se inserir a quantidade de dobras na linha de novos elementos mediante a quantidade de vezes que α é divisível por 90° e mediante a quantidade de arestas existentes na lista. A Figura 7.29 mostra o pseudocódigo do algoritmo para determinar a inserção ou não de dobras.

Figura 7.29 - Pseudocódigo do algoritmo de determinação da inserção de dobras.

```

do FindNextQualifiedEdges
  Determine Q point
  Determine  $\alpha$ 
  Determine integer  $n = \alpha/90^\circ$ 
  Determine integer  $m = \text{qualifiedEdges.size}/4$ 
  case  $n = 0$  or  $n = 1$ 
    if  $m \geq 4$ 
      ApplyTuck
  case  $n = 2$ 
    if  $m \geq 8$ 
      ApplyTuck x 2
    else
      ApplyLessTucks
  case  $n = 3$ 
    if  $m \geq 12$ 
      ApplyTuck x 3
    else
      ApplyLessTucks
  case  $n = 4$ 
    if  $m \geq 16$ 
      ApplyTuck x 4
    else
      ApplyLessTucks
until qualifiedEdgelist.size = 0

```

Fonte: próprio autor do texto.

A adição de uma dobra em uma linha de novos elementos se faz a partir de 4 arestas qualificadas em sequência e o pseudocódigo da Figura 7.29 mostra a quantidade de dobras a ser adicionada dependendo do ângulo formado e da quantidade de arestas qualificadas em sequência.

O procedimento de edição dos elementos para a inserção da dobra inicia-se com a definição dos nós n_1, n_2, n_3 e n_4 da fronteira de pavimentação, que determinam 3 arestas que serão alteradas, bem como os nós p_1, p_2, p_3 e p_4 opostos aos primeiros. Devem-se remover algumas entidades e se conectar as restantes para a formação de novos elementos, conforme esquematizado na Figura 7.30.

Figura 7.30 - Esquema do processo de inserção de dobras.

Fonte: processo de inserção de dobras

Seguindo as etapas da esquerda para a direita na Figura 7.30, removem-se os nós n_2 e n_3 , o que implica na remoção das arestas que se conectavam a eles. Todo procedimento de tratamento da malha editada foi seguido com os operadores de Euler modificados, já citados nas Seções de Intersecção e de Costura anteriormente. Segue-se com a criação das arestas $\overline{p_3n_1}$ e $\overline{n_4n_1}$, formando dois novos elementos. Por fim, uma suavização aplicada localmente reposiciona o nó n_1 , proporcionando uma melhor qualidade aos elementos que se conectam a ele. Cabe ressaltar que, após a adição da dobra, o nó n_2 não apresentará conexão com nenhum novo nó da iteração seguinte.

É importante ressaltar que os critérios adotados no algoritmo desenvolvido neste trabalho foram ajustados de modo empírico, priorizando elementos de melhor qualidade a partir das geometrias utilizadas como exemplos.

7.3.5 Inserção de Cunhas (*Wedge Module – Row Adjustment*)

Esta etapa é semelhante à adição de cunhas. Neste módulo é detectada a necessidade de se inserir cunhas (ou *wedges*) na linha de novos elementos caso haja o aumento sucessivo do tamanho dos elementos ao longo da criação de novas linhas de elementos. A introdução de uma cunha na linha de novos elementos é mostrada na Figura 7.31.

Figura 7.31 - Esquema mostrando a inserção de uma cunha na linha de elementos.

Fonte: próprio autor do texto.

Os critérios para criar uma cunha apresentados no artigo de Blacker e Stephenson geram dúvidas sobre a possibilidade de sua generalização. Determina-se que as arestas de uma nova fronteira de pavimentação, formada pela linha de novos elementos, que apresentarem uma taxa de aumento de comprimento de 1,25 exatamente e que formarem ângulos internos menores que 183° nos nós compartilhados, devem ativar a formação de cunhas. No entanto, a maneira como foi explicado o processo, gera algumas questões:

- A variação de 1,25 é referente à aresta oposta do próprio elemento?
- Adotar tais valores para os critérios geram cunhas em regiões que teriam elementos de melhor qualidade sem elas.

Não são apresentadas considerações sobre o posicionamento destas cunhas, sendo feitas apenas restrições quanto ao ângulo formado entre elas, supondo que estas apresentarão disposições ao longo de uma circunferência e permitindo a formação de ângulos bem definidos entre suas posições. Os exemplos apresentados no artigo se limitam a geometrias com arcos de circunferência bem definidos. No caso do tratamento de geometrias genéricas, pode ser necessária a criação de uma dobra em uma porção da FP que não apresente um trecho de circunferência bem definido. No entanto, não é possível determinar ângulos bem definidos formados por eventuais cunhas. Em algumas situações como a da Figura 7.32, a formação de cunhas dispostas em uma configuração específica possibilita a criação de elementos de melhor qualidade. O artigo, entretanto, não apresenta uma metodologia para permitir este posicionamento de cunhas, o que direciona à conclusão de que esta decisão não é automática no método.

Figura 7.32 - Exemplo de cunhas dispostas a cada 90°.

Fonte: (Blacker & Stephenson, 1991)

Considerando estes pontos que impossibilitam a implementação de maneira mais clara desta etapa do método, foi desenvolvido um algoritmo com algumas alterações nos critérios, direcionando as adaptações para as geometrias testadas neste trabalho. Define-se que uma aresta da FP assume o status de qualificada quando ativa os critérios definidos para a inserção das cunhas. Estes critérios, por sua vez, foram alterados no desenvolvimento deste trabalho. Não mais são verificados os ângulos, mas apenas a alteração do tamanho das arestas. São verificadas duas razões: a razão de crescimento entre arestas opostas no mesmo elemento da linha de novos elementos, denominada r_e ; e a razão entre a aresta da fronteira de pavimentação e o tamanho médio das arestas da fronteira fixa de origem, denominada r_g . Definem-se l , como o comprimento da aresta da fronteira da pavimentação, l_{op} , o comprimento da aresta oposta a l no mesmo elemento e l_{ref} , o comprimento médio da aresta das fronteiras fixas. Assim, a aresta será dita qualificada quando

$$\begin{cases} r_e = \frac{l}{l_{op}} \geq 1,4 \\ r_g = \frac{l}{l_{ref}} \geq 1,5 \end{cases}$$

Os valores de referência para estas razões foram obtidos empiricamente, a partir de ajustes com os exemplos testados. Assim, caso $r_e \geq 1,4$ ou $r_g \geq 1,5$ a aresta obterá o status de qualificada.

A outra parte do algoritmo é a definição de quais arestas qualificadas serão separadas para adicionar a cunha. O artigo propõe uma verificação de ângulos entre a eventual posição de

cunhas, mas não explica claramente qual é o procedimento e nem como tratar casos de linhas de elementos que formem curvas diferentes de arcos de circunferência bem definidos, sendo que o único critério bem expresso foi a limitação de se inserir no máximo quatro cunhas em uma linha de novos elementos fechada ou próxima de se fechar (formando um arco de ângulo próximo a 360°). Devido à ausência de informações mais detalhadas do algoritmo, desenvolveu-se uma metodologia para determinar um ângulo de circunferência formado pela sequência de arestas qualificadas, adaptando a curva formada pelas arestas a um arco de circunferência. Definida a lista de arestas qualificadas consecutivas, são traçadas as medianas da primeira e da última aresta desta lista, determinando o ponto de intersecção. A Figura 7.33 mostra um esquema do procedimento.

Figura 7.33 - Esquema de determinação do ângulo para inserção das cunhas.

Fonte: próprio autor do texto.

As arestas da fronteira de pavimentação grifadas, de pontos médios M_1 e M_2 , delimitam uma sequência de arestas qualificadas, seguindo a orientação da fronteira de pavimentação, indicada pelas setas. A intersecção das medianas das arestas grifadas determina o ponto Q , e assim calcula-se o ângulo α definido pelos vetores $\overrightarrow{QM_1}$ e $\overrightarrow{QM_2}$, seguindo a orientação da fronteira de pavimentação. Aproxima-se a razão $\alpha/90^\circ$ para um número inteiro a fim de definir quantas cunhas serão adicionadas. Além do ângulo, é verificada a quantidade de arestas qualificadas na lista em questão. São necessárias 4 arestas qualificadas seguidas para cada cunha a ser adicionada. Assim, tenta-se inserir a quantidade de cunhas na linha de novos

elementos mediante a quantidade de vezes que α é divisível por 90° e mediante a quantidade de arestas existentes na lista. Foi verificado que, primando por melhor qualidade de elementos, deve-se deixar pelo menos duas arestas entre cunhas subsequentes. A Figura 7.34 mostra o pseudocódigo do algoritmo para determinar a inserção ou não de cunhas.

Figura 7.34 - Pseudocódigo para a inserção de cunhas.

```

do FindNextQualifiedEdges
  Determine Q point
  Determine  $\alpha$ 
  Determine integer  $n = \alpha/90^\circ$ 
  Determine integer  $m = \text{qualifiedEdges.size}/4$ 
  case  $n = 0$  or  $n = 1$ 
    if  $m \geq 4$ 
      ApplyWedge
  case  $n = 2$ 
    if  $m \geq 8$ 
      ApplyWedge x 2
    else
      ApplyLessWedges
  case  $n = 3$ 
    if  $m \geq 12$ 
      ApplyWedge x 4
    else
      ApplyLessWedges
  case  $n = 4$ 
    if  $m \geq 16$ 
      ApplyWedge x 4
    else
      ApplyLessWedges
until qualifiedEdgelist.size = 0

```

Fonte: próprio autor do artigo.

A adição de uma cunha em uma linha de novos elementos se faz a partir de 4 arestas qualificadas em sequência e o pseudocódigo da Figura 7.34 mostra a quantidade de cunhas a ser adicionada dependendo do ângulo formado e da quantidade de arestas qualificadas em sequência.

O procedimento de edição dos elementos para a inserção da cunha inicia-se com a definição dos nós n_1, n_2 e n_3 da fronteira de pavimentação que determinam as duas arestas centrais na sequência de quatro arestas qualificadas, bem como os nós p_1, p_2 e p_3 opostos aos primeiros. Devem-se remover algumas entidades, criar novas e se conectar as restantes para a formação de novos elementos, conforme esquematizado na Figura 7.35.

Figura 7.35 - Esquema mostrando a inserção da cunha.

Fonte: próprio da cunha.

Seguindo as etapas da esquerda para a direita na Figura 7.35, remove-se a aresta $\overline{n_2n_3}$ e cria-se o novo nó n'_2 que deve ser posicionado nas mesmas coordenadas de n_2 . São criadas as arestas $\overline{n'_2n_3}$ e $\overline{p_2n'_2}$, em seguida os nós n_2 e n'_2 são transladados até as arestas $\overline{n'_2n_3}$ e $\overline{n_1n_2}$ assumirem o comprimento de 1/3 do original, formando uma trinca na linha de novos elementos. Em seguida, cria-se o nó n_4 posicionado de acordo com a equação de suavização isoparamétrica:

$$V_{n_4} = V_{n'_2} + V_{n_2} - V_{p_2} \quad (7.2)$$

onde os vetores V_{n_i} representam o segmento de reta entre a origem do sistema de coordenadas e o nó n_i .

São adicionadas as arestas $\overline{n_4n'_2}$ e $\overline{n_2n_4}$, formando o novo elemento de cunha. Todo procedimento de tratamento da malha editada foi realizado utilizando os operadores de Euler modificados, já citados nas Seções de Intersecção e de Costura anteriormente. Por fim, uma suavização aplicada localmente reposiciona os nós envolvidos, proporcionando uma melhor qualidade aos elementos da região.

