



PUC
RIO

ANTONIO CARLOS DE OLIVEIRA MIRANDA

*INTEGRAÇÃO DE ALGORITMOS DE GERAÇÃO DE MALHAS
DE ELEMENTOS FINITOS*

DISSERTAÇÃO DE MESTRADO

Departamento de Engenharia Civil
PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

Rio de Janeiro, Fevereiro de 1999

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO

Rua Marquês de São Vicente, 225 - Gávea
CEP 22453-900 Rio de Janeiro RJ Brasil
<http://www.puc-rio.br>

N.Cham. 624 M672i TESE UC

Título Integração de algoritmos de geração de malhas de element



Ex.1 PUCB

0135731

Antonio Carlos de Oliveira Miranda

Integração de Algoritmos de Geração de Malhas de Elementos Finitos

Tese apresentada ao Departamento de Engenharia Civil
da PUC-Rio como parte dos requisitos para obtenção do
título de Mestre em Ciências em Engenharia Civil:
Estruturas

Orientador: Luiz Fernando C. R. Martha

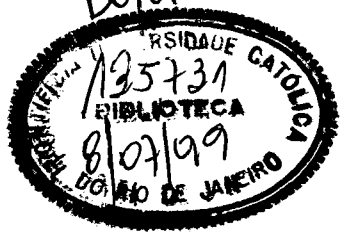
Departamento de Engenharia Civil

Pontifícia Universidade Católica do Rio de Janeiro

Rio de Janeiro, 25 de fevereiro de 1999

92866

B/d



624
M672i
TESE UC

À minha família.

Agradecimentos

Aos meus pais, pelo apoio, incentivo e confiança em todos os momentos da minha vida. Aos meus irmãos, pelo apoio e amizade.

Ao professor Luiz Fernando Martha pela orientação, incentivo, amizade e confiança durante o desenvolvimento deste trabalho.

A todas as pessoas que contribuíram de maneira direta e indireta para realização deste trabalho. Em especial, ao Joaquim Bento Cavalcante Neto pelos esclarecimentos e discussões sobre os algoritmos bidimensionais e tridimensionais; Luiz Cristovão Gomes Coelho, pelas referências e auxílio na implementação dos algoritmos de geração de malhas tridimensionais; William Vagner, pelo auxílio na implementação dos algoritmos de geração de malhas bidimensionais; Carlos Vitor de Alencar Carvalho, pela ajuda no sistema UNIX.

Aos professores do Departamento de Engenharia Civil da PUC-Rio, por estarem sempre disponíveis ao auxílio.

Aos amigos Chan, Sidiclei e Gilmar, pela amizade e pelos bons momentos vividos juntos.

Aos amigos do Pará: Devaney, Luciana, Salete, Ricardo, Alexandre, André Clementino, Iveline, Marcelo e Gerson.

Aos amigos da PUC: Aurea, Evandro, Claudinha, Ricardo Issa, Cristina, Glauco, Americo e Mario.

A todos amigos do Tecgraf.

À Capes pelo apoio financeiro durante o curso de mestrado.

Resumo

Este trabalho apresenta algoritmos de geração automática de malhas de elementos finitos bidimensionais e tridimensionais que, juntamente com outros algoritmos desenvolvidos dentro da linha de pesquisa ao qual ele está inserido, são integrados consistentemente em um pacote computacional. Os algoritmos geradores de malhas foram modificados para serem independentes dos pré-processadores, podendo assim serem incorporados em diferentes programas. Um novo algoritmo bidimensional de geração de elementos triangulares é proposto. Este algoritmo incorpora novas idéias, adaptadas de um algoritmo de geração tridimensional de malhas em volume arbitrário, e procura gerar elementos de melhor qualidade geométrica possível. Duas técnicas de geração de malhas sólidas são apresentadas: uma de mapeamento transfinito tridimensional, onde a malha sólida é construída a partir das seções transversais do modelo; e uma técnica de *sweep*, onde a malha sólida é construída pelo “arrasto” de uma seção transversal ao longo de um curva no espaço.

Os algoritmos são incorporados em um modelador geométrico tridimensional, o MG (*Mesh Generator*), e alguns exemplos são feitos para demonstrar a capacidade de modelagem do ambiente criado. Em particular são mostrados exemplos para modelagem composta com diferentes métodos de construção de malhas e diferentes tipos de elementos finitos em um mesmo modelo.

Também é feito um estudo de algumas medidas de distorção para elementos finitos planos. O cálculo dessas medidas de distorção é implementado em um modelador bidimensional gráfico interativo. A visualização das medidas de distorção é feita através de uma escala de cores.

Abstract

This work presents algorithms for two-dimensional and three-dimensional mesh generation of finite elements that, together with another algorithms developed by its research group, are integrated in a computational package. The mesh generation algorithms become independent of the preprocessor, and may be incorporated in different programs. An algorithm for two-dimensional planar triangular mesh generation is proposed. The algorithm incorporates new ideas, adapted from a three-dimensional mesh generation algorithm for arbitrary domains, which is focused on the generation of "optimal" shape elements. Two techniques of solid mesh generation are presented: a three-dimensional transfinite mapping, that generates the solid mesh interpolating cross-section meshes; and a sweep technique, that generates the solid mesh by sweeping a cross-section mesh along a curve in space.

The algorithms are integrated in a three-dimensional geometric modeler, MG (Mesh Generation), and some examples demonstrate the modeling capability of the created environment. Specifically, the examples show the capability for composite modeling with different methods of mesh generation and different types of finite elements in the same model.

A study of some distortion measures for planar finite elements is also performed. Computation of these distortion measures is implemented in a two-dimensional interactive graphics modeler. The visualization of the distortion measures is made through a color scale.

Sumário

1.	INTRODUÇÃO.....	1
1.1	OBJETIVOS DO TRABALHO	5
1.2	GERAÇÃO DE MALHAS E TRABALHOS RELACIONADOS	3
1.3	ORGANIZAÇÃO DO TRABALHO	6
2.	MEDIDAS DE DISTORÇÃO PARA ELEMENTOS FINITOS PLANOS	7
2.1	OBTENÇÃO DAS MEDIDAS DE DISTORÇÃO	8
2.1.1	<i>Medidas de Distorção do Elemento Quadrilateral de 4 Nós</i>	10
2.1.2	<i>Medidas de Distorção do Elemento Quadrilateral de 8 Nós</i>	14
2.1.3	<i>Medidas de Distorção do Elemento Triangular de 3 Nós</i>	18
2.1.4	<i>Medidas de Distorção do Elemento Triangular de 6 Nós</i>	20
2.2	REPRESENTAÇÃO DAS MEDIDAS DE DISTORÇÃO.....	20
2.3	IMPLEMENTAÇÃO E EXEMPLOS	25
3.	ALGORITMO DE TRIANGULAÇÃO ARBITRÁRIA COM CONSIDERAÇÃO DE TRINCAS	28
3.1	DESCRIÇÃO DO ALGORITMO	29
3.1.1	<i>Geração da Quadtree</i>	30
3.1.1.1	Geração da Quadtree Baseada na Discretização do Contorno	30
3.1.1.2	Refinamento da Quadtree para Forçar um Tamanho de Célula Máximo	31
3.1.1.3	Refinamento da Quadtree para Forçar Disparidade de Tamanho Mínima	31
3.1.2	<i>Procedimento de Avanço da Fronteira</i>	33
3.1.2.1	Geração de Elementos baseada em Geometria.....	33
3.1.2.1.1	Lista de Contração de Contorno.....	34
3.1.2.1.2	Geração de Elementos Ótimos	34
3.1.2.2	Geração de Elementos Baseada em Topologia	37
3.1.3	<i>Melhoria Local da Malha</i>	37
3.1.3.1	Suavização Laplaciana.....	39
3.1.3.2	Avaliação da Qualidade e Procedimento de “volta-passo” Local	39
3.2	COMPARAÇÃO COM OUTROS ALGORITMOS BASEADOS EM QUADTREE	41
4.	MALHAS ESTRUTURADAS EM 3D	50
4.1	MAPEAMENTO TRANSFINITO TRIDIMENSIONAL	51
4.1.1	<i>Reconhecimento dos Pontos entre Seções Transversais</i>	52
4.1.1.1	Escolha do Elemento Inicial	53
4.1.1.2	Construção da Lista de Arestas.....	54
4.1.1.3	Obtenção dos Pontos a partir da Lista de Arestas	55
4.1.2	<i>Interpolação e Exemplo</i>	56
4.2	TÉCNICA DE SWEEP.....	57
4.2.1	<i>Obtenção da Orientação de cada Seção Intermediária</i>	58
4.2.2	<i>Transformação dos Pontos da Seção Transversal para o Sistema Local de Coordenadas</i>	59
4.2.3	<i>Cálculo dos Pontos da Malha Sólida</i>	60
5.	MODELAGEM TRIDIMENSIONAL DE ELEMENTOS FINITOS	63
5.1	MODELAGEM DE MALHAS SÓLIDAS	64
5.1.1	<i>Extrusão</i>	66
5.1.2	<i>Sweep Curvo</i>	66
5.1.3	<i>Mapeamento Transfinito Tridimensional</i>	67
5.1.4	<i>Malha Volumétrica Genérica</i>	68
5.2	EXEMPLO DE MODELAGEM COMPOSTA	73
5.2.1	<i>Corpo-de-Prova</i>	73
5.2.2	<i>Meso-Estrutura</i>	74
6.	CONCLUSÃO.....	79
6.1	PRINCIPAIS CONTRIBUIÇÕES	79

6.2	SUGESTÕES PARA TRABALHOS FUTUROS	81
	APÊNDICE A - SENSIBILIDADE DA MÉTRICA γ/γ^* EM TRIÂNGULOS	82
	APÊNDICE B - BIBLIOTECA COMPUTACIONAL PARA GERAÇÃO DE MALHAS DE ELEMENTOS FINITOS	84
	REFERÊNCIAS BIBLIOGRÁFICAS	92

Lista de Figuras

FIGURA 2.1 - ELEMENTO FINITO QUADRILATERAL PLANO.	8
FIGURA 2.2 - SIGNIFICADO GEOMÉTRICO DAS MEDIDAS DE DISTORÇÃO.	9
FIGURA 2.3 - DIVERGÊNCIAS NORMAL E TANGENCIAL EM BORDO CURVO.	9
FIGURA 2.4 - SIGNIFICADO GEOMÉTRICO DOS COEFICIENTES E E F	12
FIGURA 2.5 - COEFICIENTES E E F NO SISTEMA LOCAL DE COORDENADAS.	13
FIGURA 2.6 - POSSIBILIDADES DO EIXO LOCAL.	13
FIGURA 2.7 - QUADRILATERAL COM BORDO CURVO.	14
FIGURA 2.8 - INTERPRETAÇÃO GEOMÉTRICA DOS COEFICIENTES $E_5 - E_8$ E $F_5 - F_8$	16
FIGURA 2.9 - RETÂNGULO DA RAZÃO NO QUADRILATERAL.	17
FIGURA 2.10 - VETORES NO BORDO CURVO.	18
FIGURA 2.11 - MEDIDAS DE DISTORÇÃO PARA O ELEMENTO TRIANGULAR PLANO.	18
FIGURA 2.12 - RAZÃO IDEAL PARA O ELEMENTO TRIANGULAR.	19
FIGURA 2.13 - ELEMENTO TRIANGULAR PLANO DE 6 NÓS.	20
FIGURA 2.14 - EXEMPLO DE UMA ESCALA DE CORES.	21
FIGURA 2.15 - TESTES DE SENSIBILIDADE DAS MEDIDAS DE DISTORÇÃO NOS ELEMENTOS FINITOS PLANOS.	22
FIGURA 2.16 - SENSIBILIDADE PARA O ELEMENTO FINITO QUADRILATERAL - TESTE (A).	22
FIGURA 2.17 - SENSIBILIDADE PARA O ELEMENTO FINITO TRIANGULAR - TESTE (B).	23
FIGURA 2.18 - SENSIBILIDADE PARA O ELEMENTO FINITO TRIANGULAR - TESTE (C).	23
FIGURA 2.19 - LISTA <i>DROP-DOWN</i> PARA ESCOLHA DE REPRESENTAÇÃO DA MEDIDA DE DISTORÇÃO.	25
FIGURA 2.20 - EXEMPLO DA IMPLEMENTAÇÃO COM A MEDIDA DE DISTORÇÃO OBLIQUIDADE.	26
FIGURA 2.21 - EXEMPLO DA IMPLEMENTAÇÃO COM A MEDIDA DE DISTORÇÃO RAZÃO.	26
FIGURA 2.22 - EXEMPLO DA IMPLEMENTAÇÃO COM A MEDIDA DE DISTORÇÃO ESTREITAMENTO.	27
FIGURA 2.23 - EXEMPLO DA IMPLEMENTAÇÃO COM A MEDIDA DE DISTORÇÃO DIVERGÊNCIA NORMAL.	27
FIGURA 3.1 - MODELO BIDIMENSIONAL E SEU REFINAMENTO NO CONTORNO.	31
FIGURA 3.2 - <i>QUADTREE</i> INICIAL DO MODELO.	32
FIGURA 3.3 - <i>QUADTREE</i> APÓS FORÇAR MAIOR TAMANHO DE CÉLULA DO CONTORNO.	32
FIGURA 3.4 - <i>QUADTREE</i> APÓS FORÇAR DISPARIDADE DE TAMANHO MÍNIMO ENTRE CÉLULAS.	33
FIGURA 3.5 - REGIÃO ÓTIMA DE VÉRTICE CANDIDATO DE UM TRIÂNGULO.	35
FIGURA 3.6 - DEFINIÇÃO DO ÂNGULO PARA UM VÉRTICE.	35
FIGURA 3.7 - SELEÇÃO DE UM NÓ CANDIDATO EM UMA TRINCA.	36
FIGURA 3.8 - MALHA GERADA APÓS PROCEDIMENTO DE AVANÇO DA FRONTEIRA.	38
FIGURA 3.9 - EXEMPLO DO PROCEDIMENTO "VOLTA-PASSO".	40
FIGURA 3.10 - MALHA GERADA APÓS MELHORIA LOCAL DA MALHA.	41
FIGURA 3.11 - MODELOS: (A) ARQUIBANCADA E (B) PÓRTICO.	43
FIGURA 3.12 - MALHA TRIANGULAR GERADA PARA O EXEMPLO DE UM QUADRADO.	44
FIGURA 3.13 - MALHA TRIANGULAR GERADA PARA O EXEMPLO DE UM CÍRCULO.	44
FIGURA 3.14 - MALHA TRIANGULAR GERADA PARA O EXEMPLO DE UM ARQUIBANCADA.	45
FIGURA 3.15 - MALHA TRIANGULAR GERADA PARA O EXEMPLO DE UM PÓRTICO.	45
FIGURA 3.16 - HISTOGRAMA DE ANÁLISE DE FORMA PARA O EXEMPLO DE UM QUADRADO.	46
FIGURA 3.17 - HISTOGRAMA DE ANÁLISE DE FORMA PARA O EXEMPLO DE UM CÍRCULO.	46
FIGURA 3.18 - HISTOGRAMA DE ANÁLISE DE FORMA PARA O EXEMPLO DE UMA ARQUIBANCADA.	47
FIGURA 3.19 - HISTOGRAMA DE ANÁLISE DE FORMA PARA O EXEMPLO DE UM PÓRTICO.	47
FIGURA 3.20 - MALHA GERADA PELO ALGORITMO DE VIANNA COM REPRESENTAÇÃO DA DISTORÇÃO RAZÃO.	48
FIGURA 3.21 - MALHA GERADA PELO ALGORITMO PROPOSTO COM REPRESENTAÇÃO DA DISTORÇÃO RAZÃO.	48
FIGURA 3.22 - MALHA GERADA PELO ALGORITMO DE CAVALCANTE NETO COM REPRESENTAÇÃO DA DISTORÇÃO RAZÃO.	49
FIGURA 4.1 - SEÇÕES TRANSVERSAIS DE UM MODELO SÓLIDO.	52
FIGURA 4.2 - ESCOLHA DO ELEMENTO INICIAL PARA LISTA DE ARESTAS.	53
FIGURA 4.3 - PONTOS BASE SELECIONADOS EM POSIÇÕES INCOERENTES.	54
FIGURA 4.4 - LISTA DE ARESTAS PARA O ELEMENTO INICIAL.	54
FIGURA 4.5 - EXEMPLO PARA CONSTRUÇÃO DA LISTA DE ARESTAS.	55
FIGURA 4.6 - SEÇÕES TRANSVERSAIS DE UM EXEMPLO PARA GERAR ELEMENTOS ATRAVÉS DE <i>MTT</i>	56
FIGURA 4.7 - MODELO GERADO A PARTIR DE SUAS SEÇÕES TRANSVERSAIS.	57
FIGURA 4.8 - <i>SWEEP</i> DEFINIDO POR UMA SEÇÃO TRANSVERSAL E UMA TRAJETÓRIA NO ESPAÇO.	58

FIGURA 4.9 - PONTOS BASES E EIXOS ORIENTAÇÃO DAS SEÇÕES TRANSVERSAIS.	58
FIGURA 4.10 - PONTOS GERADOS DO MODELO SÓLIDO.	61
FIGURA 4.11 - MODELO GERADO COM <i>SWEEP</i>	61
FIGURA 4.12 - SEÇÃO TRANSVERSAL E TRAJETÓRIA DO <i>SWEEP</i> PARA UM MODELO DE UMA MOLA.	62
FIGURA 4.13 - MODELO DE UMA MOLA GERADA POR <i>SWEEP</i>	62
FIGURA 5.1 - MÓDULOS DA ESTRUTURA DE DADOS DO MG.	65
FIGURA 5.2 - CHUMBADOR ROTULADOR CONSTRUÍDO ATRAVÉS DE EXTRUSÃO.	69
FIGURA 5.3 - SUPERFÍCIES QUE GERAM O CHUMBADOR ROTULADOR.	70
FIGURA 5.4 - BOCAL CONSTRUÍDO ATRAVÉS DE <i>SWEEP</i> CURVO.	70
FIGURA 5.5 - SUPERFÍCIE E CAMINHO DO <i>SWEEP</i> PARA O BOCAL.	71
FIGURA 5.6 - GANCHO CONSTRUÍDO POR MAPEAMENTO TRANSFINITO TRIDIMENSIONAL.	71
FIGURA 5.7 - SEÇÕES TRANSVERSAIS DO GANCHO.	72
FIGURA 5.8 - MALHA VOLUMÉTRICA EM TETRAEDROS DE UM FECHO DE PORTA DE AVIÃO.	72
FIGURA 5.9 - SUPERFÍCIE DE CONTORNO DO FECHO DE PORTA DE AVIÃO.	73
FIGURA 5.10 - MODELO DE UM CORPO-DE-PROVA.	75
FIGURA 5.11 - SUPERFÍCIES BASES PARA O CORPO-DE-PROVA.	75
FIGURA 5.12 - MALHAS GERADAS PARA O CORPO-DE-PROVA.	76
FIGURA 5.13 - CAMPOS DE TENSÕES NORMAIS À SEÇÃO TRANSVERSAL DO CORPO-DE-PROVA.	76
FIGURA 5.14 - MODELO DE UMA MESO-ESTRUTURA DE PASSARELA.	77
FIGURA 5.15 - SUPERFÍCIES UTILIZADAS PARA GERAR A MESO-ESTRUTURA.	77
FIGURA 5.16 - MALHAS GERADAS PARA A MESO-ESTRUTURA.	78
FIGURA 5.17 - CAMPOS DE TENSÕES NA MESO-ESTRUTURA.	78
FIGURA A.1- SENSIBILIDADE - TESTE (A).	82
FIGURA A.2 - SENSIBILIDADE - TESTE (B).	83
FIGURA B.1 - VETORES DE COORDENADAS E CONECTIVIDADE.	84

1 Introdução

O método dos elementos finitos é um método numérico largamente utilizado para simular algumas classes de problemas de engenharia em que a solução analítica é desconhecida ou difícil de se obter. Juntamente com a utilização dos computadores, o método tornou-se um dos mais utilizados nos dias de hoje.

Em meados dos anos 50, quando o método aparece nas análises de engenharia, cabia ao engenheiro analista a tarefa de descrever a malha, através de arquivos texto, diretamente para os programas de análise existentes na época. Um dos primeiros sistemas de uso geral foi o ICES (Integrated Civil Engineering System), desenvolvido no MIT (Massachusetts Institute of Technology).

A década seguinte trouxe o crescimento de pesquisa sobre o método, com o desenvolvimento de novos elementos e técnicas numéricas mais eficientes. O surgimento do programa SAP (Structural Analysis Program), desenvolvido em Berkeley em 1962 e cujo código fonte foi distribuído pelo mundo, influenciou de modo marcante o desenvolvimento do método.

Atualmente, utiliza-se programas gráficos de pré e pós-processamento para geração de malhas e análise de resultados. A descrição da malha através de arquivos textos foi substituída pela definição geométrica do modelo dentro de um ambiente gráfico interativo, permitindo inserir todas as informações necessárias para análise, como geometria do modelo, condições de contorno e atributos. A visualização dos resultados feita através da consulta a inúmeras folhas de listagem foi substituída pela visualização gráfica, onde os resultados podem ser apresentados sob diversas modos de representação gráfica.

Dentro dos pré-processadores gráficos, utiliza-se algoritmos que geram malhas de elementos finitos automaticamente de acordo com uma técnica de geração escolhida pelo usuário. Uma destas técnicas é a triangulação arbitrária. Nessa técnica, em geral, é dado o contorno do modelo e o algoritmo é capaz de realizar a triangulação de acordo com suas características específicas. Existe uma grande variedade de algoritmos de triangulação na literatura, cada um com características diferentes e outros que participam de uma mesma idéia. Pode-se concluir que a geração de malhas de elementos finitos bidimensionais está resolvida quanto à capacidade de gerar malhas válidas. Por outro lado é desejável que os algoritmos se tornem suficientemente confiáveis para atenderem às diversas necessidades dos

projetos de engenharia. Dentro destas necessidades pode-se incluir a melhor qualidade dos elementos, a capacidade de reconhecimento de geometria mais complexas e malhas que evoluem de acordo com as mudanças ocorridas no modelo. A qualidade da forma geométrica dos elementos, principalmente em análises da mecânica estrutural, é um dos fatores que influenciam na acurácia dos resultados obtidos. Neste sentido, é necessário que os algoritmos de triangulação permitam gerar os melhores elementos possíveis para minimizar a possibilidade de erros nos resultados. Também os pré-processadores devem oferecer a capacidade de visualizar e/ou criticar a forma dos elementos para que o usuário possa modificar de modo conveniente a malha gerada.

Outro aspecto importante está relacionado com a alteração da geometria do problema. Muitas vezes a geometria muda com a evolução do problema de análise, como no caso de propagação de trincas¹, e o algoritmo deve ser capaz de redefinir a malha de modo rápido obedecendo as novas restrições impostas pela geometria.

Outra característica importante é a capacidade de fazer uma transição entre elementos com tamanhos com ordem de grandeza diferente. A discretização do problema pode apresentar tamanhos de elementos que variam consideravelmente e o algoritmo de triangulação deve promover uma transição suave, gerando elementos com boa qualidade de forma geométrica.

Uma evolução natural da área é a geração de malhas tridimensionais. A análise bidimensional é uma simplificação de modelos tridimensionais, levada muitas vezes pela dificuldade de se criar uma representação discreta adequada para descrever a geometria e a topologia do modelo tridimensional, como também pela necessidade de redução do número de incógnitas no sistema de equações que rege o método dos elementos finitos. Percebe-se que os problemas na geração de malhas tridimensionais se multiplicam quando comparados ao caso bidimensional. Estes problemas estão relacionados principalmente com a dificuldade de gerar malhas sólidas e com a entrada de dados para um algoritmo. Um pré-processador tridimensional deve ser suficientemente versátil para permitir uma interação eficiente com o usuário e rapidez na obtenção de respostas necessárias. Além disso, deve ter código flexível a alterações e inclusões de novos métodos e aos modos de como as malhas são geradas, permitindo a sua expansão natural. O uso intensivo da Computação Gráfica e da Modelagem Geométrica permite que as dificuldades de obtenção dos dados sejam minimizados. A flexibilidade do programa pode ser obtida com técnicas de estruturação de dados e técnicas de

¹ Trincas podem ser consideradas como discontinuidades no domínio do corpo. São provocadas por falhas causadas por solicitações cíclicas, choque ou defeitos de fabricação.

programação orientada a objetos, por exemplo.

Apesar de toda a sofisticação conferida aos pré e pós-processadores existentes atualmente, estes estão em contínua evolução devida às mudanças ocasionadas pelos aperfeiçoamentos de *software* e *hardware*. Em certos casos, o algoritmo que gera a malha de elementos finitos está intrinsecamente ligado a um pré-processador. Isto é, para utilizar um algoritmo que permita gerar malhas sólidas é necessário informar os dados de entrada e estes estão relacionados com o pré-processador utilizado. Assim, ocorrendo tais mudanças é possível que os códigos fontes do algoritmo de geração de malhas tenham que ser modificados. Isto é indesejável. Logo, tem-se de adotar estratégias para que os algoritmos de malhas tenham módulos independentes dos pré-processadores. Essas estratégias devem permitir que os algoritmos de geração de malhas sejam utilizados em diferentes pré-processadores sem que haja a necessidade de reescrever seu código fonte.

Existe então a necessidade de criação de um conjunto de algoritmos para a geração de malhas que interajam de modo prático com os modeladores geométricos existentes e que permitam alterações rápidas para se acomodarem com novas tecnologias. Além disso, a qualidade geométrica dos elementos que compõem a malha gerada deve ser objeto de preocupação de todos os algoritmos.

1.1 Geração de Malhas e Trabalhos Relacionados

As malhas de elementos finitos podem ser classificadas como malhas estruturadas e malhas não-estruturadas. Malhas não-estruturadas não têm nenhum tipo de regularidade ou periodicidade, diferente de malhas estruturadas, onde existe alguma regularidade local ou global em seu arranjo topológico.

De uma forma geral, dentre as técnicas de geração de malhas estruturadas para o caso bidimensional, destacam-se os mapeamentos transfinitos (Haber, 1981; Fonseca, 1989). Dentre os algoritmos de geração de malhas não-estruturadas, destacam-se os métodos baseados em enumeração espacial recursiva (Baehamann *et al.*, 1987; Cavalcante Neto, 1994), em critérios de triangulação de Delaunay (Joe, 1986; Chew, 1989; Watson, 1981; de Floriani e Puppo, 1992) e em métodos de avanço da fronteira (Lo, 1989; Potyondy, 1993).

Na área de geração de malhas volumétricas existem vários algoritmos na literatura. Dentre as diferentes técnicas não-estruturadas usadas, existem algoritmos baseados em triangulação de Delaunay (Watson, 1981; Field, 1986; Baker, 1987; Joe, 1990), em

decomposição espacial recursiva (Yerry e Shephard, 1984; Perucchio et al., 1989, Shephard e Georges, 1991) e em métodos de avanço da fronteira (Peraire et al., 1988; Lohner e Parikh, 1988; Moller e Hansbo, 1995). Em relação às técnicas de geração de malhas estruturadas, existem, entre outros, os algoritmos de Mapeamento Transfinito Tridimensional (Perucchio, 1982; Fonseca, 1989) e os baseados na técnica do *sweep* (George, 1991).

A linha de pesquisa da Computação Gráfica Aplicada do Departamento de Engenharia Civil da PUC-Rio desenvolve há vários anos programas de análise de elementos finitos (FEMOOP - Martha *et al.*, 1996), juntamente com pré-processadores gráficos bidimensionais (MTool, 1992) e tridimensionais (MG - Gomes Coelho, 1998; EGM TT - Fonseca, 1989), e pós-processadores bidimensionais (MView, 1997) e tridimensionais (POS3D - Celes Filho *et al.*, 1991).

Faz-se aqui um breve histórico da linha de pesquisa na área de geração de malhas. O trabalho de Paulino (1988) apresenta o desenvolvimento, implementação e funcionamento de um programa interativo para geração de dados de modelos de elementos finitos de estruturas reticuladas espaciais. Também expõe uma proposta de metodologia para elaboração da arquitetura do programa, os ambientes de desenvolvimento, interface e estrutura de dados. Na mesma filosofia de trabalho de Paulino, Fonseca (1989) apresenta um pré-processador de malhas transfinitas tridimensionais de elementos finitos que são criadas a partir de seções transversais do modelo.

O trabalho de Campos (1991) aborda a geração de malhas de elementos finitos bidimensionais baseada em uma estrutura de dados topológica, consistindo de um pré-processador gráfico interativo para discretização numérica de modelos bidimensionais. O trabalho de Vianna (1992), que aborda a modelagem geométrica estendida para modelos bidimensionais, juntamente com o trabalho de Guimarães (1992) compõem um sistema integrado para problemas de elasticidade plana. Vianna também aborda um gerador automático de malhas bidimensionais baseado na técnica de avanço de fronteira.

Um outro trabalho, de Cavalcante Neto (1994), descreve um ambiente computacional gráfico interativo para análise integrada auto-adaptativa de modelos bidimensionais de elementos finitos, com aplicação em problemas estruturais mecânicos, baseado em uma técnica de enumeração espacial recursiva. Um gerador de malhas bidimensionais foi desenvolvido, combinando a técnica de *quadtree* (Baehamann *et al.*, 1987) e a técnica de triangulação por contração do contorno. O algoritmo gera os elementos no interior da região por *quadtree* e na faixa na fronteira da região por triangulação.

No que se refere à modelagem tridimensional, como mencionado, o trabalho de Fonseca (1989) descreve um pré-processador gráfico interativo para geração de modelos de elementos finitos sólidos através do mapeamento transfinito tridimensional. Gomes Coelho (1998) desenvolveu em seu trabalho o MG (*Mesh Generator*): um modelador de malhas de superfícies com geometria arbitrária, com enfoque na interseção entre retalhos de malhas de superfície distintas. Em um trabalho mais recente, Calvalcante Neto (1998) descreve uma técnica de geração de malhas volumétricas de elementos tetraédricos para domínios tridimensionais arbitrários. Até o presente trabalho, esse algoritmo ainda não estava incorporado nos modeladores geométricos para geração de malhas da linha de pesquisa.

1.2 Objetivos do Trabalho

O objetivo deste trabalho é integrar consistentemente os algoritmos de geração de malhas de elementos finitos desenvolvidos na linha de pesquisa, e complementá-los com outros algoritmos de geração de malhas bidimensionais e tridimensionais desenvolvidos a partir de novas idéias, criando uma API² consistente entre os algoritmos e os modeladores existentes. Deste trabalho resulta a criação de uma biblioteca computacional englobando os algoritmos existentes e os novos algoritmos desenvolvidos. Essa biblioteca proporcionará a incorporação dos algoritmos em outros modeladores existentes, principalmente tridimensionais, necessitando “apenas” o desenvolvimento de interface com o usuário para coleta dos dados necessários.

Um outro objetivo deste trabalho é implementar procedimentos existentes para controle geométrico de forma dos elementos bidimensionais gerados automaticamente, permitindo a visualização e crítica da distorção da forma dos elementos.

Também é objetivo deste trabalho o desenvolvimento de um algoritmo de triangulação de domínios arbitrários bidimensionais, que preve a existência de trincas, e que incorpora procedimentos que garantem a qualidade da forma dos elementos gerados.

Finalmente, o trabalho também objetiva criar procedimentos para automatizar a coleta interativa dos dados necessários para os algoritmos de geração de malhas volumétricas.