É importante ressaltar que os critérios adotados no algoritmo desenvolvido neste trabalho foram ajustados de modo empírico, priorizando elementos de melhor qualidade a partir das geometrias utilizadas como exemplos.

7.3.6 Suavização (*Smooth Module*)

O módulo de suavização apresenta duas versões:

Suavização de nós de fronteira: nesta, o foco da tentativa de melhora da qualidade dos elementos é direcionado aos nós da fronteira de pavimentação e aos nós da linha de novos elementos.

Suavização de nós internos: focada para melhorar a qualidade dos elementos com nós internos à malha e que não pertencem a nenhuma fronteira de pavimentação vigente.

O primeiro tipo é aplicado após a criação de uma nova linha de elementos e após a integração desta nova linha de elementos na malha. O segundo tipo é aplicado apenas na etapa de integração da linha de novos elementos na malha global, suavizando toda a malha. Os algoritmos referentes a esta etapa são expostos de modo relativamente claro no texto de Blacker e Stephenson e a seguir são explicados os modelos destas suavizações.

- **Suavização de nós de fronteira**

A suavização dos nós de fronteiras é composta de 3 tipos. Primeiramente, aplica-se o modelo de suavização isoparamétrico simplificado, que consiste em um ajuste ponto a ponto considerando os elementos que contém o nó selecionado. São calculados os vetores de ajuste V_i' para os nós selecionados N_i , baseando-se, para tal, nos vetores V_j , V_k e V_l dos nós N_j , N_k e N_l de cada elemento, em relação à origem O . A Figura 7.36 apresenta um esquema indicando os nós e vetores considerados para cada elemento.

Figura 7.36 - Esquema indicando os parâmetros para o modelo isoparamétrico.

Fonte: próprio autor do texto.

O vetor de ajuste do nó N_i é calculado por

$$V'_i = \frac{1}{n} \sum_{m=1}^n V_{mj} + V_{ml} - V_{mk} \quad (7.3)$$

onde o índice m indica o elemento analisado e que compartilha o nó N_i , e n é o número de elementos que compartilham este nó. É importante salientar que a ordem dos nós N_j , N_k e N_l deve ser respeitada. Eles devem ser listados em sequência horária ou anti-horária nos elementos em questão, de modo que o nó N_i não esteja entre nenhum par dos nós da lista (N_j , N_k e N_l).

Define-se o vetor de correção Δ_a como

$$\Delta_a = V'_i - V_i \quad (7.4)$$

A suavização isoparamétrica deve ser aplicada em todos os nós da fronteira de pavimentação. No entanto, caso o nó N_i seja compartilhado por 2 elementos apenas, se faz necessária a aplicação de outros ajustes como o de comprimento e o de ajuste angular. O ajuste de comprimento é calculado considerando o esquema da Figura 7.37.

Figura 7.37 - Esquema do ajuste de comprimento na suavização

Fonte: próprio autor do texto.

Considera-se que $V_{ij} = V'_i$ e que V_D é o vetor que conectava N_i e N_j antes de se aplicar a suavização isoparamétrica no nó N_i , ou seja, o vetor Δ_a . Definem-se $l_A = |V_{ij}|$ e $l_D = |V_D|$, assim o cálculo do ajuste de comprimento é dado por

$$\Delta_b = V_j - V_i + (\Delta_a + V_i - V_j) \frac{l_D}{l_A} \quad (7.4)$$

No caso do ajuste angular, deve-se realizar algumas manipulações geométricas conforme mostra a Figura 7.38.

Figura 7.38 - Esquema do procedimento de ajuste angular da suavização.

Fonte: próprio autor do texto.

O ajuste angular tem por objetivo encontrar o vetor P_{B2} . Inicia-se com a configuração da fronteira de pavimentação na situação anterior à suavização, e calcula-se P_{B1} como o vetor da bissetriz do ângulo formado entre os vetores P_{i-1} e P_{i+1} . Neste ponto, o artigo não define qual deveria ser a norma do vetor P_{B1} , então se considerou que sua norma seria igual à de P_{i+1} . Obtém-se em seguida o vetor P_{B2} na bissetriz do ângulo formado pelos vetores P_{B1} e P_i . Novamente foi definido que a norma do vetor P_{B2} seria igual à do vetor P_{B1} devido à falta desta informação no artigo. No entanto, os valores das normas de P_{B1} e P_{B2} podem ser arbitrários neste momento, não afetando o resultado do cálculo.

Segue-se com a definição do ponto Q de intersecção entre a reta definida por $\overline{N_{i-1}N_{i+1}}$ e a reta definida pelo vetor P_{B2} e pelo nó N_j , conforme é exemplificado na Figura 7.38. Assim, assumindo que $l_Q = |\overline{N_jQ}|$ e $l_D = |\overline{N_jN_i}|$, é realizada uma comparação destes valores:

$$|P_{B2}| = \begin{cases} \frac{l_Q + l_D}{2} & \text{se } l_D > l_Q \\ l_Q & \end{cases} \quad (7.5)$$

Calculado o vetor P_{B2} , define-se Δ_c como

$$\Delta_c = P_{B2} - P_i \quad (7.6)$$

Ao final, pode-se calcular o vetor de ajuste total Δ_i aplicado ao nó N_i , contanto que este nó compartilhe apenas dois elementos.

$$\Delta_i = \frac{\Delta_c - \Delta_b}{2} \quad (7.7)$$

No caso de nós com um, três ou mais elementos compartilhados, deve-se aplicar apenas Δ_a calculado anteriormente, ou seja, $\Delta_i = \Delta_a$.

- **Suavização de nós internos**

Após aplicada a suavização dos nós da fronteira, deve-se suavizar os nós internos da malha, considerando que os nós internos são aqueles que não pertencem nem às fronteiras de pavimentação e nem às fronteiras fixas. Selecionando um nó interno N_i , verificam-se quais são os outros nós conectados a ele. A Figura 7.39 mostra um exemplo da região da malha destacada para o cálculo desta suavização.

Figura 7.39 - Esquema mostrando os nós para o cálculo da suavização interna.

Fonte: próprio autor do texto.

Define-se um vetor de contribuição de ajuste de um nó N_j conectado à N_i como C_j , sendo $C_j = V_j$ para todos os casos exceto quando V_j pertence a uma fronteira fixa, o que exige uma mudança no valor deste vetor para:

$$C_j = V_j + \Delta_{c,j} \quad (7.8)$$

O vetor $\Delta_{c,j}$ é o vetor de ajuste apresentado anteriormente nesta mesma seção, considerando que a sequência $N_i N_{i+1} N_j N_{i-1}$ deve ser agora considerada como $N_i N_{j+1} N_j N_{j-1}$, possibilitando o cálculo de Δ_c .

Após a determinação dos C_j relativos a cada nó conectado a N_i , o vetor Δ_i pode ser determinado pela soma de todos estes vetores de contribuição ponderados pela normal dos mesmos.

$$\Delta_i = \sum_{j=1}^n \frac{|C_j| C_j}{\sum_{k=1}^n |C_k|} \quad (7.9)$$

onde n é o número de nós conectados a N_i .

7.3.7 Fechamento da fronteira (*Closing Boundary Module*)

Quando uma fronteira de pavimentação atinge a quantidade de 6 ou menos nós, se torna necessário fechá-la e, para realizar esta tarefa, foi desenvolvido um módulo específico no programa. A rotina verifica a quantidade de nós da fronteira e realiza os procedimentos específicos para fechá-la conforme indicado no diagrama da Figura 7.40.

Figura 7.40 - Módulo esquematizado de fechamento de fronteira.

Fonte: próprio autor do texto.

Caso a fronteira apresente apenas 4 nós, é formado um novo elemento automaticamente, considerando-os como seus nós formadores. Caso a fronteira possua 5 nós, deve-se a submeter a um algoritmo para inserir um nó a mais nesta fronteira, completando 6 nós. Este procedimento foi chamado de adição de nova faixa de elementos (*add a new band of elements*) e será explicado adiante. A fronteira com 6 nós é o padrão de tratamento e pode assumir uma série de diferentes configurações de fechamento de fronteira.

Fechar uma fronteira com 6 nós exige algumas verificações: Caso a mesma apresente um formato de rasgo (*rip*), com ou sem intersecção, conforme mostra a Figura 7.41 nos itens a e b, respectivamente, ela deve ser colapsada, unindo os elementos do seu contorno.

Figura 7.41 - Esquema para representação do colapso em fechamento de fronteira.

Fonte: próprio autor do texto.

A fronteira, para apresentar esta configuração, exige que se possua uma disposição dos nós em formato de rasgo, com uma certa simetria, tendo dois nós com ângulos internos menores ou iguais a 60° , enquanto os outros apresentam ângulos internos próximos de 180° . Estes valores de ângulos foram determinados arbitrariamente a partir de testes.

Caso a fronteira apresente um formato mais próximo de geometrias como retangular, triangular e circular, é necessário determinar qual é a configuração que será adotada para cada caso, dentre uma lista de 10 opções. Estas configurações são divididas por formato, conforme mostra a Tabela 7.1 na sequência. Os autores do artigo base não explicaram uma metodologia para definir qual padrão de fechamento deve ser adotado para cada caso e, devido a isso, foi

desenvolvida uma metodologia própria para realizar esta tarefa. A seguir, ela é explicada detalhadamente.

Definem-se os ângulos internos dos pontos de referência dos padrões de geometrias, enumerados de 0 a 5, como $u = (u_0, u_1, u_2, u_3, u_4, u_5)$ e os ângulos internos dos nós da fronteira que se deseja fechar como $a = (a_0, a_1, a_2, a_3, a_4, a_5)$. Deve-se calcular a diferença absoluta destes ângulos considerando a ordem entre eles para cada padrão da tabela. Os padrões da tabela geram um erro médio mínimo R que é obtido a partir da comparação de todas as configurações possíveis com relação à ordem dos nós da fronteira. A ordem dos nós da fronteira, que se deseja fechar, deve ser alterada cinco vezes, enviando o primeiro nó para o final da lista a cada alteração. O erro de cada ordem para cada ângulo é definido por r_i :

$$r_{i,j} = |u_i - a_{i,j}| \quad (7.10)$$

onde u_i é o ângulo interno referente ao nó $i = 0, \dots, 5$ da lista u e da lista a . A lista tem sua ordem alterada cinco vezes, deslocando o ângulo a_0 para o final a cada alteração, assim, o índice $j = 0, \dots, 5$ representa as alterações, sendo 0 a situação inicial. O erro médio E para cada padrão da tabela é dado por

$$E = \min \left(\frac{\sum_{i=0}^5 |u_i - a_{i,j}|}{6}, \text{com } j = 0, \dots, 5 \right) \quad (7.11)$$

Após definido o significado de E , deve-se encontrar o padrão que proporcione o menor valor de E .

Tabela 7.1 - Tabela dos padrões para preenchimento da fronteira com 6 nós.

Retângulo 1	Retângulo 2	Retângulo 3

Triângulo 1	Triângulo 2	Triângulo 3

Semi-Círculo 1	Semi-Círculo 2	Semi-Círculo 3

Círculo

Os ângulos da lista u são fixos em valores pré-determinados e, tomando como referência os padrões da Tabela 7.1, foram construídas a Tabela 7.2, a Tabela 7.3, a Tabela 7.4 e a Tabela 7.5 com os valores destes ângulos, em graus, para cada caso.