² API – Application Program Interface

1.3 Organização do Trabalho

O Capítulo 2 apresenta os cálculos das medidas de distorção para elementos finitos quadriláteros de 4 e 8 nós, e triangulares de 3 e 6 nós. Essas medidas são implementadas em um modelador geométrico gráfico interativo de malhas de elementos finitos (MTool, 1992), apresentando-se através de uma escala de cores que define intervalos numéricos para as medidas. Para esta representação gráfica, um estudo de sensibilidade é feito com respeito às medidas de distorção estudadas.

O Capítulo 3 propõe um algoritmo de triangulação para geração de malhas de elementos finitos, que considera trincas no modelo. Esse algoritmo incorpora uma nova técnica de melhoria local da malha, adaptada de um algoritmo de geração de malhas volumétricas de tetraedros para domínios arbitrários (Cavalcante Neto, 1998). Além disso, é feita uma comparação com respeito à qualidade das malhas geradas pelo algoritmo proposto e por outros algoritmos desenvolvidos na linha de pesquisa.

O Capítulo 4 descreve dois procedimentos para geração de malhas estruturadas de elementos finitos tridimensionais. O primeiro procedimento é o mapeamento transfinito tridimensional, onde a malha é construída a partir de suas seções transversais utilizando-se de curvas do tipo spline cúbica para interpolação das seções transversais, com o reconhecimento automático das malhas das seções transversais. O segundo procedimento é a técnica do *sweep*, onde a malha é construída a partir do “arrasto” de uma seção transversal ao longo de uma curva no espaço.

O Capítulo 5 apresenta a integração dos geradores de malhas sólidas estruturadas e não-estruturadas dentro do ambiente gráfico interativo tridimensional do programa MG (Gomes Coelho, 1998). São descritas as principais modificações que esse programa sofreu para incorporar os algoritmos de malhas volumétricas. São apresentados alguns exemplos de malhas sólidas geradas com diferentes tipos elementos finitos tridimensionais e utilizando-se diversos procedimentos de geração de malhas.

No Capítulo 6 são feitas algumas considerações finais para a conclusão do trabalho desenvolvido. São apresentadas ainda sugestões para trabalhos futuros.

O Apêndice A mostra um estudo de sensibilidade da métrica utilizada no Capítulo 2.

Finalmente, no Apêndice B é descrita a API da biblioteca computacional contendo os algoritmos de geração de malha.

2 Medidas de Distorção para Elementos Finitos Planos

Os resultados obtidos de uma análise utilizando o método dos elementos finitos para resolução de um modelo estrutural dependem principalmente da forma geométrica do elemento, da discretização e do grau utilizado no elemento. No trabalho de Burrows (1986) são feitas análises de sensibilidade de um modelo de viga submetido a momento fletor constante e demonstra-se que os valores de tensões e deslocamentos são influenciados de modo significativo pela forma geométrica dos elementos utilizados. No trabalho de Lee e Bathe (1993), demonstra-se que de acordo com a distorção do elemento existem perdas de termos de mais alto grau da interpolação utilizada na formulação isoparamétrica, não conseguindo-se representar adequadamente os campos de tensões e deslocamentos. No trabalho de Dumont e Fernandez (1997) são confirmadas as conclusões de Lee e Bathe, mas com um estudo mais completo. Por exemplo, o trabalho inclui estudos sobre distorções de espaçamentos de nós e de bordos curvos; conclui que o campo do Jacobiano nulo, em relação ao elemento, ocasiona enrijecimento nos campos de deslocamento e propõe outra formulação mais robusta na qual os efeitos de distorção tenham menor influência nos resultados.

Esses trabalhos indicam que para diminuir ou controlar resultados indevidos na análise de elementos finitos é necessário estabelecer maneiras de se verificar a forma geométrica do elemento utilizado em um dado modelo através de parâmetros que representem sua forma geométrica. Atualmente a maioria dos modeladores de elementos finitos bidimensionais utiliza geração automática de malhas, logo é de grande utilidade se ter uma ferramenta que permita visualizar rapidamente a qualidade da malha gerada no que diz respeito à forma dos elementos. Desta forma, os parâmetros que definem a geração da malha podem ser modificados para obter uma malha com melhor qualidade.

Neste capítulo são obtidos parâmetros de forma para elementos finitos planos do tipo quadrilateral de 4 e 8 nós, e triangulares de 3 e 6 nós. Também é feito um estudo da sensibilidade das medidas de formas dos elementos para definição de faixas de valores a serem representados em um sistema automático de modelagem de elementos finitos. Além disso, é feita uma apresentação da implementação em um modelador automático de malhas de elementos finitos das medidas de distorção dos elementos.

2.1 Obtenção das Medidas de Distorção

Dentre as várias medidas de distorção de elementos finitos encontrados na literatura tem-se: razão, obliquidade, estreitamento e determinante do Jacobiano, dados por Burrows (1986); convexidade, perturbação do nó do bordo curvo e divergências do bordo curvo, dados por Robison (1987, 1988); distorção de proporção dos lados, nós desigualmente espaçados e distorção angular e de bordos curvos, dados por Lee e Bathe (1993). As medidas de distorção adotadas neste trabalho são aquelas em que o autor considera de mais fácil compreensão geométrica: razão, obliquidade, estreitamentos e divergências do nó em bordo curvo. Nesta seção, essas medidas serão definidas para os elementos Serendipity quadrilaterais de 4 e 8 nós e triangulares de 3 e 6 nós.

Para um elemento quadrilateral bidimensional no sistema de coordenadas cartesiano (x, y) , como mostra a Figura 2.1, pode-se obter as seguintes medidas de distorção:

- razão;
- obliquidade;
- estreitamento.

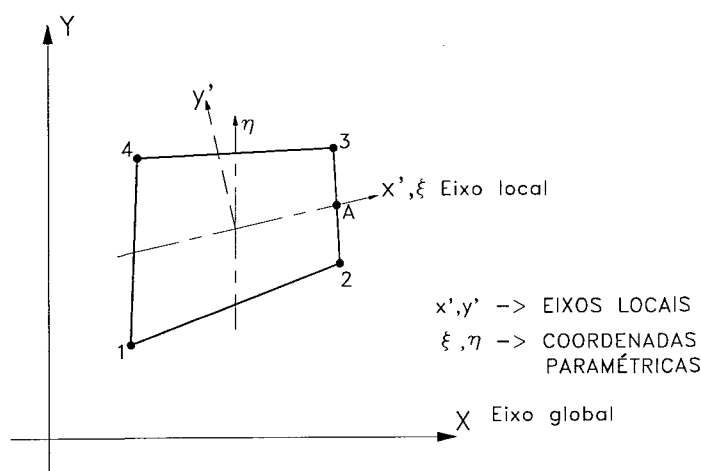


Figura 2.1 - Elemento finito quadrilateral plano.

Essas medidas de distorção de forma podem ser visualizadas na Figura 2.2. A razão é a relação entre os lados de um retângulo (Fig. 2.2a), a obliquidade é a distorção lateral do elemento (Fig. 2.2b) e o estreitamento é o encurtamento lateral do elemento em uma dada direção (Fig. 2.2c e 2.2d). Para os outros elementos (quadrilateral de 8 nós e triangular de 3 e 6 nós) essas medidas de distorção apresentam o mesmo significado geométrico.

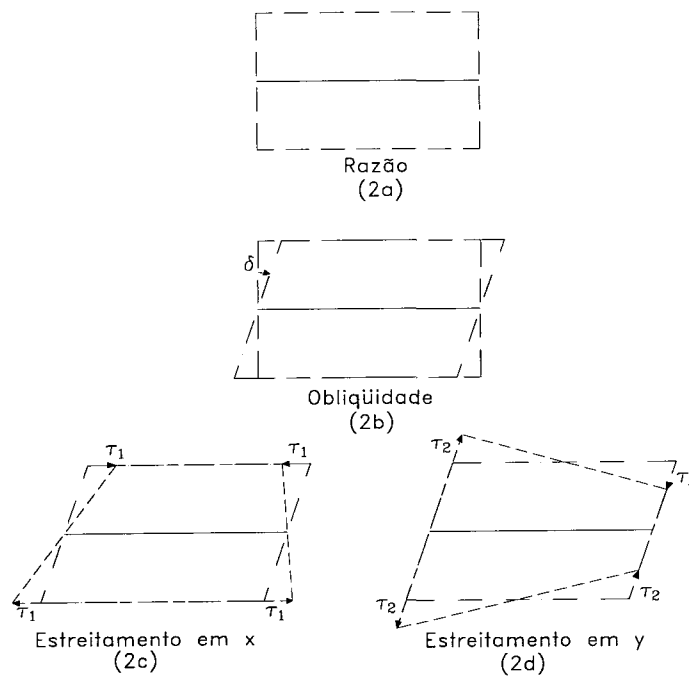


Figura 2.2 - Significado geométrico das medidas de distorção.

No caso de elementos com bordos curvos (quadrilateral de 8 nós e triangular de 6 nós) tem-se as medidas propostas por Robinson (1988): para cada lado do elemento são feitas medidas das divergências normal e tangencial do ponto do bordo curvo (Fig. 2.3) em relação ao ponto ótimo (ponto médio do lado).

No Capítulo 3 é adotada uma medida de distorção para um algoritmo triangulação, adaptada de uma medida para elementos tridimensionais, que não é usada neste Capítulo. Isto é devido aos diferentes usos das medidas de distorção. Neste Capítulo a medida de distorção tem caráter didático e de visualização, enquanto que, no Capítulo seguinte a medida de distorção serve internamente ao algoritmo.

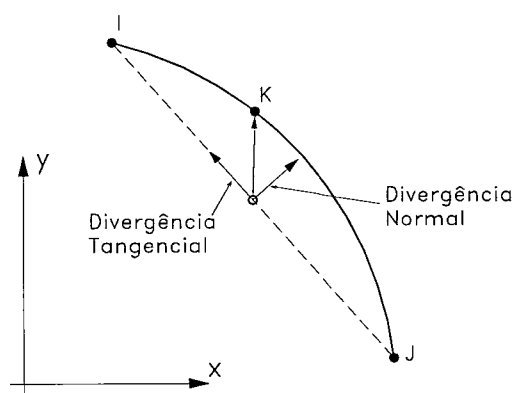


Figura 2.3 - Divergências normal e tangencial em bordo curvo.

2.1.1 Medidas de Distorção do Elemento Quadrilateral de 4 Nós

Seja o quadrilateral plano de 4 nós, no sistema global de coordenadas apresentado na Figura 2.1. A geometria do quadrilátero pode ser escrita através de funções de interpolação (formulação isoparamétrica):

$$x = \sum_{i=1}^4 P_i(\xi, \eta) x_i, \quad y = \sum_{i=1}^4 P_i(\xi, \eta) y_i, \quad (2.1)$$

onde

$$P_1 = 1/4 (1 - \xi)(1 - \eta), \quad P_2 = 1/4 (1 + \xi)(1 - \eta), \quad (2.2)$$

$$P_3 = 1/4 (1 + \xi)(1 + \eta), \quad P_4 = 1/4 (1 - \xi)(1 + \eta),$$

e ξ e η são coordenadas adimensionais com limites ± 1 . Uma forma alternativa de representação é usar polinômios básicos:

$$x = e_1 + e_2\xi + e_3\eta + e_4\xi\eta, \quad y = f_1 + f_2\xi + f_3\eta + f_4\xi\eta, \quad (2.3)$$

onde os coeficientes e e f estão relacionados com as coordenadas x_1 a x_4 e y_1 a y_4 por:

$$e_1 = 1/4 (x_1 + x_2 + x_3 + x_4), \quad e_2 = 1/4 (-x_1 + x_2 + x_3 - x_4), \quad (2.4)$$

$$e_3 = 1/4 (-x_1 - x_2 + x_3 + x_4), \quad e_4 = 1/4 (x_1 - x_2 + x_3 - x_4),$$

$$f_1 = 1/4 (y_1 + y_2 + y_3 + y_4), \quad f_2 = 1/4 (-y_1 + y_2 + y_3 - y_4),$$

$$f_3 = 1/4 (-y_1 - y_2 + y_3 + y_4), \quad f_4 = 1/4 (y_1 - y_2 + y_3 - y_4).$$

Diferentemente das funções de interpolação das equações (2.2), os coeficientes e e f tem significado geométrico como mostra a Figura 2.4.

Transportando o elemento para o sistema local $x'y'$, como mostrado na Figura 2.1, tem-se os coeficientes $e_1 = f_1 = f_2 = 0$ e os coeficientes restantes podem ser utilizados mais facilmente para o cálculo das medidas de distorção. A Figura 2.5 mostra o significado geométrico dos coeficientes no sistema local de coordenadas.

As medidas de distorção de forma são obtidas em função dos coeficientes e e f conforme mostrado abaixo:

$$\text{razão} = \frac{e_2}{f_3} \text{ ou } \frac{f_3}{e_2} \text{ (considerar o maior valor),} \quad (2.5)$$

substituindo valores de e_2 e f_3 ,

$$\text{razão} = \frac{-x_1 + x_2 + x_3 - x_4}{-y_1 - y_2 + y_3 + y_4} \text{ ou } \frac{-y_1 - y_2 + y_3 + y_4}{-x_1 + x_2 + x_3 - x_4} \text{ (maior valor)} \quad (2.6)$$

$$\text{obliquidade} = \frac{e_3}{f_3} = \frac{-x_1 - x_2 + x_3 + x_4}{-y_1 - y_2 + y_3 + y_4} \quad (2.7)$$

$$\text{estreitamento na direção } x = \frac{f_4}{f_3} = \frac{y_1 - y_2 + y_3 - y_4}{-y_1 - y_2 + y_3 + y_4} \quad (2.8)$$

$$\text{estreitamento na direção } y = \frac{e_4}{e_2} = \frac{x_1 - x_2 + x_3 - x_4}{-x_1 + x_2 + x_3 - x_4} \quad (2.9)$$

A partir de um elemento quadrilateral no sistema global de coordenadas é necessário estabelecer as coordenadas dos nó em um sistema local de coordenadas, como mostra a Figura 2.1. A origem do sistema local (ponto "0") é determinada por

$$V0 = \frac{1}{4} (V1 + V2 + V3 + V4), \quad (2.10)$$

onde V_i é o vetor posição do nó i .

O eixo local é determinado pela linha que une a origem ($V0$) e o ponto A (ponto médio do lado $\overline{12} - VA$), e o correspondente vetor unitário, x' , é determinado pela relação

$$x' = V0A / \text{abs}(V0A), \quad (2.11)$$

onde

$$V0A = VA - V0 \quad (2.12)$$

O eixo y local é agora definido por seu vetor unitário, y' , onde

$$y' = z \times x', \quad (2.13)$$

onde z é o vetor unitário perpendicular ao plano.

As coordenadas cartesianas x' e y' do nó 1 no sistema local são denotados por x'_1 e y'_1 , respectivamente. Estas são avaliadas pelos produtos escalares dos vetores

$$x'_1 = V01 \cdot x', \quad y'_1 = V01 \cdot y' \quad (2.14)$$

onde

$$V01 = V1 - V0. \quad (2.15)$$

Analogamente para os nós 2, 3, e 4:

$$x'_2 = V02 \cdot x' \qquad y'_2 = V02 \cdot y' \qquad (2.16)$$

$$x'_3 = V03 \cdot x' \qquad y'_3 = V03 \cdot y'$$

$$x'_4 = V04 \cdot x' \qquad y'_4 = V04 \cdot y'$$

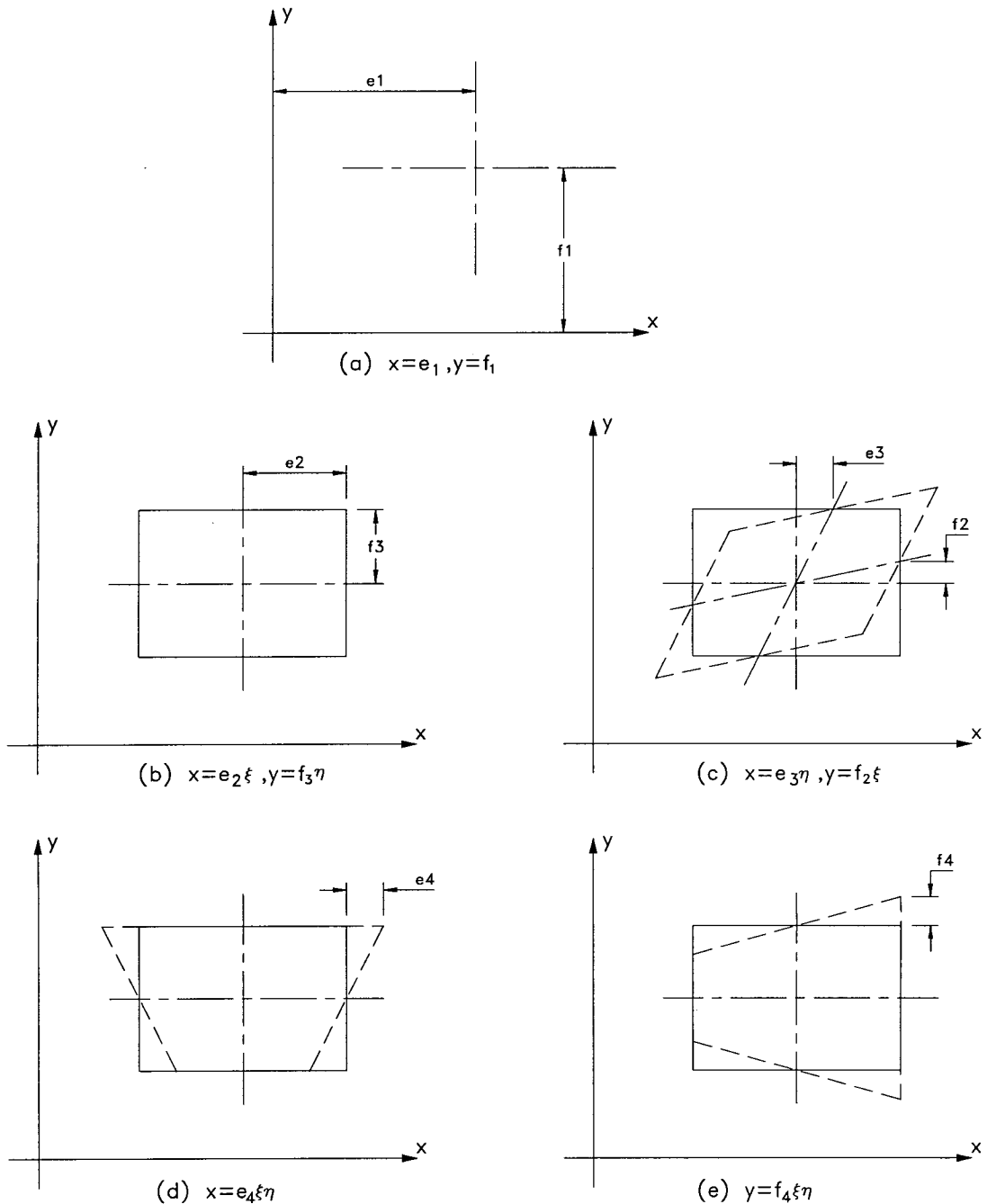


Figura 2.4 - Significado geométrico dos coeficientes e e f .

Pode-se observar que para o elemento quadrilateral plano de 4 nós é possível obter quatro posições para os eixos locais. A Figura 2.6 mostra duas delas, sendo as outras duas

possíveis posições semelhante a essas duas. Para cada posição obtida dos eixos locais, valores diferentes para as medidas de distorção também são obtidas. Neste caso, assumem-se as maiores distorções de forma para a razão e obliquidade. Os valores dos estreitamentos quase sempre são os mesmos quando os eixos locais trocam de posição, apenas ocorre uma troca de valores devida a esta mudança de posição.

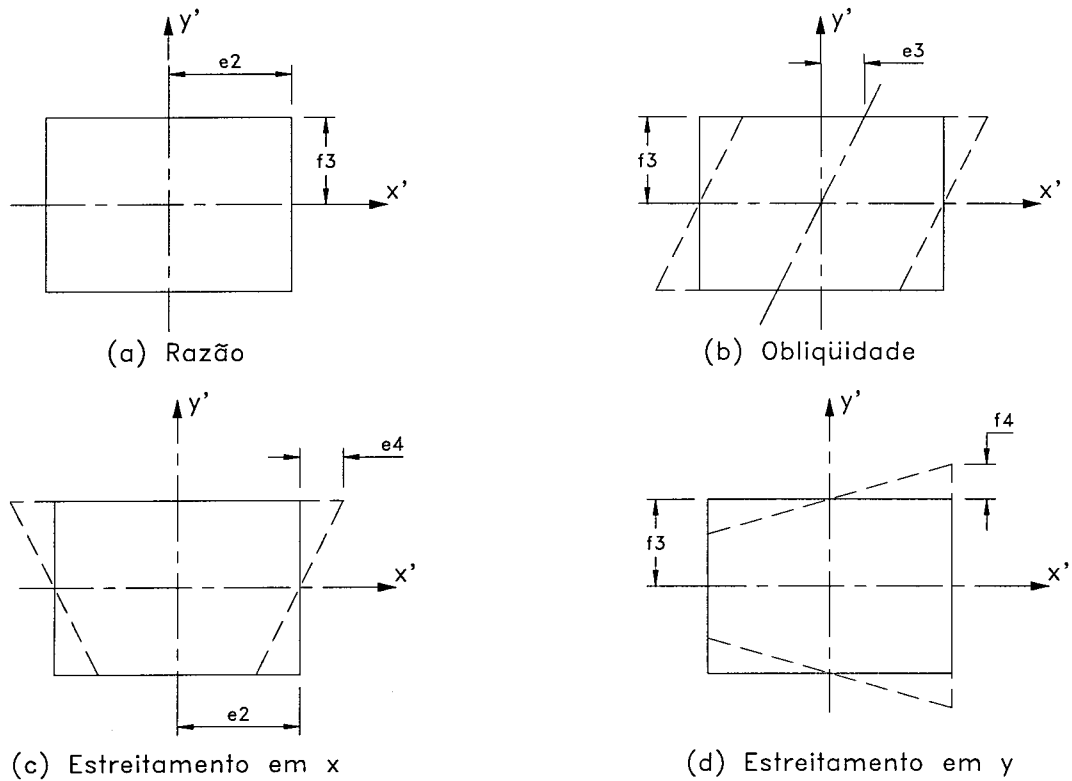


Figura 2.5 - Coeficientes e e f no sistema local de coordenadas.

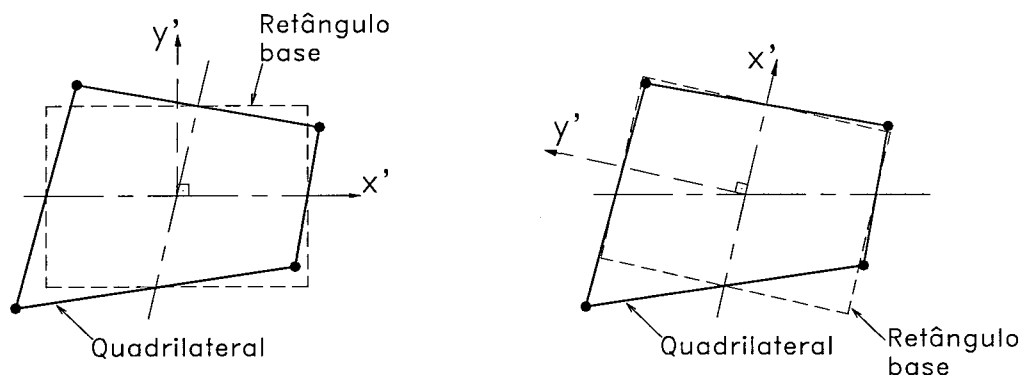


Figura 2.6 - Possibilidades do eixo local.

2.1.2 Medidas de Distorção do Elemento Quadrilateral de 8 Nós

A Figura 2.7 mostra um elemento finito quadrilateral de 8 nós com bordos curvos. Os nós estão referenciados como nós de vértices (1 – 4) e nós de bordo curvo (5 – 8). O sistema de eixos locais para o elemento de 8 nós é obtido da mesma forma como foram obtidos os eixos locais para o quadrilateral de 4 nós. Também existem quatro opções para o sistema de eixos locais.

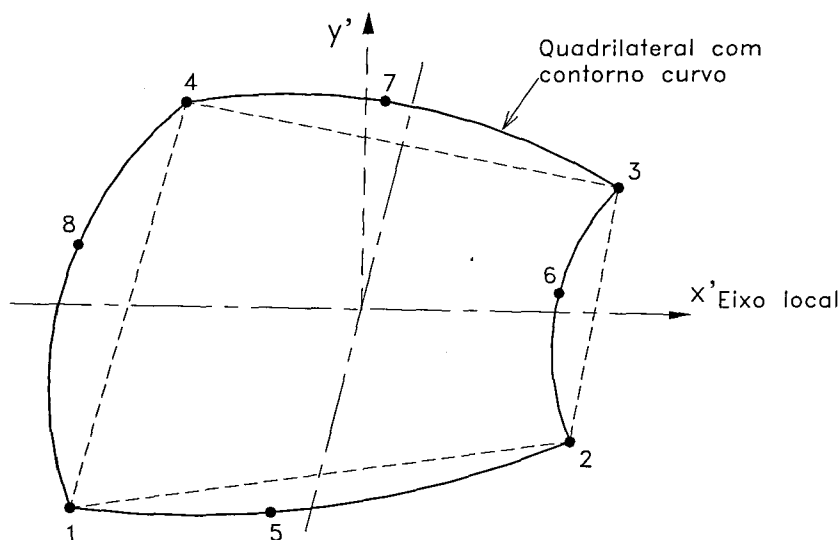


Figura 2.7 - Quadrilateral com bordo curvo.

A geometria de um elemento quadrilateral de 8 nós pode ser expressa na forma polinomial simples:

$$x = e_1 + e_2\xi + e_3\eta + e_4\xi\eta + e_5\xi^2 + e_6\eta^2 + e_7\xi^2\eta + e_8\xi\eta^2 \quad (2.17)$$

$$y = f_1 + f_2\xi + f_3\eta + f_4\xi\eta + f_5\xi^2 + f_6\eta^2 + f_7\xi^2\eta + f_8\xi\eta^2,$$

onde x e y são coordenadas dos pontos no sistema global e os coeficientes e e f são combinações das coordenadas nodais. Os coeficientes e e f são dados por:

$$e_1 = -1/4 (x_1 + x_2 + x_3 + x_4) + 1/2 (x_5 + x_6 + x_7 + x_8) \quad (2.18)$$

$$e_2 = 1/2 (x_6 - x_8)$$

$$e_3 = 1/2 (-x_5 + x_7)$$

$$e_4 = 1/4 (x_1 - x_2 + x_3 - x_4)$$

$$e_5 = \frac{1}{4} (x_1 + x_2 + x_3 + x_4) - \frac{1}{2} (x_5 + x_7)$$

$$e_6 = \frac{1}{4} (x_1 + x_2 + x_3 + x_4) - \frac{1}{2} (x_6 + x_8)$$

$$e_7 = \frac{1}{4} (-x_1 - x_2 + x_3 + x_4) + \frac{1}{2} (x_5 - x_7)$$

$$e_8 = \frac{1}{4} (-x_1 + x_2 + x_3 - x_4) + \frac{1}{2} (-x_6 + x_8).$$

Os coeficientes f são definidos de forma análoga para a coordenada y .

Os oito coeficientes $e_1 - e_4$ e $f_1 - f_4$ tem o mesmo significado geométrico da Figura 2.4. O significado geométrico dos oito coeficientes restantes $e_5 - e_8$ e $f_5 - f_8$ está ilustrado na Figura 2.8.

As seguintes medidas de distorção são obtidas:

$$\text{razão} = \frac{e_2}{f_3} \text{ ou } \frac{f_3}{e_2} \text{ (considerar o maior valor, ver Figura 2.9), ou então,} \quad (2.19)$$

$$\text{razão} = \frac{x_6 - x_8}{-y_5 + y_7} \text{ ou } \frac{-y_5 + y_7}{x_6 - x_8} \text{ (considerar o maior valor)} \quad (2.20)$$

$$\text{obliquidade} = \frac{e_3}{f_3} + \frac{f_2}{e_2} = \frac{-x_5 + x_7}{-y_5 + y_7} + \frac{y_6 - y_8}{x_6 - x_8} \quad (2.21)$$

$$\text{estreitamento na direção } x = \frac{f_4}{f_3} = \frac{y_1 - y_2 + y_3 - y_4}{-y_5 + y_7} \cdot \frac{1}{2} \quad (2.22)$$

$$\text{estreitamento na direção } y = \frac{e_4}{e_2} = \frac{x_1 - x_2 + x_3 - x_4}{x_6 - x_8} \cdot \frac{1}{2} \quad (2.23)$$

Para os bordos curvos, tem-se as medidas de divergência normal e divergência tangencial do ponto do bordo curvo propostas por Robison (1988). Essas medidas são dadas por:

$$\text{Divergência Tangencial (TD)} = \frac{\alpha}{\frac{1}{2} \text{abs}(v_{ij})} \quad (2.24)$$

$$\text{Divergência Normal (ND)} = \frac{\gamma}{\frac{1}{2} \text{abs}(v_{ij})}$$

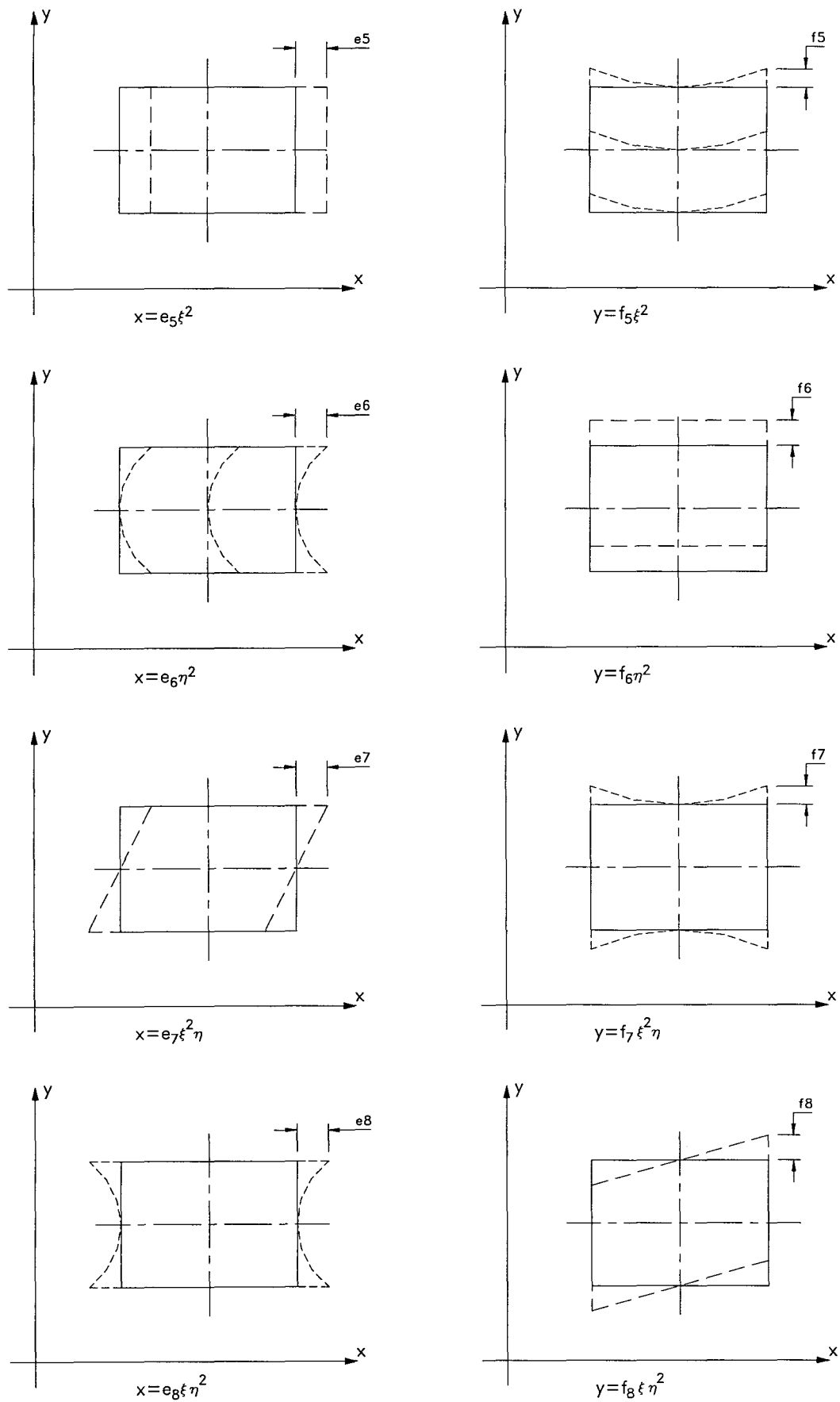


Figura 2.8 - Interpretação geométrica dos coeficientes $e_5 - e_8$ e $f_5 - f_8$.