Tabela 7.2 - Ângulos internos dos padrões de fronteira no formato retangular

Retângulo 1	
u_0	90°
u_1	90°
u_2	180°
u_3	90°
u_4	90°
u_5	180°
Retângulo 2	
u_0	90°
u_1	90°
u_2	180°
u_3	180°
u_4	90°
u_5	90°
Retângulo 3	
u_0	90°
u_1	90°
u_2	180°
u_3	90°
u_4	180°
u_5	90°

Tabela 7.3 - Ângulos internos dos padrões de fronteira no formato triangular

Triângulo 1	
u_0	60°
u_1	180°
u_2	60°
u_3	180°
u_4	60°
u_5	180°
Triângulo 2	
u_0	60°
u_1	60°
u_2	210°
u_3	150°
u_4	90°
u_5	180°
Triângulo 3	
u_0	60°
u_1	60°
u_2	210°
u_3	150°
u_4	90°
u_5	180°

Tabela 7.4 - Ângulos internos dos padrões de fronteira no formato semi-circular

Semi-Círculo 1	
u_0	120°
u_1	70°
u_2	170°
u_3	170°
u_4	70°
u_5	120°
Semi-Círculo 2	
u_0	90°
u_1	78,75°
u_2	157,5°
u_3	157,5°
u_4	157,5°
u_5	78,75°
Semi-Círculo 3	
u_0	140°
u_1	130°
u_2	90°
u_3	90°
u_4	130°
u_5	140°

Tabela 7.5 - Ângulos internos dos padrões de fronteira no formato circular

Círculo	
u_0	120°
u_1	120°
u_2	120°
u_3	120°
u_4	120°
u_5	120°

Definida a configuração que gerou o menor erro médio para os ângulos internos, são criados os novos nós, as novas arestas e os novos elementos necessários para fechamento da fronteira seguindo o padrão escolhido.

Quando, eventualmente, é obtida uma fronteira com 5 nós, se faz necessário criar mais um nó para tratá-la da maneira padronizada. Foi desenvolvida uma abordagem que não existe no artigo base e que cria uma faixa de novos elementos na malha, procurando uma saída a partir da fronteira de pavimentação em questão até uma fronteira fixa. O algoritmo procura a maior aresta da fronteira de pavimentação e, em seguida, busca sequencialmente as arestas opostas, até alcançar uma aresta de uma fronteira fixa. Em seguida, todas as arestas buscadas são divididas em duas e os elementos desta faixa são, conseqüentemente, divididos em dois. Caso a busca por arestas termine na própria fronteira de pavimentação ou em outra presente no interior da malha, o processo é reiniciado com a segunda maior aresta da fronteira de pavimentação em foco e o processo segue sucessivamente. A Figura 7.42 mostra um esquema representativo da criação desta faixa de elementos.

Figura 7.42 - Esquema da adição da faixa de elementos em uma fronteira com 5 nós.

Fonte: próprio autor do texto.

Este tratamento de fronteiras com 5 nós resulta na quebra da homogeneidade de fronteiras fixas. Após a divisão desta aresta fixa, se faz necessário criar um algoritmo de suavização que modifique as fronteiras fixas alteradas por este procedimento a fim de tornar sua representação por arestas mais equalizada. Isso, no entanto, não foi feito neste trabalho pois

se considerou que envolveria o campo de pós-processamento da malha. Esta tarefa foi, consequentemente, direcionada para trabalhos futuros

7.3.8 Medição da Qualidade de Elemento

Foi necessário definir um algoritmo para medição da qualidade dos elementos gerados. Os elementos são comumente comparados aos seus modelos de referência, sendo o quadrado para elementos quadriláteros e o triângulo equilátero (ou isósceles) para elementos triangulares. No caso do foco deste trabalho, a verificação da qualidade de quadriláteros pode ser realizada por cálculos vinculados com o produto vetorial dos vetores formados pelas arestas com o tamanho das arestas e com os ângulos internos do elemento. Foi adotada a última opção por ser a mais simples e que atende às necessidades que surgiram ao longo do desenvolvimento do programa.

(Lindholm, 1983) propõe que o coeficiente de qualidade de 2 triângulos que compartilham uma aresta pode ser calculado por uma relação entre os coeficientes dos seus 2 triângulos formadores, sendo a configuração de maior q aquela a ser adotada:

$$q = \frac{2}{\left(\frac{1}{q_1} + \frac{1}{q_2}\right)} \quad (7.12)$$

onde q_1 e q_2 são os coeficientes de qualidade dos triângulos. No entanto, esta composição não pode ser tomada como medidora da qualidade de quadriláteros, pois o triângulo base é equilátero e a junção de 2 triângulos equiláteros não forma um quadrado. Se faz necessário, portanto, buscar ou desenvolver outros modelos de medição de qualidade de elementos quadriláteros.

Borouchaki e Frey (1998) apresentam um modelo de cálculo de qualidade de quadriláteros utilizado em seu processo de otimização de malhas. Considerem-se os quatro ângulos internos α_k , com $k = A, B, C$ e D , de um elemento quadrilátero Q , conforme mostra a Figura 7.43.

Figura 7.43 - Ângulos internos do elemento.

Fonte: próprio autor do texto.

O fator de qualidade de forma η do elemento Q é:

$$\eta(Q) = \max\left(0, 1 - \frac{2}{\pi} \max_k \left(\left|\frac{\pi}{2} - \alpha_k\right|\right)\right) \quad (7.13)$$

O fator de qualidade assume valor 1 caso o elemento seja um retângulo e 0 se o quadrilátero apresentar algum ângulo menor que zero ou maior que π . Assim, tendo o fator de qualidade para cada elemento, pode-se calcular o fator de qualidade global q de uma malha M como:

$$q(M) = \sum_{Q_i \in \mathcal{Q}} \eta(Q_i) \quad (7.14)$$

onde $\mathcal{Q}$ é o conjunto de quadriláteros que compõem a malha M .

Dentre os algoritmos apresentados, foi utilizado o algoritmo de Borouchaki e Frey (1998) para a mensuração do fator de qualidade de cada elemento da malha.

8 Tecnologias Empregadas

8.1 Linguagem de Programação

Os algoritmos de criação de malhas exigem certa velocidade de processamento, pois grande parte dos métodos exige verificações iterativas relacionando elementos da malha. Linguagens computacionais de alto nível, como *Python*, *Matlab*, *C#* e *Java*, apresentam processamento mais lento do que linguagens de nível mais baixo como *C* e *C++*, porque carregam funcionalidades e pacotes prontos a mais que linguagens mais básicas. Etapas como a verificação de intersecção de elementos e a verificação na inserção de novos nós na malha são exemplos de procedimentos típicos e laboriosos computacionalmente. Foi utilizada a linguagem *C++* devido à sua flexibilidade de programação com a abstração de classes, à velocidade de processamento e à compatibilidade de comunicação com outros *hardware*. Toda a codificação foi desenvolvida na *IDE (Integrated Development Environment) Visual Studio 2012*. Nos itens seguintes, se discutem as bibliotecas que foram adotadas para a execução de terminadas tarefas no programa desenvolvido.

8.2 Ferramenta de Exibição Gráfica

No ramo de aplicações gráficas desenvolvidas em *C++*, existem duas bibliotecas que são constantemente utilizadas tanto no meio acadêmico quanto no comercial. São conjuntos de rotinas e funções que se comunicam de maneira otimizada com os *hardware* gráficos, ou *GPUs (Graphic Processing Units)*. Conhecidas como *API (Application Programming Interface)*, estas duas bibliotecas são a *DirectX* e a *OpenGL*. A primeira é uma *API* projetada para funcionar apenas em sistemas operacionais da plataforma *Windows*. Já a segunda possui a característica de, além de funcionar em ambiente *Windows*, também ser compatível com vários outros sistemas operacionais como *UNIX* e *Mac*. Novas versões da *API* como *OpenGL ES* e *WebGL* permitem o desenvolvimento de aplicações gráficas tanto para *web browsers* como para dispositivos móveis (Celulares e *tablets*). É sabido que as duas bibliotecas apresentam desempenho semelhante com relação à velocidade e qualidade gráfica (HILLESLAND, 2013), sendo as diferenças de eficiência notáveis apenas em aplicações gráficas que exigem alto desempenho das placas de vídeo como renderizações muito detalhadas e com efeitos de iluminação e textura complexos. Com relação ao cumprimento dos objetivos deste trabalho,

não serão apresentadas situações com tamanha exigência de processamento gráfico e os recursos utilizados serão apenas os necessários para a representação de curvas e linhas por aproximação de segmentos de retas e pontos, bem como a utilização de polígonos para representar os elementos. Devido à semelhança de eficiência entre as duas *APIs*, optou-se pela portabilidade e pelo conhecimento prévio do autor do trabalho na biblioteca *OpenGL*. As bibliotecas *OpenGL* mais atuais, como a 4.5, incorporaram um processamento muitas vezes mais rápido que os das versões anteriores. Foi adotado um novo *pipeline* de fluxo de informações, direcionando a maior parte dos processos e cálculos para a GPU, que paraleliza os cálculos nos vários núcleos da placa gráfica, diminuindo o uso da CPU (ANGEL; SHREINER, 2012). As tarefas a serem realizadas na GPU e as informações dos vértices são enviadas como *arrays* de valores e como *buffers* para o *pipeline* do *OpenGL*, possibilitando controle direto do que é processado pela placa gráfica. A Figura 8.1 esquematiza a arquitetura do fluxo de informações na GPU.

Figura 8.1 - Pipeline da biblioteca OpenGL.

Fonte: (Angel & Shreiner, 2012)

É verificado que a GPU recebe as coordenadas no espaço 3D ou 2D dos vértices e também as propriedades desses vértices, como cores, normais, coordenadas de textura, índices de conexão, entre outros. As informações dos vértices são recebidas no *VertexProcessor*, onde é executado o *vertex-shader*, uma rotina específica do *OpenGL*. Os *shaders* são códigos em linguagem *GLSL*, semelhante ao *C*, executados pela placa gráfica e que consideram valores de entrada e de saída para cada etapa do *pipeline* de funcionamento da renderização. A etapa *Clipper and Primitive Assembler* realiza a seleção de apenas vértices contidos no espaço do campo de visão da “câmera” do sistema. Já a etapa *Rasterizer* processa a imagem gerada pela cena 3D ou 2D interpolando pixels entre os vértices, unindo-os por retas. Dessa maneira são formados os triângulos das superfícies dos modelos. Por fim, o processo do *Fragment Processor* forma os fragmentos interpolados pelos vértices, executa o *fragment-shader*, que utiliza um código *GLSL* manipulando os valores de entrada e retornando um valor de cor para

o fragmento, entendendo-se fragmentos por possíveis candidatos a pixel que será apresentado na imagem final.

A implementação desta *API* em *C++* é feita mediante algumas bibliotecas disponíveis publicamente ou comercialmente. Testaram-se de início as bibliotecas *freeGLUT* e *GLEW*, que possibilitaram a criação de uma janela com a exibição da saída da placa gráfica. No entanto, não foi possível associar as versões mais atuais da *API OpenGL*, da 3.0 em diante, em conjunto com elementos de interface de usuário como botões e caixas de texto empregando estas bibliotecas por problemas de compatibilidade. Seria necessário criar os próprios controles de interface de usuário, o que é inviável neste trabalho. Assim, direcionando a busca de bibliotecas disponíveis para o atendimento da necessidade de se reproduzir os modelos gráficos em conjunto a uma interface com usuário, determinou-se que seria adotado o *framework Qt 4.8*, conforme é descrito na sequência.

8.3 Interface de Usuário

Após vasta pesquisa de bibliotecas integrando as funcionalidades da *API OpenGL*, optou-se pelo uso do *framework Qt 4.8* neste trabalho, apesar de já estarem disponíveis versões mais atuais, tal como a *Qt 5.4*. A biblioteca apresenta uma série de funcionalidades que a torna versátil e aplicável em diversos tipos de programas, além de ser multiplataforma e ser apresentada com uma vasta documentação. O fato de escolher uma versão mais antiga é decorrente desta ser uma em que a sua associação ao código sendo desenvolvido pode ser realizada com a simples incorporação dos arquivos compilados e dos arquivos de "cabeçalho" ao projeto. Versões mais atuais direcionam os desenvolvedores que incorporam esse *framework* a utilizarem uma *IDE* própria do mesmo desenvolvedor, chamada *QtCreator*, para implementar as funcionalidades da biblioteca. O *framework* disponibiliza classes que criam janelas e que contém conjuntos de elementos de interface de usuário, chamadas *Widgets*. Um tipo específico de *Widget* disponível na biblioteca incorpora a *API OpenGL* internamente e permite, em associação com a biblioteca *GLEW*, exibir a saída da placa gráfica.

9 Resultados da Implementação

Um protótipo de um programa para a geração de malhas, nomeado *GreenMesh* e implementando o método de pavimentação, foi desenvolvido. O programa foi estruturado com um ambiente para visualização das geometrias importadas de um CAD próprio do Laboratório de Mecânica Offshore da USP. As informações de *loops*, de arcos de circunferência e de linhas retas são, entretanto, transferidas por meio de um arquivo de texto em formato XML. A codificação deste arquivo é explicada no Apêndice B. A janela inicial da aplicação é mostrada na Figura 9.1.

Figura 9.1 - Tela inicial do GreenMesh

Fonte: próprio autor do texto.

Nela podem-se observar os seguintes botões:

Abrir Arquivo: Abre uma nova janela para a busca do arquivo tipo XML com as informações do cabo umbilical, do tubo flexível ou da geometria que se deseja utilizar como base para a criação de malhas.

Gerar Malha: Inicia o módulo de geração de malhas utilizando o algoritmo de pavimentação. Após a conclusão do processo, é aberta uma nova janela exibindo a malha renderizada.

Neste espaço são visualizadas as geometrias das seções transversais de cabos umbilicais e tubos flexíveis após o arquivo XML ser carregado, conforme o exemplo da Figura 9.2.

Figura 9.2 - Visualização do arquivo de geometrias importado no GreenMesh.

Fonte: próprio autor do texto.

Além da imagem visualizada, é possível selecionar um dos componentes desta seção que estão todos listados à esquerda da imagem e, após a definição de um tamanho médio de aresta, clicando no botão “Gerar Malha”, o programa processará a geometria do componente selecionado e abrirá uma nova janela mostrando a malha criada.

Durante a confecção da ferramenta, notou-se a grande variedade de configurações que podem ocorrer durante o processo de geração da malha. O posicionamento dos elementos, as intersecções e os ângulos formados entre eles indicam o quão complexa é a previsão de

configurações para se criar os tratamentos para todas as ocorrências. Configurações geradas por figuras esbeltas exigem tratamento diferente daqueles voltados às configurações geradas por figuras com furos, por exemplo. Outro fator muito importante é a variação das configurações conforme se altera o tamanho médio do elemento. Essa alteração permite a obtenção, com o método de pavimentação, de configurações e elementos bastante diferentes dos anteriores. As conexões entre elementos, os tratamentos dos módulos de intersecção e de costura podem se alterar completamente para uma mesma geometria, apenas se alterando a discretização das fronteiras. Novamente, este elemento de aleatoriedade torna o refinamento do programa, a fim de tratar todas as possíveis ocorrências, um procedimento bastante trabalhoso.

Tentou-se gerar automaticamente as malhas de geometrias próprias dos componentes das seções transversais de cabos umbilicais e tubos flexíveis. Geraram-se as malhas de alguns exemplos representativos e que são expostos na sequência. A Figura 9.4 e a Figura 9.5 mostram malhas de retângulos na sequência, representando a seção transversal retangular de tendões. Nota-se que houve uma singularidade na segunda malha, com a criação de uma linha a mais interna à grade retangular. Este fenômeno é influenciado pela diferença de comprimento das arestas do retângulo, assim, quanto mais diferentes forem os comprimentos delas, maiores são as chances de as singularidades geométricas ocorrerem. Outro fator que contribui neste fenômeno é o critério adotado para ativação da costura entre arestas, pois devido à distância entre as arestas não atingir o valor esperado pelo critério, o módulo de costura não é ativado. A Figura 9.3 mostra as etapas da criação de uma malha irregular em um retângulo.

Figura 9.3 - Sequência de criação de elementos no retângulo.

Fonte: próprio autor do texto.

Nas configurações iniciais (a), (b) e (c) nota-se a distorção nos elementos gerados devido à diferença de comprimento das arestas da lateral e da base do retângulo. Isto contribui na instabilidade numérica para a não geração de uma malha estruturada. A distorção é propagada durante a geração da malha, ocasionando uma grande distância entre as arestas da fronteira de pavimentação da configuração (d). Como a distância entre as arestas é marginalmente maior que o definido no critério do programa para ativar o módulo de costura, uma nova linha de elementos é adicionada na configuração (e). Em seguida, o módulo de costura é ativado, pois se obtêm distâncias menores que o critério definido. A fronteira de pavimentação é colapsada e são obtidas duas novas fronteiras, mas com 5 nós cada uma, conforme mostrado na

configuração (f). Cria-se, portanto, cada faixa de elementos para adicionar mais um nó a cada fronteira, permitindo determinação do padrão de 6 nós mais adequado. Assim, a malha é fechada na configuração (g). A forma de criação de novos quadriláteros no algoritmo proposto por Blacker e Stephenson apresenta problemas logo na sua etapa inicial, na Figura 9.3.a, onde os quadriláteros da lateral do retângulo já são criados com uma certa distorsão. Conforme visto na definição dos vetores de projeção dos nós na Seção 7.3.1 deste texto, o módulo deste vetor é calculado apenas considerando o ângulo do nó base da FP e os tamanhos das duas arestas compartilhadas por este nó na própria FP, o que pode resultar em discrepâncias de tamanhos entre elementos próximos. Seria mais interessante para a qualidade geral da malha se o algoritmo conseguisse ajustar automaticamente a qualidade dos quadriláteros da linha de novos elementos, tomando como referência outras arestas da FP que estejam próximas. Esta ocorrência pode ser uma falha perigosa do método, comprometendo a qualidade da malha final.

Figura 9.4 - Malha estruturada de um retângulo.

Próprio autor do texto

Figura 9.5 - Malha não estruturada de um retângulo.

Fonte: próprio autor do texto

Outra geometria frequente na seção de cabos umbilicais e tubos flexíveis é a coroa que representa a seção transversal de capas plásticas, mangueiras e outros componentes em formatos próximos de tubos. A Figura 9.6 e a Figura 9.7 mostram uma malha menos discretizada e outra mais refinada de uma coroa, respectivamente. Neste caso, foi necessário o tratamento específico das fronteiras a fim de se obter malhas estruturadas de coroas, pois o procedimento de discretizar tanto a fronteira externa quanto a interna com arestas de mesmo tamanho, procedimento padrão adotado neste trabalho, gerou malhas mais distorcidas, como a mostrada na Figura 7.7. Ferramentas comerciais possuem procedimentos de controle da fronteira que possibilitam as discretizar a partir de um reconhecimento prévio da geometria, classificando os *loops* para direcionar o tratamento mais eficiente e gerar malhas de melhor qualidade. Como este tipo de automatização exige relativo tempo de desenvolvimento e se desvia do escopo do projeto, optou-se por direcionar esta tarefa para trabalhos futuros. Assim, foi realizado um tratamento específico para as coroas, discretizando as fronteiras interna e externa com o mesmo número de arestas.

Figura 9.6 - Coroa pouco discretizada.

Fonte: próprio autor do texto.

Figura 9.7 - Coroa mais discretizada.

Fonte: próprio autor do texto

Outra geometria regular bastante usual nas seções transversais de cabos umbilicais e tubos flexíveis é o círculo, que pode ser a seção de um *filler* ou de tendões de armadura de

tração. A Figura 9.8 e a Figura 9.9 mostram malhas de um mesmo círculo, mas com diferente discretização.

Figura 9.8 - Malha de um círculo pouco refinada.

Fonte: próprio autor do texto

Figura 9.9 - Malha de círculo mais refinada.

Fonte: próprio autor do texto.

Partindo para geometrias mais irregulares, foi testada a geometria da seção transversal de um *filler* comum em cabos umbilicais. Foram criadas malhas de elemento base com arestas de 20, 24, 30, 36 e 40 milímetros, e são apresentadas na Figura 9.10, Figura 9.11, Figura 9.12, Figura 9.13 e Figura 9.14 respectivamente.

Figura 9.10 - Malha do *filler* com aresta base de 20 mm.

Fonte: próprio autor do texto

Figura 9.11 - Malha do *filler* com aresta base de 24 mm.

Fonte: próprio autor

Figura 9.12 - Malha do *filler* com aresta base de 30 mm.

Fonte: próprio autor do texto.

Figura 9.13 - Malha do *filler* com aresta base de 36 mm.

Fonte: próprio autor do texto.

Figura 9.14 - Malha do *filler* com aresta base de 40 mm.

Fonte: próprio autor do texto.

Foram mensurados os fatores globais de qualidade para cada uma das cinco malhas utilizando o algoritmo apresentado na Seção 7.3.8 e estão indicados no gráfico da Figura 9.15.

Figura 9.15 - Gráfico com a qualidade média das malhas do *filler*.

Fonte: próprio autor do texto.

A redução do tamanho da aresta base propiciou um aumento aproximadamente linear na qualidade geral da malha, o que tende a indicar que o método foi implementado de maneira satisfatória. Os métodos geradores de malhas melhoram, por natureza, a qualidade média da malha conforme se aumenta o refinamento, ou seja, conforme é reduzido o tamanho do elemento base.

O método propõe o tratamento também de geometrias com furos e vazios, assim, a cargo de exemplo, foi gerada uma geometria simples de uma placa com furo. Sua malha é mostrada na Figura 9.16.

Figura 9.16 - Malha de uma placa com furo.

Fonte: próprio autor do texto.

No caso de geometrias mais irregulares que podem surgir nas seções transversais de *fillers*, como alguns exemplos provenientes de catálogos mostrados neste texto, foi desenhada a seção transversal de um *filler* genérico o suficiente para testar o programa desenvolvido. A geometria contém várias curvas, algumas convexas outras côncavas e com 2 furos irregulares, além disso apresenta algumas regiões delgadas. A Figura 9.17 mostra as geometrias dos *fillers* carregadas no *GreenMesh*.

Figura 9.17 - Exemplo de *fillers* importados no GreenMesh.

Fonte: próprio autor do texto.

Selecionado um dos *fillers* e impondo-se o tamanho de aresta médio de 13 milímetros, aplicou-se o método de pavimentação nesta geometria e o resultado é mostrado na Figura 9.18.

Figura 9.18 - Malha de um *filler* genérico inspirado nos exemplos de umbilicais do catálogo da empresa Akker.

Fonte: próprio autor do texto.

Alguns pontos são interessantes de salientar no caso desta malha: a região central mostrou como é a tendência de elementos formados na junção de duas vertentes de elementos, provenientes dos contornos dos furos. A ordem de criação de linhas de novos elementos condiciona os elementos formados. Caso se tivesse iniciado a criação das linhas de novos elementos nos furos em ordem diferente da adotada, o resultado da malha seria outro. A malha da Figura 9.18 mostra que pequenas diferenças nas fronteiras fixas que tornam a figura assimétrica podem gerar grandes diferenças na malha. Observa-se que as diferenças são propagadas para o centro da figura e se tornam regiões pouco estruturadas.