As medidas α e γ estão ilustradas na Figura 2.10. v_{ij} é o vetor entre os nós i e j de uma aresta do elemento. O vetor v_{ij} é dado por

$$v_{ij} = v_j - v_i \quad (2.25)$$

O ponto médio entre os nós da aresta do elemento tem o vetor

$$v_0 = 1/2 (v_i + v_j), \quad (2.26)$$

e o vetor local, do ponto médio v_0 até o nó de contorno curvo v_k ,

$$v_{0k} = v_k - v_0. \quad (2.27)$$

Os vetores unitários normal e tangencial são α e γ , respectivamente. São dados por:

$$\alpha = \frac{v_{ij}}{\text{abs}(v_{ij})} \quad (2.28)$$

$$\gamma = \alpha \times \beta,$$

onde β é o vetor unitário perpendicular ao plano. Os valores de α e γ são obtidos pelos produtos escalares:

$$\alpha = v_{0k} \cdot \alpha \quad (2.29)$$

$$\gamma = v_{0k} \cdot \gamma.$$

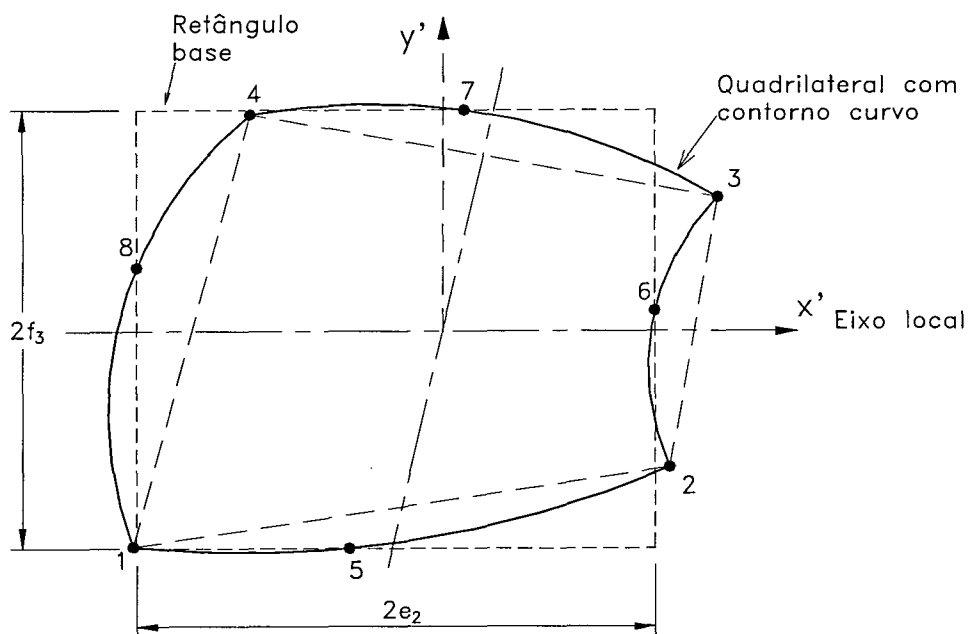


Figura 2.9 - Retângulo da razão no quadrilateral.

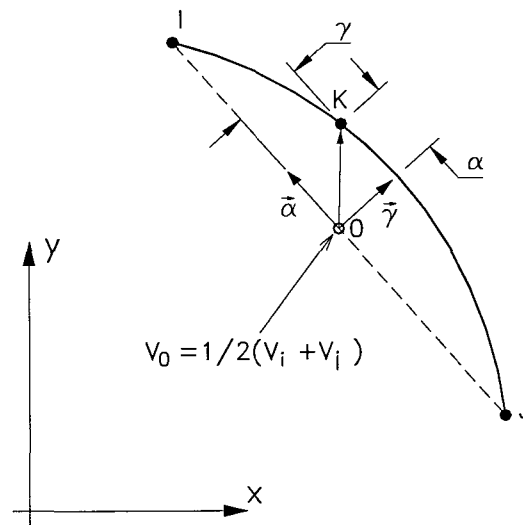


Figura 2.10 - Vetores no bordo curvo.

2.1.3 Medidas de Distorção do Elemento Triangular de 3 Nós

A formulação das medidas de distorção para elementos triangulares é feita de modo similar aos elementos quadrilaterais, sendo que não existem os estreitamentos. A razão e a obliquidade estão ilustradas na Figura 2.11.

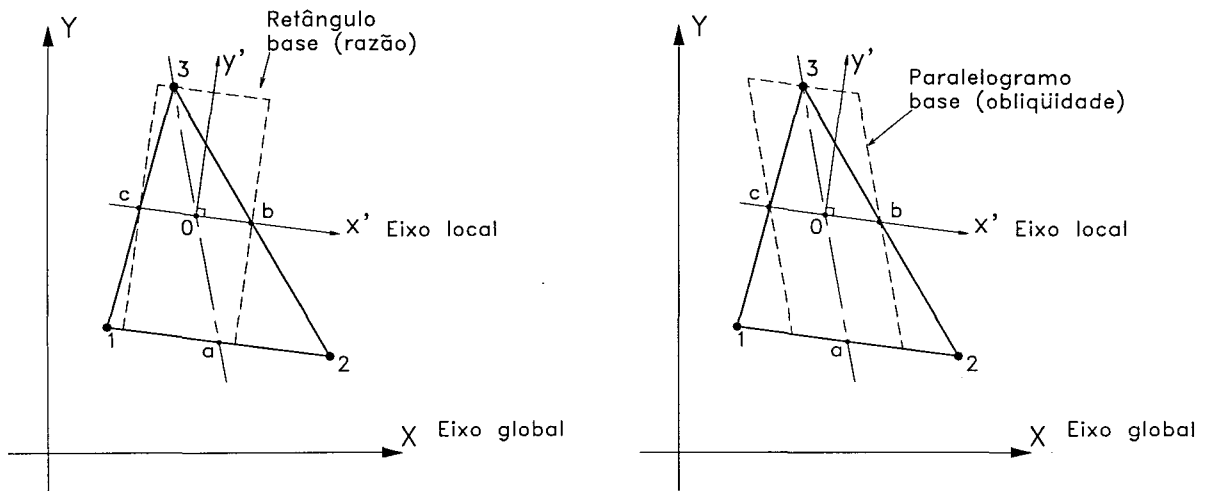


Figura 2.11 - Medidas de distorção para o elemento triangular plano.

Seja o elemento triangular na Figura 2.11, no sistema global de coordenadas. Os pontos a , b e c são os pontos médios das arestas do elemento. A origem sistema de eixos locais de coordenadas é dada pela média dos pontos b e c . A direção do eixo x' é paralela a aresta $\overline{12}$ e o sentido é dado pela origem 0 e ponto b . Os mesmos procedimentos feitos para o elemento quadrilateral são feitos para o cálculo das coordenadas dos pontos do elemento

triangular no sistema local. Deve-se considerar que o sistema de eixos locais pode se encontrar de três modos diferentes; em cada opção o eixo local x' é paralelo a cada aresta do elemento triangular. Tem-se desse modo três medidas para a razão e três medidas para a obliquidade, sendo utilizadas as maiores medidas.

Um ponto a ser notado no cálculo das medidas de distorção em elementos triangulares é que o elemento triangular sem distorções (ideal) é o que tem a forma equilátera, necessitando-se assim fazer uma correção no cálculo das medidas de distorção. A partir da Figura 2.12, tem-se que

$$\text{razão} = \frac{L\sqrt{3}/2}{L/2} = \sqrt{3} = AR_{\text{ideal}}. \quad (2.30)$$

Logo a razão ideal para os elementos triangulares é $\sqrt{3} \approx 1,73$.

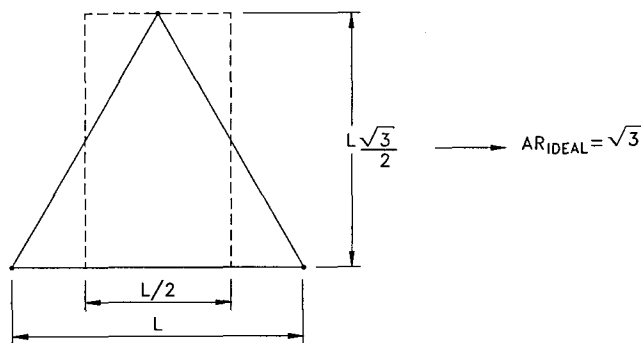


Figura 2.12 - Razão ideal para o elemento triangular.

De acordo com a interpretação geométrica dos coeficientes vista para o elemento quadrilateral de 4 nós, pode-se depreender as medidas de distorção para elementos triangulares de 3 nós:

$$\text{razão} = \frac{1}{2} \frac{x_2 - x_1}{y_3 - y_a} \cdot AR_{\text{ideal}} \text{ ou } 2 \frac{y_3 - y_a}{x_2 - x_1} \cdot (1 / AR_{\text{ideal}}) \quad (\text{maior valor}), \quad (2.31)$$

$$\text{obliquidade} = \frac{x_3 - x_a}{y_3 - y_a}. \quad (2.32)$$

2.1.4 Medidas de Distorção do Elemento Triangular de 6 Nós

Para o elemento triangular de 6 nós ilustrado na Figura 2.13, existem também três

modos diferentes para o sistema de eixos locais. Os mesmos procedimentos de cálculo vistos para o elemento triangular de 3 nós são feitos para o elemento triangular de 6 nós (pontos a , b , c e a origem). Os parâmetros de forma para os contornos curvos são os mesmos adotados para o quadrilateral de 8 nós, isto é, a divergência normal e a divergência tangencial

As medidas de distorção são inferidas a partir da interpretação geométrica com o elemento no sistema local de coordenadas. As medidas de distorção são dadas por:

$$\text{razão} = \frac{x_5 - x_6}{y_3 - y_4} \text{ AR}_{\text{ideal}} \text{ ou } 2 \frac{y_3 - y_4}{x_5 - x_6} (1 / \text{AR}_{\text{ideal}}) \quad (\text{maior valor}) \quad (2.33)$$

$$\text{obliquidade} = \frac{x_3 - x_4}{y_3 - y_4} + \frac{y_5 - y_6}{x_5 - x_6}. \quad (2.34)$$

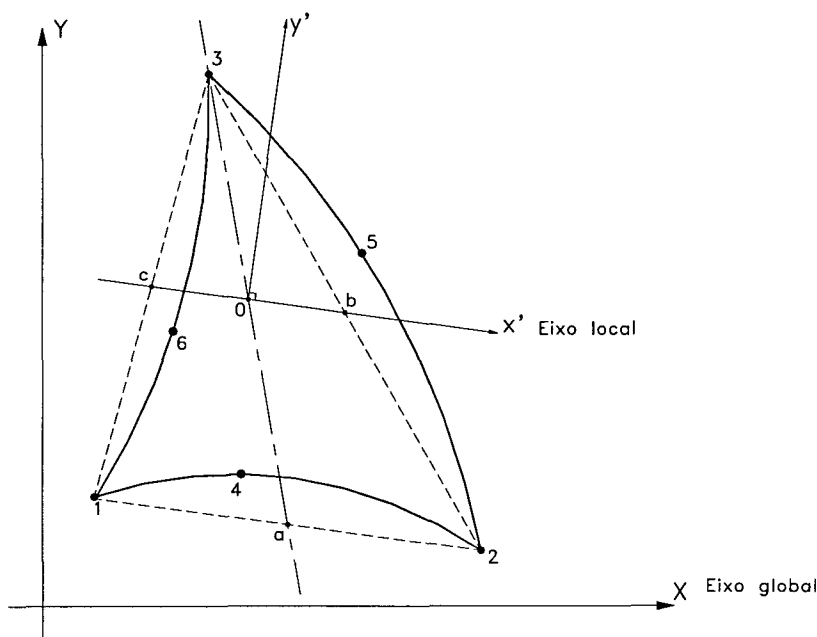


Figura 2.13 - Elemento triangular plano de 6 nós.

2.2 Representação das Medidas de Distorção

Com o objetivo de implementar as medidas de distorção em um modelador de malhas de elementos finitos de modo a permitir uma rápida e fácil visualização da distorção dos elementos, concebeu-se uma representação através de uma escala de cores. Em uma escala de cores cada cor representa um intervalo numérico para a medida de distorção. O elemento é “pintado” pela cor no qual sua medida de distorção se encontra. A Figura 2.14 mostra um exemplo da escala de cores representado a razão do elemento.

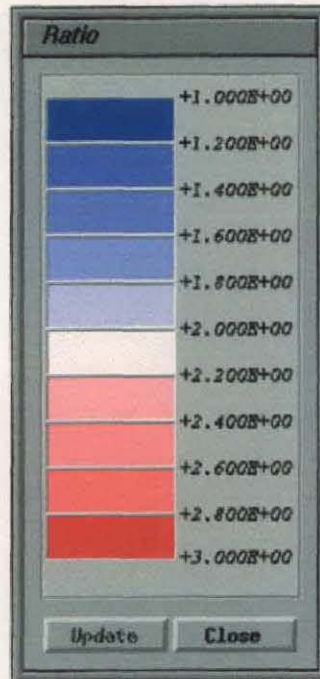


Figura 2.14 - Exemplo de uma escala de cores.

Vale ressaltar que o objetivo da implementação é apenas uma representação visual da medida de distorção para um dado elemento, levando ao usuário do modelador geométrico a informação de possíveis elementos que julgue com qualidade indesejável.

Nesse tipo de representação é necessário informar os valores máximos e mínimos, e a subdivisão das cores. Com o intuito de avaliar estes valores, torna-se necessário fazer testes de sensibilidade das medidas de distorção. Esses testes são:

- (a) Para o quadrilátero de 4 nós com coordenadas $(0,0)$, $(1,0)$, $(1,1)$ e $(0,1)$, varia-se a posição do nó 3 com coordenada $(1,1)$ de $x = 0,2$ à $x = 2,5$ e y constante, como mostra a Figura 2.15a. Os resultados estão na Figura 2.16.
- (b) Para o elemento triangular de 3 nós com coordenadas $(-\sqrt{3}/3,0)$, $(\sqrt{3}/3,0)$ e $(0,1)$, um triângulo equilátero de altura 1, varia-se a posição do nó 3 $(0,1)$ de $y = 0,2$ à $y = 3$, como na Figura 2.15b. Os resultados estão na Figura 2.17.
- (c) Para o elemento triangular de 3 nós com coordenadas $(-1,0)$, $(1,0)$ e $(0, \sqrt{3})$, varia-se a posição do nó 3 $(0, \sqrt{3})$ de $x = 0$ à $x = 3$, como mostra a Figura 2.15c. Os resultados estão na Figura 2.18.

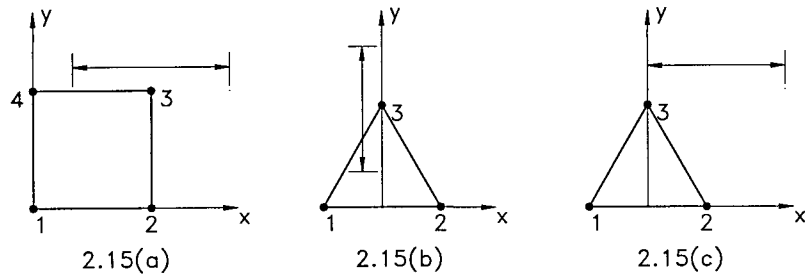


Figura 2.15 - Testes de sensibilidade das medidas de distorção nos elementos finitos planos.

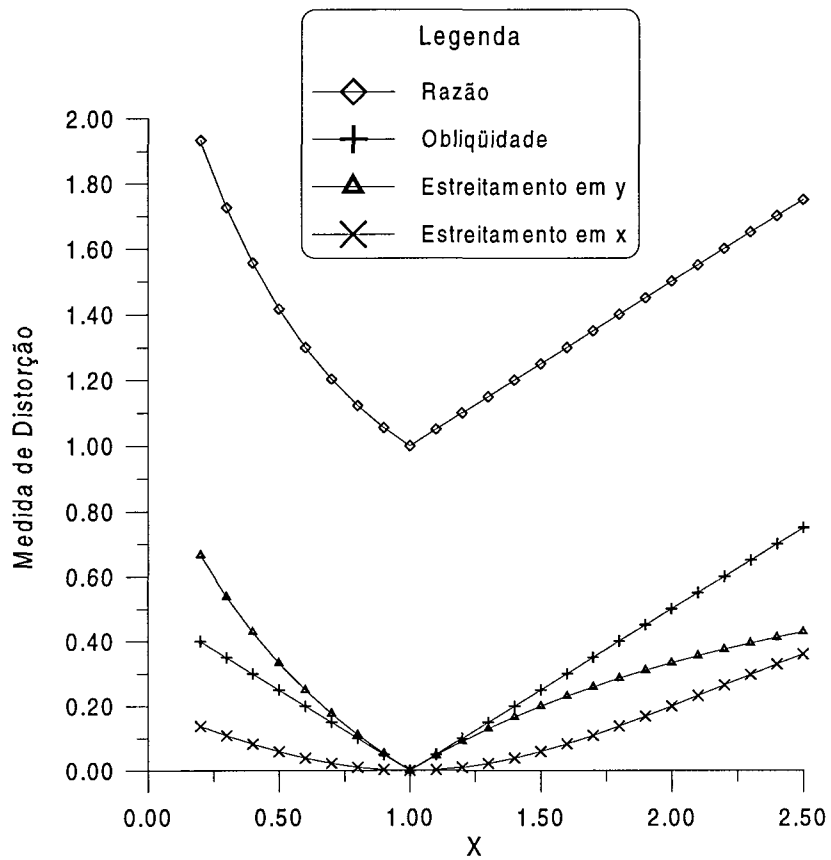


Figura 2.16 - Sensibilidade para o elemento finito quadrilateral - teste (a).

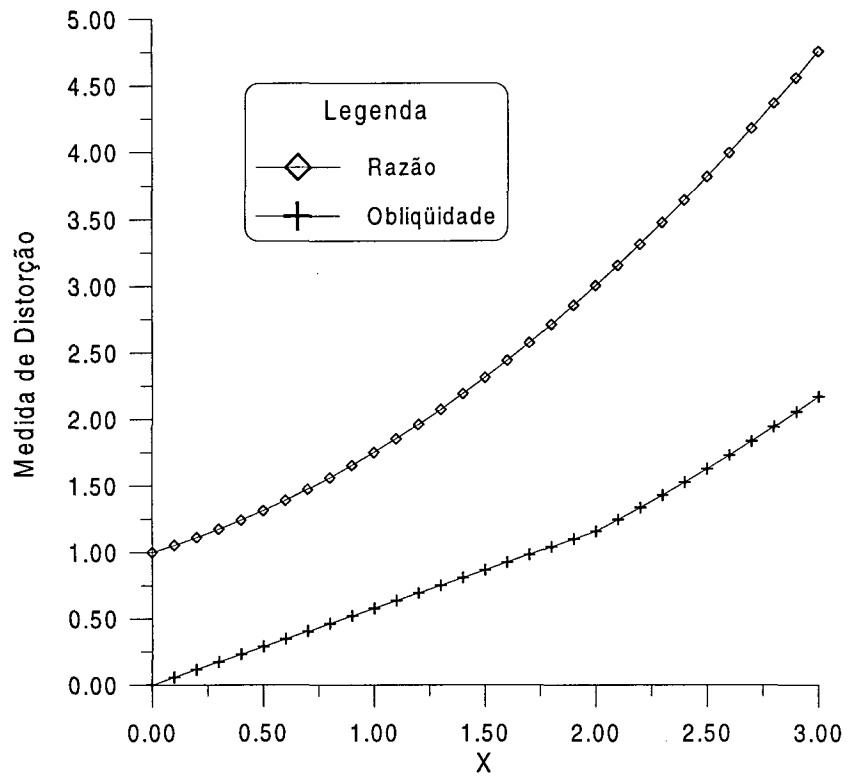


Figura 2.17 - Sensibilidade para o elemento finito triangular - teste (b).

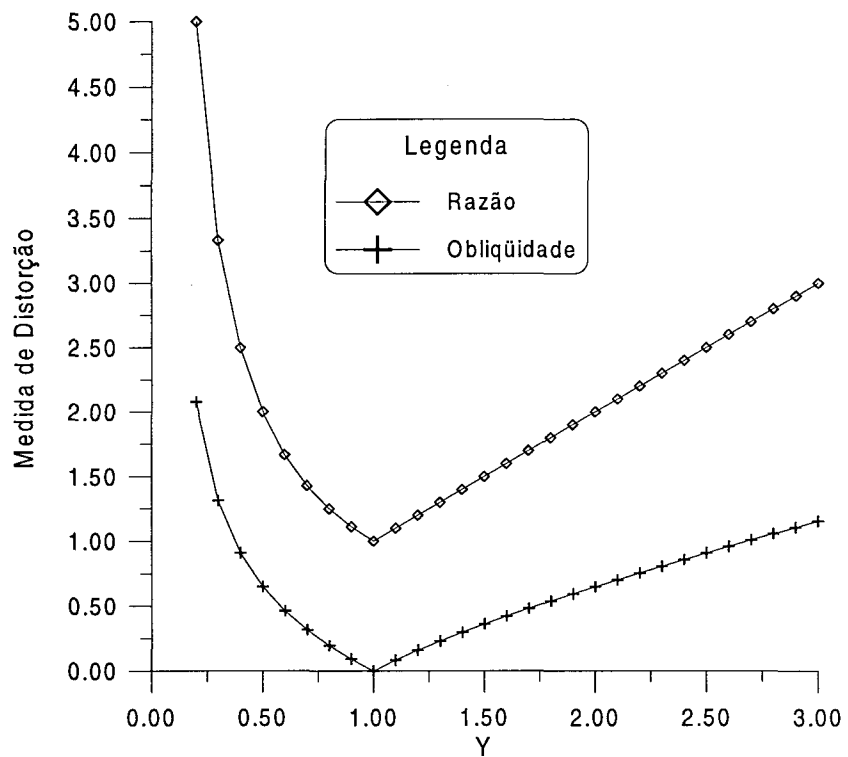


Figura 2.18 - Sensibilidade para o elemento finito triangular - teste (c).

De modo geral, os valores máximos e mínimos das medidas de distorção (Figuras 2.16 a 2.18) são estas mostradas Tabela 2.1.

Tabela 2.1 - Valores máximos e mínimos das medidas de distorção.

Medidas de Distorção	Min	Max
Razão	1	∞
Obliquidade	0	∞
Estreitamentos	0	∞
Divergência Normal	0	∞
Divergência Tangencial	0	1

Percebe-se com base nas Figuras 2.16 a 2.18 que as medidas de distorção razão e obliquidade, comum para os elementos triangulares e quadrilaterais, tem um aumento mais significativo nos elementos triangulares, quando os pontos variam de posição. Com objetivo de aplicar um valor máximo único para a escala de cores de cada medida, adotou-se um valor médio que represente adequadamente a razão e a obliquidade para os elementos. Estes valores, mostrados na Tabela 2.2, foram conseguidos a partir de testes e tentativas de representação no modelador geométrico utilizado. Para os estreitamentos, a variação é pequena em relação às outras medidas, significando assim um valor máximo menor. Os valores máximos das divergências foram obtidas a partir das tentativas de visualização no modelador geométrico.

Tabela 2.2 - Máximos e mínimos adotados na representação das medidas de distorção.

Medidas de Distorção	Min	Max
Razão	1	3
Obliquidade	0	2
Estreitamentos	0	1
Divergência Normal	0	0.5
Divergência Tangencial	0	0.5

2.3 Implementação e Exemplos

A implementação das funções de visualização das medidas de distorção foi feita no modelador geométrico Mtool (1992), que utiliza o sistema gráfico CD (1996) e o sistema de interface IUP/Lua, um toolkit portátil para interface com usuário (Iup/Lua, 1995). Esse ambiente permite a portabilidade automática, sem alteração do código, para outras plataformas onde o sistema funciona, como estações de trabalho baseados no Sistema Operacional Unix e micro-computadores da linha PC.

No ambiente gráfico do Mtool, uma lista do tipo *drop-down* na barra de ferramentas foi inserida, como mostra a Figura 2.19. O modelo é *pintado* instantaneamente quando é feita a seleção de uma das opções na lista.



Figura 2.19 - Lista *drop-down* para escolha de representação da medida de distorção.

As Figuras 2.20 a 2.23 ilustram alguns exemplos de modelos bidimensionais com malhas de elementos finitos geradas pelo modelador geométrico Mtool. Os elementos são *pintados* de acordo com sua cor no qual sua medida de distorção se encontra.

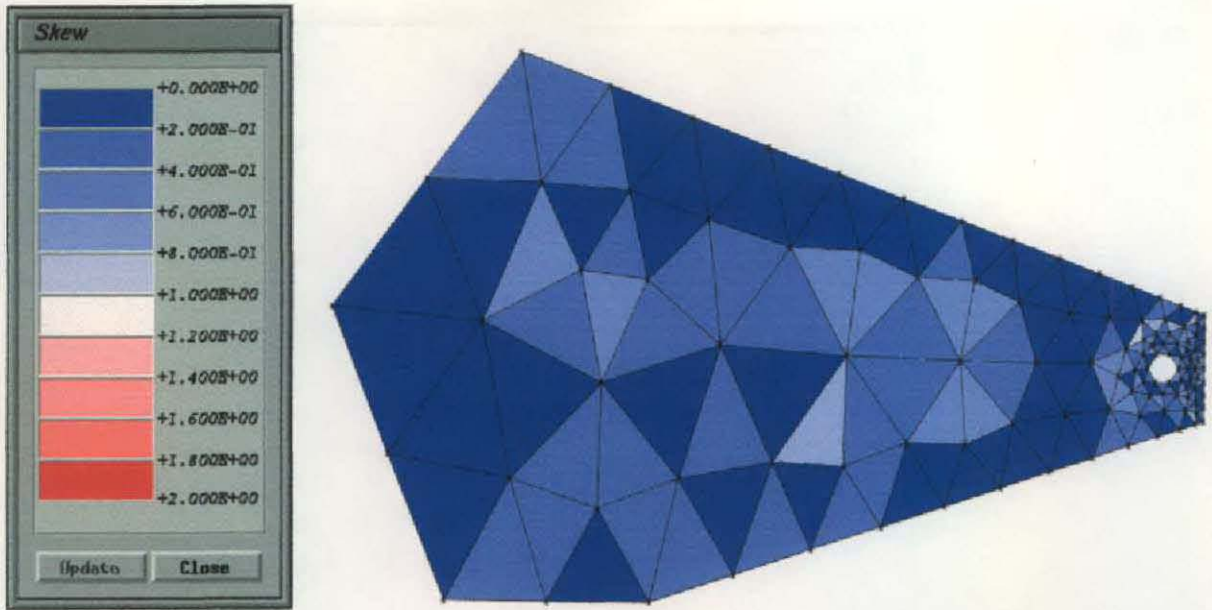


Figura 2.20 - Exemplo da implementação com a medida de distorção obliquidade.

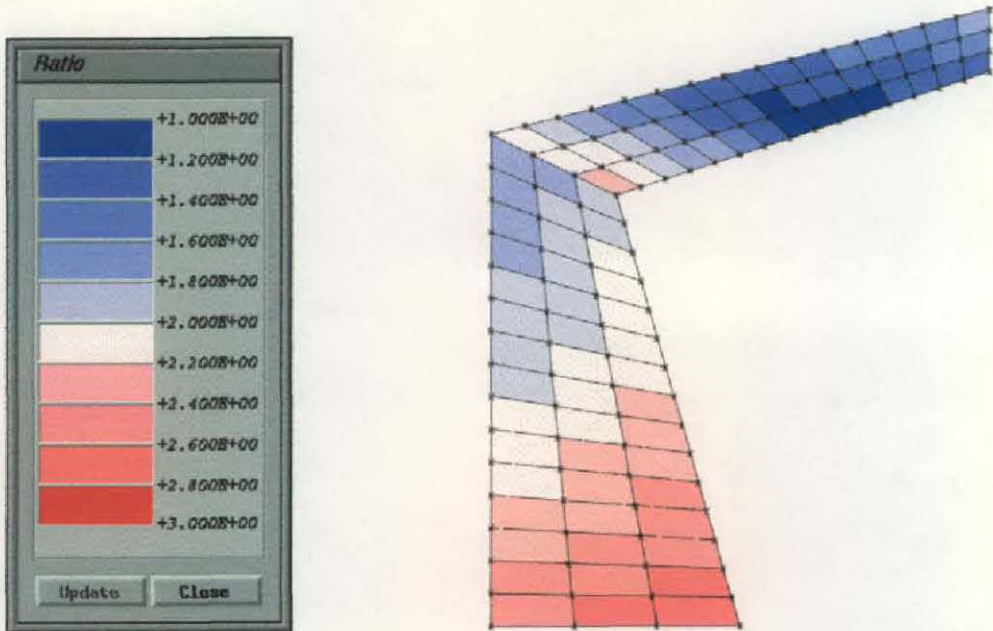


Figura 2.21 - Exemplo da implementação com a medida de distorção razão.

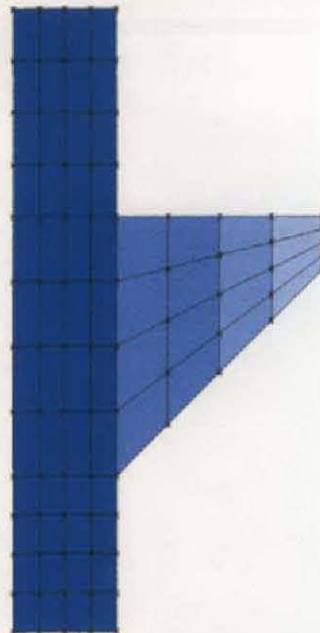
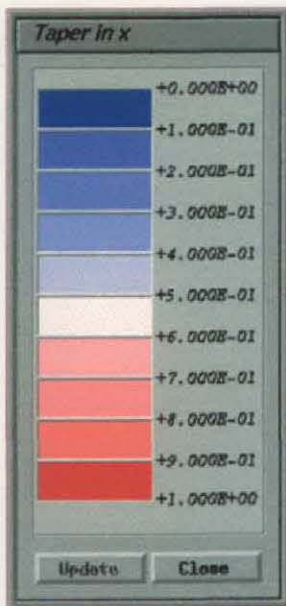


Figura 2.22 - Exemplo da implementação com a medida de distorção estreitamento.