Pode ser observado um problema referente ao módulo de suavização no entorno dos furos. Os menores arcos convexos destas fronteiras geram linhas de novos elementos que aumentam o tamanho a cada nova linha criada. Estes elementos, entretanto, estão sendo submetidos à suavização geral da malha a cada iteração, mesmo nas iterações em que sua região não sofre alterações. Estas sucessivas suavizações resultam na retração da fronteira de pavimentação, já que o algoritmo de suavização proposto por Blacker e Stephenson (1991) tem a tendência de retrain a fronteira de pavimentação em curvas convexas, e isso explica os elementos achatados no entorno das fronteiras fixas dos furos. Uma possibilidade de se resolver o problema seria com a restrição da aplicação da suavização apenas em elementos de qualidade abaixo de um determinado valor, no entanto esta mudança exige um estudo de efetividade.

Outra ocorrência evidente nesta malha de um *filler* irregular é a necessidade de um pós-processamento da malha. A necessidade não é visível apenas nesse caso. Todos os exemplos mostrados neste trabalho podem ser submetidos a uma verificação de elementos próximos de losangos que podem ser colapsados. Este procedimento verificaria se a qualidade dos elementos no entorno do elemento em questão melhoraria com o seu colapso. Caso a verificação seja positiva, o elemento é colapsado e os elementos do entorno são unidos. O trabalho, porém, não teve o foco em procedimentos de pós-processamento de malha e, por isso, eventuais melhorias e lapidações na ferramenta GreenMesh devem ser realizadas em trabalhos futuros.

10 Conclusões

Uma vasta pesquisa bibliográfica dos métodos geradores de malhas foi realizada. Foi avaliado o método mais adequado para construir malhas de geometrias de seções de cabos umbilicais e tubos flexíveis mediante a avaliação de critérios definidos. Determinou-se que o método de pavimentação atenderia às necessidades aqui expostas e foi desenvolvido um programa de computador que o incorporou. O desenvolvimento do protótipo de programa gerador de malhas foi satisfatório e gerou malhas de exemplos característicos de estruturas de cabos umbilicais e tubos flexíveis prontas para serem utilizadas em cálculos com o Método dos Elementos Finitos. Conforme quantificado no exemplo da qualidade das malhas do *filler* na Figura 9.15, foi possível obter bons valores médios de qualidade das malhas e uma progressão crescente na melhora da qualidade conforme a malha é refinada. Tais resultados apontam para uma implementação satisfatória do Método de Pavimentação

A implementação do programa foi essencial para o entendimento das dificuldades e desafios que surgem durante a criação de uma ferramenta de geração de malhas. Devido à particularidade de componentes destes equipamentos e à dificuldade de se tratar automaticamente geometrias genéricas da melhor maneira, alguns pontos importantes do processo de geração de malhas foram levantados.

Se faz necessária a criação de uma metodologia de reconhecimento das fronteiras que relacione cada entidade (linhas e curvas) com suas vizinhanças em geometrias genéricas. O tratamento da geometria de uma coroa, ou anel, mostrou que é imprescindível relacionar as duas fronteiras (interna e externa) a fim de as discretizar de maneira favorável para a criação de uma malha estruturada. No programa, foi desenvolvida uma rotina específica para este caso, em que se discretiza com um mesmo número de arestas tanto a fronteira interna quanto a externa.

Ao final do desenvolvimento do protótipo do programa, concluiu-se que os módulos de tratamento de intersecções e de costuras deveriam ser reformulados. Uma abordagem mais genérica poderia ser realizada sem a preocupação de se manipular apenas os elementos da linha de novos elementos, pois esta diferenciação promoveu maiores trabalhos e complicações de uso, como as relatadas nas seções deste trabalho. O tratamento de intersecções e de costuras se baseou majoritariamente na identificação de padrões de configuração e na edição da malha de

cada um separadamente, o que promove margem para o surgimento de exceções nas eventuais configurações que estiverem nos padrões pré-determinados. Se faz, portanto, necessária a implementação de um código mais genérico na forma de tratar estas ocorrências, visando melhoria na robustez do programa.

Observaram-se as limitações de modelagem do método proposto no artigo de Blacker e Stephenson (1991). Seja por ausência de informação, ou por alguma etapa apresentar decisões questionáveis, o método de pavimentação apresenta gargalos críticos no seu funcionamento. Pode-se ressaltar os seguintes pontos:

- Há discussões sobre a estrutura de dados mais indicada para ser utilizada no programa, no quesito da forma de modelar as entidades geométricas e como elas se relacionam. Vertentes que discorrem sobre meias-arestas (*half-edge*) como a de Silva Filho, *et al.* (2014), tratam de abordagens diferenciadas.
- Verificou-se, ao final, a necessidade de se criar uma metodologia para o pré-processamento das fronteiras, possibilitando a identificação de características de cada *loop* dado, a ponto de os discretizar da maneira mais adequada para a geração de malhas de melhor qualidade. Neste trabalho, criou-se uma excessão na discretização das fronteiras de anéis, o que não é interessante para um programa genérico e automático.
- A ordem de aplicação dos módulos do método de pavimentação é questionável, pois seja neste trabalho, como nos de White e Kinney (1997) e Silva Filho, *et al.* (2014), são propostas alterações na ordem das etapas do método ou até mesmo nos próprios módulos, por todos os trabalhos alegarem uma melhora de eficiência.
- Também com relação à estrutura de dados, o procedimento para tratamento de intersecções e costura não é claro em nenhum artigo que explica o método de pavimentação. Apresentar uma visão geral de uma possível estrutura de dados a ser adotada, colabora para a descrição da metodologia para tratar estes 2 módulos.
- Os critérios para ativação da inserção de dobras e cunhas na malha, apresentados por Blacker e Stephenson (1991), propõem valores questionáveis e que, em alguns casos, tendem a gerar singularidades no processo e também não é observada a priorização da qualidade dos elementos.

Foi observado um problema na aplicação do módulo de suavização. O artigo base do método de pavimentação não detalha se a suavização durante a criação da malha deve ser aplicada em regiões específicas ou em toda a malha ou na fronteira de pavimentação. Como apresentado na Figura 9.18, a primeira linha de novos elementos gerados a partir dos furos da figura sofreu sucessivas suavizações, o que resultou a retração da fronteira de pavimentação do local. O algoritmo de suavização tende a retrair a fronteira de pavimentação de uma região convexa e, após sucessivas suavizações, os elementos de uma região convexa desta fronteira de pavimentação tendem a ficar achatados.

11 Comentários Finais e Trabalhos Futuros

A primeira etapa deste trabalho possibilitou a imersão no estado da arte dos métodos geradores de malhas para figuras bidimensionais. Concluiu-se que esta etapa é o passo inicial e essencial para a criação de malhas tridimensionais. A existência de grande quantidade de trabalhos tratando deste assunto ao longo de décadas mostra como se trata de um problema atual e com vários ramos de aplicação na ciência. Dentre a vasta bibliografia disponível, selecionou-se o trabalho que mais bem atendeu aos requisitos de implementação no problema de gerar malhas para seções transversais de cabos umbilicais e de tubos flexíveis. Vários artigos apresentam métodos superficialmente, omitindo etapas importantes para seu entendimento pleno. Também se verificaram vários métodos apresentados como genéricos, mas cujas explicações não desenvolviam a aplicação em geometrias genéricas e com vazios internos. No entanto, tendo filtrado a grande quantidade de trabalhos encontrados, foi possível desenvolver discussões sobre aqueles mais bem apresentados e mais genéricos, promovendo a seleção final do algoritmo de pavimentação.

Apresentou-se o programa *GreenMesh*, um protótipo de ferramenta computacional que recebe informações de geometrias e *loops*, permitindo a visualização das figuras e a geração de malhas das geometrias selecionadas. Nele foi implementado o método de pavimentação e foram geradas malhas satisfatórias das seções transversais representativas de componentes de tubos flexíveis e de cabos umbilicais. O programa atendeu à proposta e as malhas geradas podem ser facilmente utilizadas em rotinas de simulações pelo Método dos Elementos Finitos. Adicionalmente à proposta deste trabalho, a etapa mais difícil para a geração de malhas tridimensionais de cabos umbilicais e tubos flexíveis foi alcançada, já que praticamente todas as malhas 3D dos componentes destes dispositivos são figuras de seções transversais planas projetadas no espaço tridimensional ao longo de um eixo helicoidal. Assim, outro trabalho futuro que se pode desenvolver é a adaptação do *GreenMesh* para gerar as estruturas tridimensionais destes equipamentos. Realizar este trabalho não apresentará muitas dificuldades, pois será necessário apenas gerar as malhas tridimensionais com a projeção da malha da seção transversal dos componentes ao longo de um eixo helicoidal e implementar um algoritmo de iluminação para aperfeiçoar a visualização dos modelos. Pode ser verificada certa dificuldade no desempenho do processamento gráfico, pois, dependendo do modelo, será necessário criar vários polígonos, várias arestas e vários nós no espaço tridimensional, o que pode exigir muito processamento gráfico da máquina.

O programa, no entanto, é apenas um protótipo. Ao final deste trabalho de mestrado foi percebida a necessidade de se alterar a forma com que são feitos os tratamentos de intersecção e costura, para permitir situações mais genéricas sem muito trabalho extra de desenvolvimento. Esta alteração resultaria em mudanças na estrutura de dados e, conseqüentemente, alterações em todos os módulos do programa. Seu desempenho pode ser melhorado com mais testes de variadas geometrias e com testes de “stress”, que são testes de aumento e de diminuição sucessiva do tamanho do elemento nas malhas, até limites de tamanhos de elementos inviáveis.

Assim, estas modificações e melhorias são tarefas de interessante foco em trabalhos futuros para promover melhorias de desempenho e da abrangência de uso da ferramenta.

12 Bibliografia

- AKKER solutions - power cable catalog*. (05 de 2015). Fonte: Akker Solutions.
- Angel, E., & Shreiner, D. (2012). *Interactive Computer Graphics: A Top-Down Approach with Shader-Based OpenGL*. Addison Wesley.
- (2002). *API_RP_17B - recommended practice for flexible pipe*.
- (1998). *API_SPEC_17E - specification for subsea production control umbilicals*. American Petroleum Institute.
- Atalay, F. B., Xu, D., & Ramaswami, S. (2009). Quadrilateral Meshes with Bounded Minimum Angle. *ACM*.
- Azarenok, B. (2012). 2D structured grid generation method producing a mesh with prescribed properties near boundary. *Engineering with Computers*, 409-418.
- Baehmann, P. L., Wittchen, S. L., Shephard, M. S., Grice, K. R., & Yerry, M. A. (1987). Robust Geometricallybased, Automatic Two-Dimensional Mesh Generation. pp. 1043-1078.
- Baerentzen, J. A., Gravesen, J., Anton, F., & Aanaes, H. (2013). Triangle Mesh Generation: Delaunay Triangulation. Em J. Baerentzen, J. Gravesen, F. Anton, & H. Aanaes, *Guide to Computational Geometry Processing* (pp. 241-260). Springer.
- Bakke, T. (2009). *UmbiliCAD® - User Manual*. Ultra Deep.
- BANK, R. E., & SMITH, R. K. (1997). Mesh Smoothing Using a Posteriori Error Estimates. *SIAM Journal on Numerical Analysis*, 979-997.
- Blacker, T. D., & Stephenson, M. B. (1991). Paving: a new approach to automated quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering*, vol. 32, 811-847.
- Bommes, D., Zimmer, H., Lévy, B., Pietroni, N., Puppo, E., Silva, C., & Zorin, M. T. (2012). Quad-Mesh Generation and Processing: a survey. *ACM*.
- Borouchaki, H., & Frey, P. J. (1998). Adaptive triangular-quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering*, Vol. 41, 915 -934.
- Bykat, A. (1976). Automatic generation of triangular grid : I-subdivision of a general polygon into convex subregions. II-triangulation of convex polygons. *International Journal for Numerical Methods in Engineering*. Volume 10, 1329-1342.
- Cavendish, J. C. (1974). Automatic triangulation of arbitrary planar domains for the finite element method. *International Journal for Numerical Methods in Engineering*, 679–696.
- Cavendish, J. C., & Wixom, J. A. (1975). Finite element mesh generation for planar and shell type structures. *ACM*, 35 - 37.