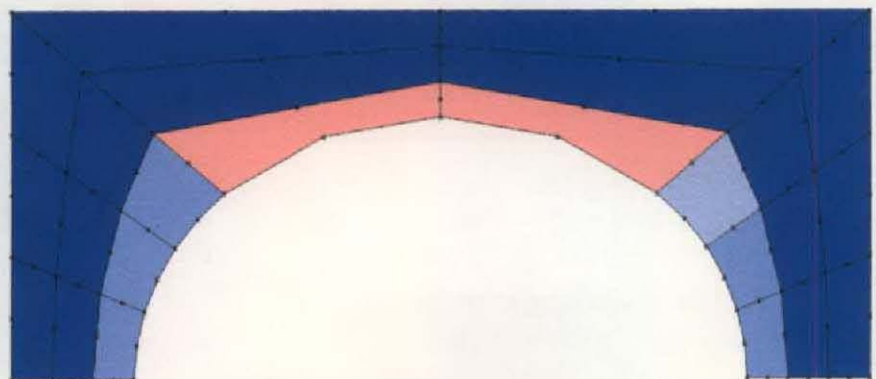
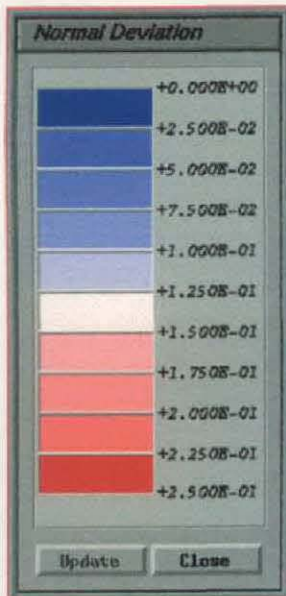


Figura 2.23 - Exemplo da implementação com a medida de distorção divergência normal.

3 Algoritmo de Triangulação Arbitrária com Consideração de Trincas

Este capítulo descreve um algoritmo para geração de malhas triangulares para domínios bidimensionais de forma arbitrária. O algoritmo também funciona para domínios com uma ou múltiplas trincas. Essas trincas podem ser internas ou localizadas no contorno do domínio.

O algoritmo é uma adaptação para o caso bidimensional do trabalho de geração automática de elementos tetraédricos em volumes arbitrários de Cavalcante Neto (1998). O algoritmo foi desenvolvido para atender quatro requisitos específicos. Primeiro, o algoritmo deve produzir elementos de boa forma, evitando a geração de elementos de má qualidade, se possível. Apesar do algoritmo não garantir limites na qualidade dos elementos, cuidados são tomados a cada passo para gerar os melhores elementos possíveis. Além disso, o algoritmo contém um procedimento *a posteriori* que procura melhorar localmente a qualidade dos elementos gerados.

O segundo requisito é que o algoritmo deve gerar uma malha que se conforme com uma dada discretização existente no contorno do domínio. Isto é importante porque permite uma reconstrução local, por exemplo, em uma região perto de uma trinca se propagando. Isto é, um grupo relativamente pequeno de elementos perto de uma trinca pode ser eliminado, criando um vazio na malha. Quando a trinca se propaga, o algoritmo pode ser usado para gerar novos elementos que preencham este vazio, e que se conformam com os elementos que não foram removidos.

O terceiro requisito do algoritmo é que ele deve gerar uma boa transição entre regiões com diferentes tamanhos de elementos. Em uma análise de fraturamento, por exemplo, não é incomum que elementos perto da trinca tenham um tamanho característico duas ordens de magnitude menor do que de outros elementos no domínio. Alguns outros algoritmos encontrados na literatura funcionam melhor quando todos os elementos gerados têm um tamanho característico semelhante. Isto é claramente inaceitável no caso de trincas, e o algoritmo proposto foi desenvolvido para ser capaz de gerar malhas com boa transição entre diferentes tamanhos de elementos. Este aspecto também é importante em análises adaptativas, onde regiões com altos gradientes da solução devem ser refinadas, enquanto em outras regiões os elementos podem ter tamanhos maiores.

O quarto requisito é a necessidade de tratamentos geométricos especiais para o caso de trincas. Esse requisito surge porque trincas são normalmente idealizadas com arestas distintas representando os dois lados da trinca, mas geometricamente coincidentes. Isto significa que nós em lados opostos de arestas da trinca têm coordenadas idênticas. O algoritmo deve ser capaz de discriminar entre os nós e escolher aquele que se localiza na face correta da trinca.

Algoritmos de triangulação arbitrária já foram abordados em outros trabalhos, entre os quais se destacam o trabalho de Vianna (1992) e Cavalcante Neto (1994) dentro da presente linha de pesquisa. Diferente desses trabalhos, o algoritmo aqui apresentado, embora bidimensional, incorpora avanços conseguidos na geração de elementos em volume arbitrário que só foram concebidos devido à complexidade do problema tridimensional. De um modo geral, o algoritmo tem como objetivo gerar elementos de melhor qualidade em relação aos algoritmos anteriores e considerar o caso de trincas no contorno discretizado do modelo.

Muitos outros algoritmos de triangulação foram desenvolvidos seguindo enfoques diversos. Não é o objetivo aqui a revisão desses trabalhos. Pode-se citar apenas como referência os trabalhos de Lo (1985), de Floriani e Puppo (1992), Watson (1981) e Joe e Simpson (1986). Este capítulo apresenta uma descrição detalhada dos passos do algoritmo proposto para geração de malhas triangulares. Além disso, é feita uma comparação de tempo e qualidade entre os algoritmos de triangulação em domínios arbitrários baseados em *quadtree* desenvolvidos anteriormente na linha de pesquisa.

3.1 Descrição do Algoritmo

O algoritmo proposto incorpora aspectos de técnicas para geração de malhas bem conhecidas na literatura e define alguns passos originais adicionais. O algoritmo se baseia em uma técnica de avanço da fronteira (Shaw e Pitchen, 1978; Vianna, 1992; Potyondy, 1993), mas usa também uma técnica de decomposição espacial recursiva (Wawrzynek, 1991; Sammet, 1984), no caso um árvore quaternária (*quadtree*), para desenvolver diretrizes locais usadas para definir o tamanho das elementos gerados. A técnica de avanço da fronteira usada neste algoritmo é baseada em um procedimento padrão encontrado na literatura, mas com alguns passos adicionais. Para melhorar a qualidade das malhas geradas (no que diz respeito à forma dos elementos), um procedimento de melhoria local *a posteriori* é usado.

Os dados de entrada do algoritmo são descritos por uma lista de nós, definidos por suas coordenadas, e uma lista com o número de arestas de cada circuito do modelo (circuito

externo e os circuitos internos se existirem). Esse tipo de estrutura de dados tem alguns aspectos a serem considerados: pode representar geometrias de qualquer forma de uma maneira simples, incluindo furos, cavidades e trincas, e pode ser facilmente incorporado em qualquer sistema de elementos finitos.

O algoritmo é organizado nos seguintes passos, que serão descritos nesta seção:

I. Geração da quadtree

- a. Geração baseada na discretização do contorno fornecida***
- b. Refinamento para forçar um tamanho de célula máximo***
- c. Refinamento para forçar disparidade de tamanho mínima entre células adjacentes***

II. Procedimento de avanço da fronteira

- a. Geração de elementos baseado em geometria***
- b. Geração de elementos baseado em topologia***

III. Melhoria local da malha

- a. Suavização Laplaciana***
- b. Avaliação de qualidade e procedimento de “volta-passo” local com eliminação de elementos***

3.1.1 Geração da Quadtree

Como mencionado anteriormente, a finalidade principal da *quadtree* é desenvolver diretrizes locais usadas para definir o tamanho dos elementos triangulares a serem gerados durante o procedimento de avanço da fronteira. A distribuição do tamanho dos elementos triangulares através do domínio é deduzida pelo tamanho das arestas de contorno fornecida como dado de entrada.

A geração da *quadtree* envolve três passos. Em um primeiro passo, a *quadtree* é iniciada baseada nos dados de entrada. Em outros dois passos, a *quadtree* é posteriormente refinada. Um exemplo hipotético bidimensional, mostrado na Figura 3.1, é usado aqui para ilustrar o processo de geração da *quadtree*. Este modelo possui uma trinca no seu lado direito. Na ponta da trinca, o contorno é contraído como se tivesse elementos finitos especialmente colocados. O modelo de contorno apresenta um grau de refinamento gradativamente maior do lado esquerdo para o lado direito.

3.1.1.1 Geração da Quadtree Baseada na Discretização do Contorno

Inicialmente, um quadrado envolvente é criada baseada nas coordenadas cartesianas máximas e mínimas dos nós dos dados de entrada. Esse quadrado é a célula raiz da *quadtree*, conforme ilustrado na Figura 3.2. No primeiro passo da geração da *quadtree*, cada aresta de

contorno é usada para determinar a profundidade local da subdivisão. A célula da *quadtree* contendo o ponto médio de cada aresta de contorno é determinada. Se o tamanho da célula é maior que um fator do comprimento da aresta de contorno, então esta célula é subdividida em quatro células menores. Este processo é repetido recursivamente e acaba quando o tamanho da célula é menor do que o fator do comprimento da aresta de contorno. O procedimento é repetido para todas as arestas do contorno do modelo fornecidas. Com relação ao fator, é recomendado um valor entre 0,7 e 1,4 (Potyondy, 1993). Neste algoritmo adotou-se um fator de 1,0 por recomendação do trabalho de Cavalcante Neto (1994). O resultado está ilustrado na Figura 3.2.

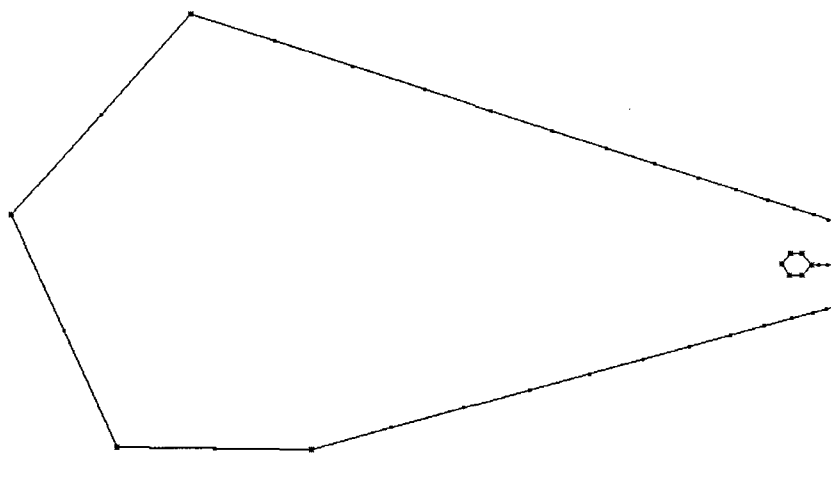


Figura 3.1- Modelo bidimensional e seu refinamento no contorno.

3.1.1.2 Refinamento da Quadtree para Forçar um Tamanho de Célula Máximo

A construção inicial da *quadtree* pode deixar células no interior do domínio com tamanho exageradamente maior do que das células que se situam no contorno. Neste passo, a *quadtree* é refinada para garantir que nenhuma célula no interior do domínio seja maior do que a maior célula no contorno. Isto evitará a geração de elementos excessivamente grandes no interior do domínio. A Figura 3.3 mostra a *quadtree* resultante após esta operação.

3.1.1.3 Refinamento da Quadtree para Forçar Disparidade de Tamanho Mínima

Neste passo, a *quadtree* é processada para forçar um único nível de diferença de profundidade entre células vizinhas. Isto força uma transição natural entre regiões com diferentes graus de refinamento. Esta operação é realizada percorrendo-se a *quadtree* e examinando o nível de refinamento entre células adjacentes. Se a diferença é maior do que um nível, as células apropriadas são refinadas até que o critério seja satisfeito. A Figura 3.4

mostra a *quadtree* gerada após este procedimento. Pode-se notar que as células próximas à ponta da trinca foram afetadas por esta fase, se comparada com a *quadtree* da fase anterior.

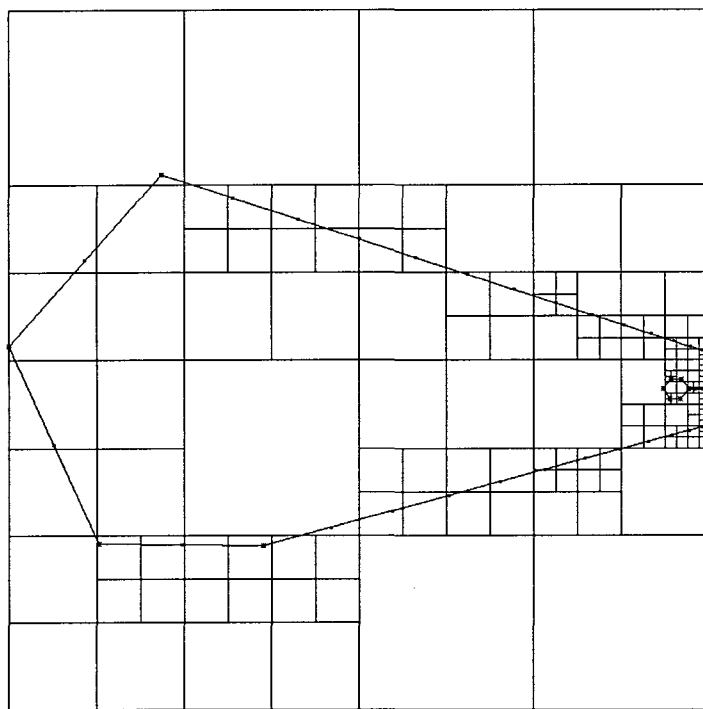


Figura 3.2 - *Quadtree* inicial do modelo.

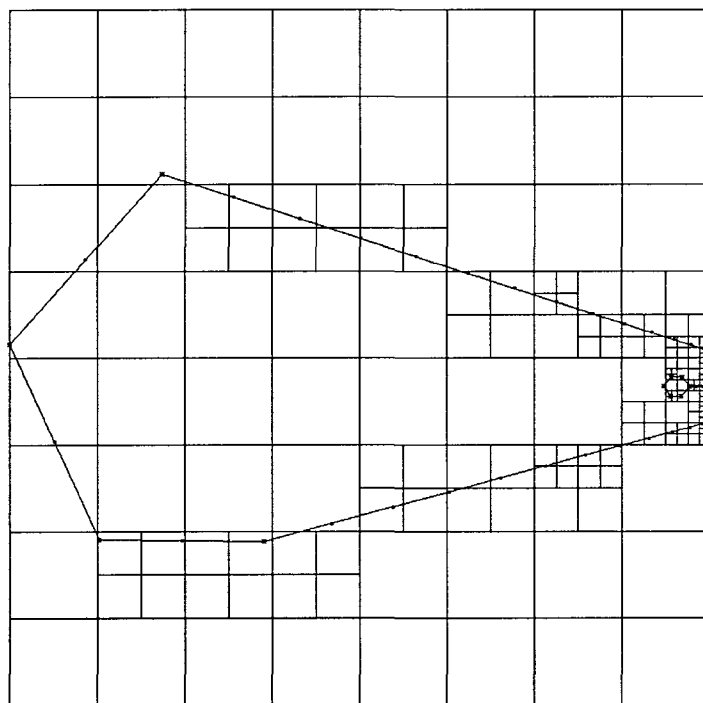


Figura 3.3 - *Quadtree* após forçar maior tamanho de célula do contorno.