- Chae, S.-W., & Jeong, J.-H. (1997). Unstructured surface meshing using operators. *Proceedings, 6th International Meshing Roundtable*, 281-291.
- Chernikov, A. N., & Chrisochoides, N. P. (2008). Algorithm 872: parallel 2D constrained delaunay mesh generation. *ACM Transactions on Mathematical Software*.
- Chew, L. P. (1993). Guaranteed-Quality mesh generation for curved surfaces. *ACM*.
- Choi, J. (2011). Development of centering method for automatic generation of a quadrilateral mesh. *International Journal of CAD/CAM*.
- Demonstração do SketchPad*. (17 de 06 de 2015). Fonte: YouTube:
https://www.youtube.com/watch?v=USyoT_Ha_bA
- Farin, G., Hoschek, J., & Kim, M. -S. (2002). *Handbook of computer aided design*. Elsevier.
- Fenwick, J. E. (2005). *A survey of meshing algorithms and a multi-purpose mesh generator*. University of California, San Diego.
- Haber, R., Shephard, M., Abel, J., Gallagher, R., & Greenberg, D. (1978). A generalized graphic preprocessor for two-dimensional finite element analysis. *ACM*.
- Haxton, M. M. (1999). *Norway Patente Nº 5,902,958*.
- Hillesland, K. (2013). OpenGL and DirectX. *SIGGRAPH*.
- Ho-Le, K. (1988). Finite element mesh generation methods: a review and classification. *IEEE*.
- Joe, B. (1995). Quadrilateral mesh generation in polygonal regions. *Computer Aided Design, Vol.27*, 209-222.
- Johnen, A., Ernst, D., & Geuzaine, C. (2014). Sequential decision-making approach for quadrangular mesh generation. *Springer*.
- Jones, R. E. (1975). The QMESH mesh generation package. *ACM*.
- Klein, R., & Straiber, W. (1995). Large mesh generation from boundary models with parametric face representation. *ACM*, 431 - 440.
- Knapp, R. H., Le, T., & Cruickshank, M. (1991). Design methodology for undersea umbilical cables. *IEEE*, 1320-1328.
- Lai, Y.-K., Hu, S.-M., & Martin, R. R. (2009). Automatic and topology-preserving gradient mesh generation for image vectorization. *ACM*.
- Lee, Y. T., Pennington, A. d., & Shaw, N. K. (1984). Automatic finite-element mesh generation from geometric models - A point-based approach. *ACM*, 287 - 311.
- Liang, X., & Zhang, Y. (2013). An octree-based dual contouring method for triangular and tetrahedral mesh generation with guaranteed angle range. *Springer*, 211 - 222.

- Lindholm, D. A. (1983). Automatic triangular mesh generation on surfaces of polyhedra. *IEEE Transactions on Magnetics*, Vol. MAG-19, No. 6, 2539-2542.
- Lo, S. H. (1985). A new mesh generation scheme for arbitrary planar domains. *International Journal for Numerical Methods in Engineering*, 1403-1426.
- Lohner, R., Parikh, P., & Gumbert, C. (1988). Interactive generation of unstructured grid for three dimensional problems. *Numerical Grid Generation in Computational Fluid Mechanics*, 687-697.
- Ma, X., & Zhao, G. (2014). An automated approach to quadrilateral mesh generation with complex geometric feature constraints. *Engineering with Computers*.
- Marcum, D. L., & Weatherill, N. P. (1995). Unstructured grid generation using iterative point insertion and local reconnection. *AIAA Journal*, vol 33, no. 9, 1619-1625.
- MARINTEK. (2010). *Uflex 2D User Manual*.
- Miranda, A. C., & Martha, L. F. (2014). Hierarchical template/based quadrilateral mesh generation. *Springer*.
- Navarro, X. R. (2009). *Paving the path towards automatic hexahedral mesh generation*. Barcelona: Phd. Thesis, Universitat Politècnica de Catalunya.
- Nexans Official Website. (08 de 05 de 2015). Fonte: Nexans Official Website: http://www.nexans.com/eservice/Corporate-en/navigatepub_310000_-32913/Nexans_wins_power_umbilical_contract_from_OneSubse.html
- Nowotny, D. (1997). Quadrilateral mesh generation via geometrically optimized domain decomposition. *Proceedings, 6th International Meshing Roundtable*, 309-320.
- Owen, S. J. (1998). A survey of unstructured mesh generation technology. *CiteSeerX*.
- PATRAN, M. (2015). <http://www.mscsoftware.com/product/patran.html#ww444985>. Acessado em 28/02/2015 12:00:00.
- Peraire, J., Vahdati, M., Morgan, K., & Zienkiewicz, C. (1987). Adaptive remeshing for compressible flow computations. *Journal of Computational Physics*, 449-466.
- Phillips, L. R., Mitchiner, J. L., Blacker, T. D., & Lin, Y. T. (1988). A knowledge system for automatic finite element mesh generation: AMEKS. *ACM*, 668-678.
- Pirzadeh, S. (1993). Unstructured viscous grid generation by advancing-layers method. *AIAA*, 420-434.
- Provasi, R., Meirelles, C. O., & Martins, C. A. (2010). CAD software for cable design: A three-dimensional visualization tool. *ASME 2010 29th International Conference on Ocean, Offshore and Arctic Engineering*, 447-455.
- (2015). *PRYSMIAN deep water umbilicals catalog*.

- Rebay, S. (1993). Efficient unstructured mesh generation by means of Delaunay Triangulation and Bowyer-Watson algorithm. *Journal Of Computational Physics*, vol. 106, 125-138.
- Rodriguez, J., Weinbub, J., Pahr, D., Rupp, K., & Selberherr, S. (2013). Distributed high-performance parallel mesh generation with ViennaMesh. *Springer*, 548-552.
- Ruppert, J. (1992). A new and simple algorithm for quality 2-dimensional mesh generation. *ACM*.
- SACK, J. -R., & URRUTIA, J. (1999). *Handbook of computational geometry*. North-Holland.
- Sadek, E. A. (1980). A scheme for the automatic generation of triangular finite elements. *International Journal for Numerical Methods in Engineering*, Vol. 15, 1813-1822.
- Schöberl, J. (1997). NETGEN: An advancing front 2D/3D-mesh generator based on abstract rules. *Springer*, 41-52.
- Sheffer, A., & Üngör, A. (2001). Efficient adaptive meshing of parametric models. *ACM*, 59-70.
- Shewchuk, J. R. (1996). Triangle: Engineering a 2d quality mesh generator and delaunay triangulator. *First Workshop on Applied Computational Geometry, ACM*, 124-133.
- Shewchuk, J. R. (2000). Mesh generation for domains with small angles. *ACM*.
- Shi, M., Zhang, Y. F., Loh, H. T., Bradley, C., & Wong, Y. S. (2005). Triangular mesh generation employing a boundary expansion technique. *Springer*, 54–60.
- Silva Filho, J. B., Tsuzuki, M. S., Ueda, E. K., Martins, T. C., Takimoto, R. Y., Gallo, G. B., . . . Tiba, H. (2014). Mesh Generation for Surfaces with Distinct Boundary Segmentation. *12th IEEE International Conference on Industrial Informatics (INDIN)*.
- Stroud, I., & Nagy, H. (2011). *Solid Modelling and CAD Systems*. Springer.
- Takayama, K., Panozzo, D., Sorkine-Hornung, A., & Sorkine-Hornung, O. (2013). Sketch-Based Generation and Editing of Quad Meshes. *ACM*.
- Talbert, J. A., & Parkinson, A. R. (1990). Development of an automatic, generator using quadrilateral elements and bézier curve boundary definition two-dimensional finite element mesh. *International Journal for Numerical Methods in Engineering*, 1551-1567.
- Tam, T. K., & Armstrong, C. G. (1991). 2D finite element mesh generation by medial axis subdivision. *Advances in Engineering Software*, Vol.13, 313-324.
- TECHNIP flexible pipe catalog. (2015).
- Tracy, F. T. (1977). Graphical pre- and post-processor for 2-dimensional finite element method programs. *SISGRAPH*, 8-12.
- Unruh, V., & Anderson, D. C. (1992). Feature-based modeling for automatic mesh generation. *Engineering with Computers*, Springer, 1-12.

- White, D. R., & Kinney, P. (1997). Redesign of the paving algorithm: Robustness enhancements through element by element meshing. *Proceedings, 6th International Meshing Roundtable, Sandia National Laboratories*, 323-335.
- Wördenweber, B. (1983). Finite-element mesh generation from geometric models. *COMPEL - The international journal for computation and mathematics in electrical and electronic engineering*, Vol. 2 Iss 1 pp. 23 - 33.
- Yerry, M. A., & Shephard, M. S. (1984). Automatic three-dimensional mesh generation by the modified-octree technique. *International Journal for Numerical Methods in Engineering*, Vol. 20, 1965-1990.
- Zhu, J., Zienkiewicz, O., Hinton, E., & Wu, J. (1991). A new approach to the development of automatic quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering*, Vol.32, 849-866.

13 Apêndice

A. História da Geometria Computacional

O termo CAGD (*Computer Aided Geometry Design*) foi primeiramente apresentado por R. Barnhill e R. Riesenfeld em 1974 na Universidade de Utah, em uma conferência sobre o tema para reunir pesquisadores dos Estados Unidos e Europa (FARIN; HOSCHEK; KIM, 2002). As bases para esta linha de pesquisa foram motivadas por trabalhos anteriores à década de 70. Já na época da Roma antiga, os cascos de madeira de embarcações eram fabricados com base em modelos de curvas reutilizáveis, permitindo que algumas geometrias dos barcos fossem armazenadas e evitando trabalho redundante de projeto. Os projetistas de embarcações de Veneza, entre os séculos XIII e XVI, desenvolveram projetos de cascos de embarcações com curvas semelhantes às NURBS (*Non Uniform Rational Basis Spline*) dos dias atuais. Este tipo de curva está brevemente explicado no Apêndice C. Seguindo estes exemplos históricos registrados, é direta a conclusão de que o desenho de curvas aplicado a projetos mecânicos teve seu início na indústria naval, sendo assimilada no século XX pela indústria automotiva e aeronáutica, tomando os modelos representativos da geometria de Monge e da geometria projetiva de Pascal. A partir do final da Segunda Guerra Mundial, estudiosos como R. Liming e S. Coons iniciaram trabalhos para relacionar curvas de desenhos de aeronaves com padrões numéricos de curvas parametrizadas que são possíveis de serem armazenados em tabelas (FARIN; HOSCHEK; KIM, 2002).

Com o advento dos primeiros computadores, ao longo das décadas de 50 e 60, pesquisadores direcionaram seus trabalhos para representar desenhos virtualmente, sem a necessidade de armazenar grande quantidade de documentos de projeto. Na empresa General Motors ocorreu o desenvolvimento do primeiro sistema CAD (*Computer-aided Design*)/CAM (*Computer-aided Manufacturing*) chamado DAC-I (*Design Augmented by Computer*) que permitia gerar alguns poucos modelos de curvas e superfícies. Concomitantemente, outros pesquisadores em países como Inglaterra, Estados Unidos e França também focavam no desenvolvimento de sistemas CAD, iniciando o intercâmbio de conhecimentos apenas a partir da década de 70 com a criação da disciplina CAGD. Paralelamente, avanços na criação de dispositivos gráficos da época propiciavam o desenvolvimento da tecnologia do *hardware* dos monitores, sendo a tecnologia dos monitores iniciada com os de raios catódicos, ou CRT (*Cathode Ray Tube*). Sua utilização teve início nos osciloscópios e foi essencial para a criação de *sistemas* CAD, pois permitia o desenho de curvas em sua tela. O primeiro sistema de computação gráfica interativa foi desenvolvido no doutorado de Ivan Sutherland, no MIT (*Massachusetts Institute of Technology*), chamado *SketchPad*. O sistema era composto por um painel multifunção, uma caneta ótica e um monitor de raios catódicos. A Figura 13.1 mostra uma foto do dispositivo sendo operado.

Figura 13.1 - Sistema Sketchpad de Ivan Sutherland.

Fonte: (Demonstração do SketchPad, 2015)

Paralelamente a essas pesquisas, foi notável a contribuição de Paul de Faget de Casteljaud que desenvolveu um algoritmo para o cálculo de curvas de Bézier durante seu trabalho no laboratório da companhia automobilística francesa Citroën. Também em Paris, Pierre Bézier,

que trabalhava na empresa Renault, desenvolvia a representação de curvas por polinômios (verificar Apêndice C sobre curvas de Bézier), o que foi a base para o desenvolvimento do programa de CAD/CAM da empresa, chamado UNISURF. Estes estudos no âmbito automotivo influenciaram a empresa aeronáutica francesa Dassault a desenvolver um sistema de CAD chamado EVE, que evoluiu para o atualmente reconhecido programa CATIA.

Já em 1960, C. de Boor trabalhava no laboratório de pesquisas da companhia General Motors e utilizava B-Splines (verificar Apêndice C) na representação de curvas, as consolidando como modelo matemático em CAGD. As B-Splines se tornaram o núcleo de desenhos de curvas em sistemas CAD, futuramente, na década de 70. Posteriormente a C. de Boor, em 1968, K. Versprille apresentou um modelo de representação de curvas que culminou nas B-Splines generalizadas, ou NURBS (*Non-Uniform Rational B-Splines*), em sua tese de doutorado e rapidamente este tipo de representação de curvas se tornou a base para qualquer desenho dos sistemas CAD até os dias atuais. Sua eficiência se deve em grande parte pela possibilidade de representar curvas de variados formatos, permitindo a generalização de sua aplicação de acordo com a necessidade do tipo de curva de design. O alcance a diferentes públicos atingiu todos os setores de design, como ocorreu nas empresas Boeing, SDRC e Unigraphics (FARIN; HOSCHEK; KIM, 2002).

B. Formato do Arquivo de Geometria de Entrada do GreenMesh

O arquivo de entrada da ferramenta GreenMesh foi codificado como um arquivo de texto em formato XML. São especificadas entidades com as diferentes hierarquias esquematizadas nas figuras a seguir:

Figura 13.2 - Elementos de nível hierárquico 0, 1 e 2.

Fonte: próprio autor do texto.

Figura 13.3 - Elementos pertencentes ao *Loop*. Níveis de hierarquia de 2, 3, 4.

Fonte: próprio autor do texto.

Figura 13.4 - Elementos contidos na entidade *Arc*.

Fonte: próprio autor do texto.

Figura 13.5 - Elementos contidos na entidade *Line*.

Fonte: próprio autor do texto.

Cada componente representa figuras formada por conjuntos de *loops*, sendo cada *loop* transformado em uma fronteira fixa no programa. Uma armadura de tração, por exemplo, seria representada como um conjunto de componentes, com cada um formado por um *loop* externo, representando um tendão. Os *loops* são formados de uma lista ordenada de *Segments*. Limitou-se o trabalho com a utilização de arcos de circunferência (*Arc*) e segmentos de retas (*Line*) compondo as entidades *Segments*, pois estes tipos de entidades geométricas já permitem a representação de figuras típicas de seções transversais de tubos flexíveis e cabos umbilicais. Assim, a entidade do tipo *Arc* armazena as coordenadas cartesianas do ponto inicial e final do arco, bem como o ponto médio deste arco. No caso da entidade tipo *Line*, são armazenadas apenas as coordenadas cartesianas dos pontos final e inicial. A propriedade *Loop Type* indica o tipo do *loop*, assumindo o valor 0 para *loops* externos e o valor 1 para *loops* internos. É importante ressaltar que o *loop* deve ter seus segmentos dispostos em ordem com o ponto final de um segmento sendo o igual ao ponto inicial do próximo segmento. Um projeto *Cable* pode ter quantos componentes (*Component*) forem necessários e o algoritmo de geração de malha tratará um componente por vez, conforme o usuário determina qual deles terá a malha criada. A Figura 13.6 mostra um trecho de um arquivo do tipo XML para a entrada no GreenMesh.

Figura 13.6 - Exemplo de trecho do arquivo de entrada do GreenMesh.

```

1  <?xml version="1.0" encoding="utf-8" standalone="yes"?>
2  <Cable>
3    <Component>
4      <Id>0</Id>
5      <Name>FILLER0</Name>
6      <Color_Id>1</Color_Id>
7      <Loop>
8        <Loop_Type>0</Loop_Type>
9        <Segments>
10         <Arc>
11           <StartPoint>
12             <X>-143.137631458619</X>
13             <Y>-116.785736930346</Y>
14           </StartPoint>
15           <MiddlePoint>
16             <X>0.350074510423085</X>
17             <Y>-67.9292100325689</Y>
18           </MiddlePoint>
19           <EndPoint>
20             <X>146.862368541381</X>
21             <Y>-106.785736930346</Y>
22           </EndPoint>
23           <CenterPoint>
24             <X>9.37180992611732</X>
25             <Y>-329.559537087703</Y>
26           </CenterPoint>
27           <Angle_Length>67.3136143870978</Angle_Length>
28         </Arc>
29         <Arc>
30           <StartPoint>
31             <X>146.862368541381</X>
32             <Y>-106.785736930346</Y>
33           </StartPoint>
34           <MiddlePoint>
35             <X>355.764981292252</X>
36             <Y>26.0623076359446</Y>
37           </MiddlePoint>
38           <EndPoint>
39             <X>118.035208047554</X>
40             <Y>95.152534674592</Y>
41           </EndPoint>

```

Fonte: próprio autor do texto.

C. Formas Geométricas e Modelos Representativos

Na geometria computacional, trabalha-se com primitivas geométricas definidas como pontos, segmentos de retas, arcos e curvas, que, quando unidas, podem formar figuras planas ou superfícies no espaço. Com o intuito de se representar matematicamente todas estas entidades geométricas e possibilitar manipulações computacionais, foi definido um sistema em que todas as formas geométricas dos espaços 2D e 3D, parametrizáveis ou não, são representadas por grupos dessas entidades. O sistema é proveniente da geometria projetiva e da geometria afim, que se utilizam da álgebra linear para desenvolver as formulações matemáticas. A definição de um espaço afim é dada por um ponto a_o e um conjunto de vetores $v_1, \dots, v_n$, de modo que cada ponto do espaço afim pode ser escrito como:

$$\mathbf{P} = \mathbf{a}_o + \xi_1 \mathbf{v}_1 + \dots + \xi_n \mathbf{v}_n = \mathbf{a}_o + \mathbf{x} \mathbf{A} \quad (13.1)$$

onde $\mathbf{x} = [\xi_1 \ \xi_2 \ \dots \ \xi_n]$ é o vetor das coordenadas afins e $\mathbf{A} = [\mathbf{v}_1 \ \mathbf{v}_2 \ \dots \ \mathbf{v}_n]^T$ são os vetores que formam a base. No espaço homogêneo, os pontos são representados pelas coordenadas afins e por mais uma coordenada que assume o valor zero quando representa um vetor e um quando representa um ponto do espaço projetivo.

Um ponto e um vetor podem ser definidos, em coordenadas no espaço homogêneo 2D, respectivamente:

$$\mathbf{P}_1 = \begin{Bmatrix} \xi_1 \\ \xi_2 \\ 1 \end{Bmatrix} \quad \text{e} \quad \mathbf{v}_1 = \begin{Bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ 0 \end{Bmatrix}$$

Seguindo a notação acima, pode-se definir a formulação paramétrica de um vetor a partir de 2 pontos, se forem consideradas apenas as coordenadas afins dos pontos, ou seja, desconsiderando o último termo da coordenada homogênea:

$$\mathbf{v}_1 = (1 - t)\mathbf{P}_o + t\mathbf{P}_1, \quad 0 \leq t \leq 1 \quad (13.2)$$

Caso sejam consideradas as coordenadas homogêneas dos pontos, a equação 13.2 pode representar um P_i ponto entre os pontos P_o e P_1 e pertencente ao vetor $\mathbf{v}_1$, apenas definindo o valor t .

$$\mathbf{P}_i = (1 - t)\mathbf{P}_o + t\mathbf{P}_1, \quad 0 \leq t \leq 1 \quad (13.3)$$

Um dos postulados da geometria define que 3 pontos não colineares no espaço definem um triângulo. Convertendo este postulado para a forma paramétrica, tem-se que:

$$\mathbf{P}_i = (1 - t - u)\mathbf{P}_o + t\mathbf{P}_1 + u\mathbf{P}_2, \quad 0 \leq t + u \leq 1 \quad (13.4)$$

onde o ponto P_i é um ponto pertencente ao triângulo formado por (P_o, P_1, P_2) . Essa forma paramétrica de se definir estas entidades caracteriza as figuras Simplex. Define-se que a figura Simplex 0-dimensional é um ponto, a 1-dimensional é uma reta, a 2-dimensional um triângulo e a 3-dimensional um tetraedro. É possível se abstrair as figuras Simplex para o caso n-dimensional. Neste trabalho, serão abordadas apenas figuras planas e, por isso, análises considerando modelos tridimensionais e entidades volumétricas não serão desenvolvidas.

Além de segmentos de retas e de vetores, as figuras planas podem apresentar curvas de variadas naturezas. As curvas podem ser arcos de circunferências ou polinômios interpolando pontos de controle. Em ferramentas de CAD, costumam-se utilizar as curvas parametrizadas e interpoladas por polinômios, como é o caso das curvas de Bézier, das B-Splines e das NURBS.

Uma curva pouco utilizada nas aplicações de edição gráfica é uma curva interpolação polinomial passante nos pontos. Nela, o número de nós n exige termos até grau $n - 1$:

$$P(u) = c_0 + c_1 u + c_2 u^2 + c_3 u^3 + \dots + c_{n-1} u^{n-1}, \quad 0 \leq u \leq 1 \quad (13.5)$$

onde $P(u)$ é ponto na curva, c_i são os vetores de coeficientes da curva a serem encontrados para cada coordenada, e $c_i = \begin{Bmatrix} c_{ix} \\ c_{iy} \\ c_{iz} \end{Bmatrix}$. Assim, assumindo que para cada valor de u a equação

paramétrica retorna um ponto pertencente ao polinômio, estipulam-se as condições dos pontos de controle tal que, para 4 pontos de controle tem-se

$$P(0) = P_0; \quad P\left(\frac{1}{3}\right) = P_1; \quad P\left(\frac{2}{3}\right) = P_2; \quad P(1) = P_3$$

É obtido, portanto, o sistema:

$$p = A \cdot c \rightarrow \begin{Bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1/3 & (1/3)^2 & (1/3)^3 \\ 1 & 2/3 & (2/3)^2 & (2/3)^3 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{Bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{Bmatrix} \quad (13.6)$$

Retomando que os coeficientes c_i 's são as incógnitas do sistema, a solução para a representação da curva será obtida por:

$$c = A^{-1} \cdot p \quad (13.7)$$

A interpolação dos pontos por curvas polinomiais pode ser insatisfatória caso se deseje criar curvas que se aproximam dos pontos, mas não sejam passantes. Ao assumir a distribuição dos pontos de controle como passos de $1/3$ em u , podem ocorrer inconsistências numéricas e distorções, dependendo do posicionamento destes pontos. Já o modelo de curva Bézier é largamente utilizado em ferramentas CAD e aplicações gráficas. Nele é aproximada uma cúbica e consideram-se 4 pontos de controle para fazer a aproximação da linha passando apenas por P_0 e P_3 , com $u = 0$ e $u = 1$, respectivamente. A Figura 13.7 mostra um exemplo da curva de Bézier e como as tangentes $\overrightarrow{P_0 P_1}$ e $\overrightarrow{P_2 P_3}$ controlam sua curvatura.

Figura 13.7 - Exemplo de curva de Bézier.

Fonte: (Angel & Shreiner, 2012)

Define-se, assim, que $P_0 = P(0)$ e $P_3 = P(1)$. As tangentes são aproximadas linearmente para

$$P'(0) \approx 3(P_1 - P_0) \quad (13.8)$$

$$P'(1) \approx 3(P_3 - P_2) \quad (13.9)$$

Derivando a equação da curva de cúbica (13.5), tem-se:

$$P'(u) = c_1 + 2uc_2 + 3u^2c_3 \quad (13.10)$$

Substituindo (13.10) em (13.8) e em (13.9):

$$3(P_1 - P_0) = c_1 \quad (13.11)$$

$$3(P_3 - P_2) = c_1 + 2c_2 + 3c_3 \quad (13.12)$$

Assim, após algumas manipulações, os coeficientes da curva são obtidos através do sistema:

$$c = \begin{Bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 3 & 0 & 0 \\ 1 & -2 & -5 & -8 \\ 1 & 1 & 1 & 1 \end{bmatrix}^{-1} \cdot \begin{Bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{Bmatrix} \quad (13.13)$$

Outra curva muito utilizada em programas CAD e em aplicações gráficas é a B-Spline, pois sua suavização é mais acentuada que a das curvas até agora citadas. No caso de uma curva cúbica, a definição de uma B-Spline é feita com 6 pontos de controle e calculando 2 curvas semelhantes à de Bézier, mas considerando 4 pontos de controle subsequentes e com 3 em comum. Supondo que se tem uma sequência de 6 pontos de controle, estes pontos são divididos

em 2 conjuntos de pontos formadores de 2 curvas $P(u)$ e $Q(u)$, que compartilham 3 destes pontos:

$$P_q = \begin{Bmatrix} P_{i-3} \\ P_{i-2} \\ P_{i-1} \\ P_i \end{Bmatrix}; \quad P_p = \begin{Bmatrix} P_{i-2} \\ P_{i-1} \\ P_i \\ P_{i+1} \end{Bmatrix} \quad (13.14)$$

Determinam-se as equações formadoras das curvas:

$$Q(u) = u^T \cdot M \cdot P_q \quad (13.15)$$

$$P(u) = u^T \cdot M \cdot P_p \quad (13.16)$$

São aproximadas as seguintes relações:

$$P(0) = Q(1) \approx \frac{1}{6}(P_{i-2} + 4P_{i-1} + P_i) \quad (13.17)$$

$$P'(0) = Q'(1) \approx \frac{1}{2}(P_i - P_{i-2}) \quad (13.18)$$

Com isso pode-se voltar para a equação da curva principal $P(u)$ considerando os coeficientes $P(u) = u^T \cdot c$

$$c_0 = \frac{1}{6}(P_{i-2} + 4P_{i-1} + P_i) \quad (13.19)$$

$$c_1 = \frac{1}{2}(P_i - P_{i-2}) \quad (13.20)$$

Por simetria, em $P(1)$:

$$P(1) = c_0 + c_1 + c_2 + c_3 = \frac{1}{6}(P_{i-1} + 4P_i + P_{i+1}) \quad (13.21)$$

$$P'(1) = c_1 + 2c_2 + 3c_3 = \frac{1}{2}(P_{i+1} - P_{i-1}) \quad (13.22)$$

Assim, após algumas manipulações, encontra-se a matriz M da curva:

$$M = \frac{1}{6} \begin{bmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \quad (13.23)$$

As curvas NURBS apareceram para flexibilizar as B-Splines e são largamente utilizadas em programas CAD atuais. Sua característica principal é aumentar ou diminuir a relevância de

cada ponto de controle com a inclusão de pesos aos pontos de controle. Ao se considerar os pesos, os pontos têm sua posição alterada, tornando a curva maleável ponto a ponto.

$$P_i = w_i \begin{bmatrix} x(u)_i \\ y(u)_i \\ z(u)_i \end{bmatrix} \quad (13.24)$$

Modelos geométricos 2D e 3D podem conter uma grande quantidade de informações dependendo da complexidade das formas geométricas que se deseja representar com a computação gráfica. Os sistemas CAD/CAM que proporcionaram a manipulação destas informações começaram a ser consolidados e largamente utilizados em meados da década de 80, possibilitando o desenvolvimento gráfico de modelos 2D e 3D a nível comercial, com pacotes de ferramentas computacionais sendo aprimorados até os dias atuais.

Até a década de 70, os sistemas existentes apenas permitiam a construção de modelos *wireframe*, que são representações de figuras planas e tridimensionais através de seus contornos, compostos de vértices e arestas. Este tipo estrutura de dados se tornava limitada quando o modelo necessitava de informações referentes às superfícies, como se pode observar no modelo da Figura 13.8. Representar objetos 3D apenas por seus contornos pode resultar em visualizações pouco claras ou até a visualização de objetos impossíveis de se construir.

Figura 13.8 - Modelo de peça representada por *wireframe*.

Fonte: (Stroud & Nagy, 2011)

O conceito de estruturação dos dados como CSG (*Constructive Solid Geometry*) foi criado paralelamente por vários pesquisadores diferentes no final da década de 1970. Com isso foi possibilitada a representação de sólidos por partes menores compostas de primitivas e superfícies, além de armazenar informações da sua composição construtiva. O sólido se torna uma composição de outros conjuntos de primitivas associados pelas operações booleanas OR, AND e INTERSECT, gerando árvores de operações. A Figura 13.9 mostra um esquema de árvore de montagem de um modelo sólido.

Figura 13.9 - Representação de um modelo CSG com sua árvore de operações booleanas.

Fonte: (Stroud & Nagy, 2011).

A estruturação de dados do tipo CSG foi largamente utilizada em programas de CAD, mas foi substituída, até os dias atuais, pela estrutura BRep (*Boundary Representation*). Esse modelo organiza as entidades geométricas em dois conjuntos: a topologia, que armazena conjuntos de entidades e a geometria, que lista as entidades geométricas. Um exemplo ilustrativo é o caso de uma superfície, onde os *loops* das fronteiras e suas bordas formadoras estão contidas no conjunto topologia, enquanto as curvas (segmentos de retas e curvas) e a superfície estão contidas no grupo da geometria. A Figura 13.10 mostra um diagrama destes conjuntos e como as entidades se relacionam.

Figura 13.10 - Diagrama dos conjuntos de entidades da estrutura Brep.

Fonte: próprio autor do texto.

A estrutura BRep apresenta várias vantagens na manipulação das informações geométricas dos modelos, sendo adotada na grande maioria dos programas de CAD atuais. Um exemplo de ganho do modelo BRep sobre o CSG é no acesso à detalhes de geometria. Obter as características de uma superfície com furos se torna uma tarefa mais direta com a estrutura BRep, enquanto na estrutura CSG deve-se considerar o conjunto de operações que formam esta superfície. No presente trabalho foi usada a estrutura BRep para representar as figuras planas, consideradas como faces, pois promove o fácil acesso às propriedades das fronteiras, tanto internas quanto externas.

D. Tipos de Malhas (classificação dos elementos e classificação das malhas)

Malha é a divisão de um domínio geométrico em formas menores e mais simples, como triângulos e quadriláteros em um espaço bidimensional, bem como tetraedros e hexaedros em um espaço tridimensional. Malhas são largamente utilizadas em várias áreas assistidas por computadores: Na geografia e cartografia, malhas são criadas para representar a características dos relevos; na computação gráfica, os objetos e superfícies devem ter suas geometrias aproximadas para malhas para que possam ser processadas nas placas gráficas; também se faz essencial o uso de malhas nas simulações físicas para resolver numericamente sistemas de equações diferenciais parciais. O presente trabalho é focado em malhas com finalidade de

simulação física de carregamentos em estruturas bidimensionais pelo Método dos Elementos Finitos.

A criação de uma malha inicia-se com a definição de uma geometria de referência, ou seja, o domínio geométrico. No caso do escopo deste trabalho, apenas figuras planas e malhas bidimensionais foram analisadas. O modelo geométrico plano computacional é o dado de entrada para a geração de malha e deve ser carregado a partir de um arquivo do tipo texto, conforme foi estipulado para este trabalho.

A composição das malhas é feita basicamente por pontos (nós) e segmentos de retas que os interligam, formando triângulos ou quadriláteros. A configuração da conexão dos nós com os segmentos de retas possui algumas propriedades, conforme exposto por Bommes, et al. (2012). A propriedade de valência de um nó é o número de segmentos de retas que incidem sobre ele. Define-se também o termo estrela de um nó como o conjunto de elementos que se conectam nele. Os segmentos que pertencem apenas a um elemento, são denominados segmentos de fronteira, já os segmentos que são compartilhados por dois elementos são chamados de segmentos internos.

Os tipos de malhas são comumente caracterizados conforme os padrões descritos, por exemplo, em Sack e Urrutia (1999):

- **Estruturada ou Regular**

A malha é dita estruturada quando todos seus nós são topologicamente padronizados. A malha se assemelha a uma grade retangular que foi mapeada para a figura geométrica, no caso de elementos retangulares, e todos os nós deverão apresentar valência igual a 2 ou 4.

Entende-se o termo valência de um nó como o número de ligações que este faz com outros nós em seu entorno. A Figura 13.11.a mostra um exemplo de uma malha estruturada

- **Semi-Estruturada ou Semi-Regular**

Este tipo de malha é bastante variado. Pode ser considerada como um conjunto de sub-regiões de malhas estruturadas acopladas entre si formando a figura total. No artigo de Bommes, et al. (2012) este tipo de malha é dividido em 2 para diferenciar as malhas semi-estruturadas que tem a maioria dos nós de valência 4 e as que não apresentam esta característica. Neste estudo não foi considerada esta distinção por não haver motivo relevante observado nos métodos geradores de malhas aqui avaliados. A Figura 13.11.c mostra um exemplo deste tipo de malha.

- **Não-Estruturada**

Este é o caso da malha apresentar nós irregulares. Se a malha for composta por quadriláteros, um nó irregular é aquele de valência 3 ou mais de 4.

Figura 13.11 - a) Exemplo de malha estruturada; b) Exemplo da malha irregular; c) Exemplo de malha semi-estruturada.

Fonte: (Sack & Urrutia, 1999).

As malhas podem ser avaliadas quanto à qualidade dos seus elementos. Triângulos são avaliados a partir da sua proximidade a triângulos equiláteros e é verificado o seu maior ângulo interno. Em malhas com quadriláteros, a qualidade dos elementos é medida verificando a proximidade dos ângulos internos com 90° e verificando se o tamanho das arestas opostas é próximo.