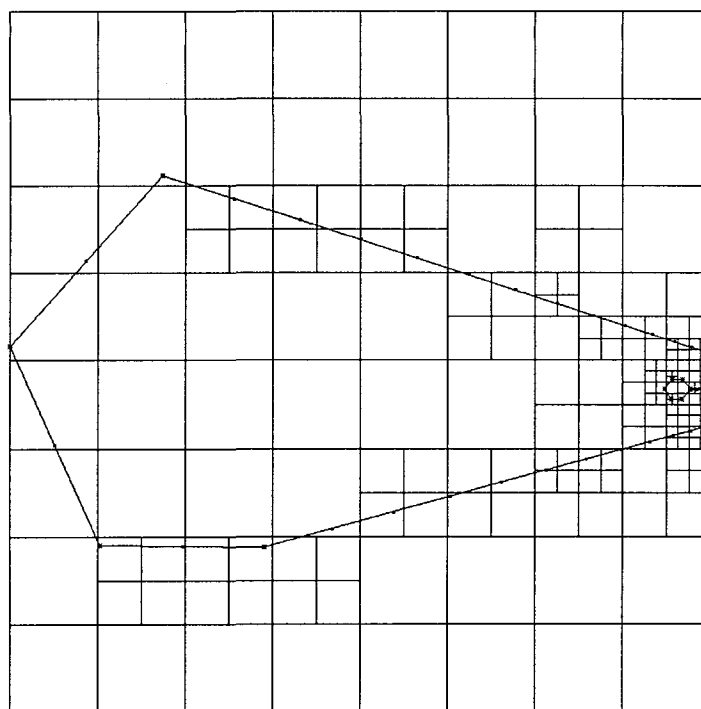


Figura 3.4 - *Quadtree* após forçar disparidade de tamanho mínimo entre células.

3.1.2 Procedimento de Avanço da Fronteira

O procedimento de avanço da fronteira, descrito nesta seção, começa pelo contorno que limita o domínio a ser preenchido com uma malha triangular (termo aqui utilizado para definir uma triangulação). Elementos triangulares são “extraídos” ou “cortados” do domínio um por vez. Quando um elemento é extraído, o contorno limitante é atualizado e o processo é repetido. O procedimento termina quando o domínio inteiro contiver uma malha triangular.

Neste algoritmo, o processo de avanço da fronteira é dividido em duas fases. Em uma primeira fase, uma geração de elementos baseada em geometria é tentada para gerar elementos de formas ótimas. Após esta fase ideal ser exaurida, e elementos ótimos não poderem ser mais criados, passa-se para geração de elementos baseada em topologia, criando elementos válidos na região restante, mas não necessariamente de boa forma.

3.1.2.1 Geração de Elementos baseada em Geometria

Idealmente, a malha no domínio seria inteiramente gerada na fase baseada em geometria. Isto depende da geometria e topologia do contorno do modelo fornecido e está fortemente relacionado com a discretização de contorno fornecida.

3.1.2.1.1 Listas de Contração de Contorno

O processo começa com a criação de uma fronteira de avanço inicial, que é formada pelos segmentos discretizados do contorno fornecido como dado de entrada. Os segmentos do contorno corrente são armazenados em duas listas duplamente encadeadas. A primeira lista é uma lista de arestas ativas, que contém as arestas de contorno que não foram ainda usadas em uma tentativa de gerar triângulos válidos. A outra é uma lista de arestas rejeitadas, isto é, que contém as arestas que falharam na geração de elementos para a fase corrente. Inicialmente, as arestas de contorno fornecidas são inseridas na primeira lista e a segunda lista é vazia.

A lista de arestas ativas é ordenada pelo comprimento das arestas. Isto garante que a primeira aresta selecionada será sempre a aresta com o menor comprimento. Isto tem sido recomendado por outros autores (Peraire et al., 1988) para prevenir que elementos grandes penetrem em regiões com arestas de comprimento pequenos.

Também é conveniente para alguns passos do algoritmo ter uma estrutura de dados adicional que contém as duas arestas do contorno adjacentes a cada nó da fronteira de avanço corrente. Esta estrutura de dados é inicializada para todos os nós do contorno fornecido. A estrutura de dados é atualizada quando o procedimento de contração de contorno progride.

3.1.2.1.2 Geração de Elementos Ótimos

Na fase de geração de elementos baseada em geometria, o contorno corrente avança tentando formar triângulos baseados principalmente em considerações geométricas. A cada passo, uma aresta do contorno, referenciada com *aresta base*, é escolhida da lista de arestas ativas. Esta possui o menor comprimento na fronteira corrente e sua normal aponta para o interior da região a ser triangulada.

Os critérios para gerar um triângulo nesta fase estão explicados com auxílio da Figura 3.5. O procedimento é dividido nos seguintes passos:

- A localização ótima NI para o vértice do triângulo a ser formado é determinada com a ajuda da *quadtree*. A célula da *quadtree* contendo o ponto médio da aresta base é determinada. O ponto ótimo NI se localiza na linha perpendicular a aresta base passando pelo seu ponto médio, M . A distância do ponto ótimo para o ponto médio da aresta base é igual ao tamanho da célula da *quadtree*.
- O ponto ótimo define uma região ótima onde o vértice do triângulo a ser formado está localizado. Essa região é um setor de círculo cujo centro é o ponto ótimo e cujo raio é proporcional ao tamanho da célula da *quadtree*. Na implementação corrente, um fator igual

a 0,85 foi adotado para essa proporção. Esse círculo define um limite superior para a distância entre o vértice alvo do triângulo e o ponto médio da aresta base. Um limite inferior é definido para garantir que o triângulo a ser gerado tenha uma área maior que a menor área aceitável. Nesta implementação este limite inferior é definido pelo triângulo com altura igual a $1/10$ do comprimento da base. A região ótima é usada por duas razões. Primeiro, para assegurar qualidade na forma de todos os elementos gerados e, segundo, para assegurar que novos nós internos somente sejam criados quando é estritamente necessário e sempre em boas posições.

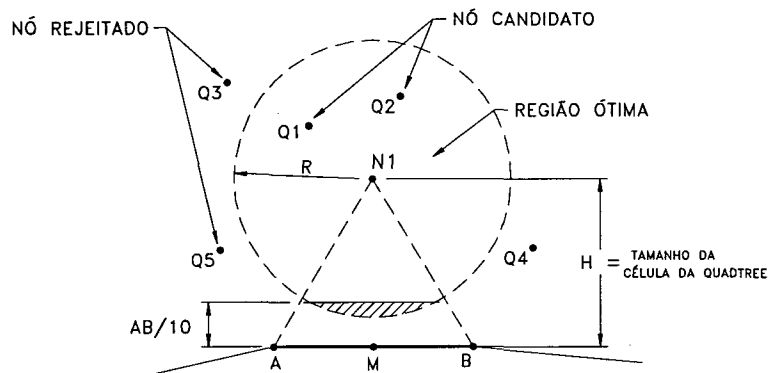


Figura 3.5 - Região ótima de vértice candidato de um triângulo.

- Se nenhum nó existente está dentro da região ótima, um novo nó é inserido na localização ótima $N1$ e um elemento é gerado usando este nó. Se somente um nó existe nesta região, este nó é usado para formar o elemento. Se mais de um nó são encontrados na região ótima, eles são classificados de acordo com o ângulo que eles formarão com a aresta base. Uma lista auxiliar é usada para selecionar eficientemente esses nós. O nó que formará o maior ângulo incluso (veja Figura 3.6) é usado para gerar o elemento.

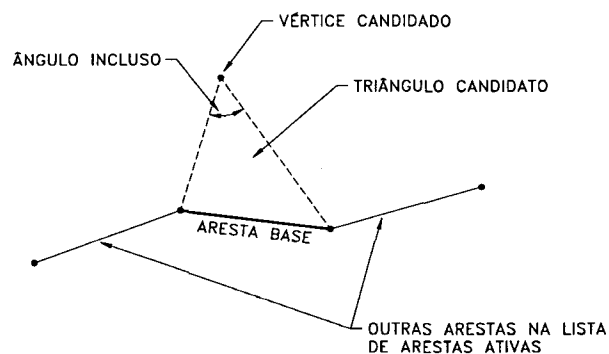


Figura 3.6 - Definição do ângulo para um vértice.

- Testes geométricos adicionais são realizados para assegurar que as arestas do novo

elemento não interceptam nenhuma das arestas existentes na fronteira de avanço. Neste caso, o elemento é rejeitado.

- Como mencionado anteriormente, para problemas de fraturamento poderão existir dois nós com as mesmas coordenadas. A Figura 3.7 ilustra este caso. O algoritmo seleciona o nó apropriado usando um teste simples, que se baseia nas duas arestas de contorno adjacentes ao nó. Quando dois nós candidatos com as mesmas coordenadas localizados na trinca são selecionados para formar um elemento, o nó que se situa no mesmo lado da aresta base com respeito à trinca é escolhido. As normais das arestas da trinca adjacentes aos nós selecionados são usadas para realizar esse teste. O teste assume que as curvas das trincas são suaves (sem mudanças de direção abrupta e sem bifurcação, o que pode ser uma limitação em algumas simulações de problemas de fraturamento).

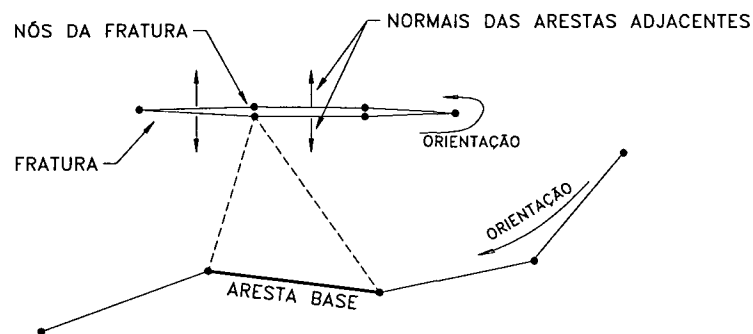


Figura 3.7 - Seleção de um nó candidato em uma trinca.

- Uma vez que um triângulo válido é formado para a aresta base corrente, a lista de arestas ativas é atualizada. Isto é feito através dos seguintes passos. Primeiro, a aresta base é removida da lista de arestas ativas. Então, para as outras arestas do elemento: cada aresta é eliminada se coincide com uma aresta já existente na lista, ou é inserida na lista como uma nova aresta.
- Devido aos limites geométricos impostos pela fase corrente do avanço da fronteira, existem situações em que o algoritmo falha em formar um triângulo válido para a aresta base corrente do contorno. Nestes casos, a aresta base corrente é removida da lista de arestas ativas da contração de contorno e é armazenada na lista separada de faces rejeitadas. Pode acontecer que esta aresta seja removida subsequentemente desta última lista se ela for usada na criação de um triângulo válido para uma aresta base adjacente.
- Quando não existem mais arestas na lista de arestas ativas da contração de contorno, o algoritmo tenta gerar elementos usando as arestas que foram rejeitadas previamente. Pode

acontecer de arestas que falharam previamente na tentativa de formar um elemento possam ser capazes de funcionar agora, uma vez que a fronteira muda com a adição de novas arestas de novos elementos gerados. A fase de geração de elementos baseada em geometria acaba quando não existem mais arestas em ambas as listas da contração de contorno (em cujo caso uma malha ótima foi gerada) ou quando uma aresta rejeitada falha uma segunda vez.

3.1.2.2 Geração de Elementos Baseada em Topologia

O objetivo desta fase do algoritmo é forçar a geração de triângulos válidos, mesmo que os novos elementos não satisfaçam os limites usados na fase anterior para forma dos elementos.

A geração de elementos baseada em topologia começa quando uma aresta de contração do contorno falha duas vezes na tentativa de gerar um elemento ótimo. A lista de arestas de contorno rejeitadas da fase anterior é transformada na lista de arestas ativas de contorno. É sempre possível gerar uma triangulação válida do ponto de vista topológico para qualquer polígono simples (Figueiredo e Carvalho, 1991).

Na fase de geração de elementos baseada em topologia, qualquer nó próximo da face base corrente é selecionado e armazenado em uma lista local de nós candidatos. O nó que tem o maior ângulo com respeito à aresta base é escolhido para a geração do novo triângulo. Se as arestas deste triângulo incluso não interceptarem qualquer outra aresta da fronteira de avanço corrente, o elemento é criado e o contorno é contraído. A fase baseada em topologia acaba quando a lista de arestas ativas do contorno fica vazia.

A Figura 3.8 ilustra a malha gerada após os passos anteriormente descritos.

3.1.3 Melhoria Local da Malha

A última fase do procedimento de avanço da fronteira descrito pode gerar elementos triangulares de forma não-ótima. Esta seção descreve dois procedimentos *a posteriori* para melhorar a qualidade da malha. O primeiro é uma técnica de suavização convencional que consiste no reposicionamento de nós baseada na média das coordenadas nodais. A segunda é um procedimento de “volta-passo”, que remove arestas de elementos de forma ruim para criar uma região onde elementos com melhor forma possam ser gerados.

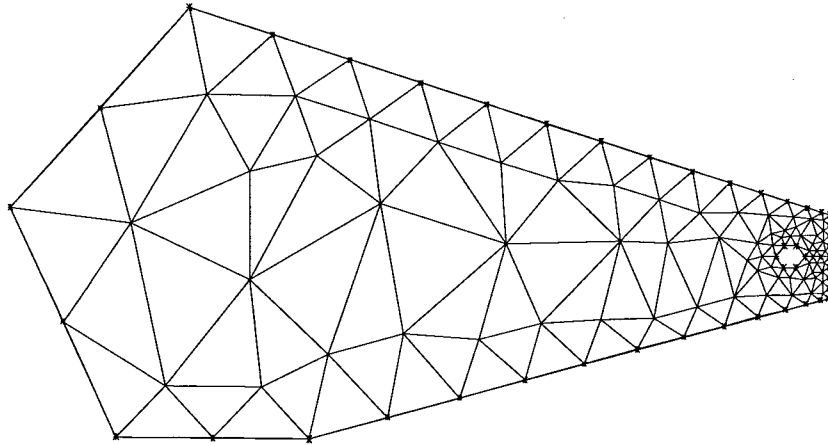


Figura 3.8 - Malha gerada após procedimento de avanço da fronteira.

Os procedimentos de melhoria local implicam que medidas de qualidade da forma dos elementos são necessárias. Neste caso, a métrica de qualidade de forma adotada foi adaptada do caso tridimensional para o bidimensional. Esta métrica é a razão normalizada entre a raiz quadrada média (*root mean square*) dos comprimentos das arestas do triângulo e a área do triângulo. É dada por:

$$\gamma = \frac{S_{rms}^2}{A}, \quad (3.1)$$

onde A é a área do elemento triangular e S_{rms} é dado por:

$$S_{rms} = \sqrt{\frac{1}{3} \sum_{i=1}^3 S_i^2}, \quad (3.2)$$

onde S_i é o comprimento da aresta i do elemento.

Na realidade, a métrica usada neste trabalho é razão γ/γ^* , onde γ^* é a métrica para um elemento equilátero ideal igual a $4/\sqrt{3} \approx 2,3094$. O valor da razão é usado no processo para decidir que elementos serão removidos na malha de elementos triangulares.

Essa métrica é uma adaptação do caso tridimensional que caracteriza a qualidade de um elemento tetraédrico e segue o mesmo comportamento para os testes de sensibilidade feitos por Parthasarath *et al.* (1993). Os testes de sensibilidade para o elemento triangular encontram-se no Apêndice A.

3.1.3.1 Suavização Laplaciana

Uma técnica de suavização é usada para melhorar a qualidade da malha através do reposicionamento de nós dentro de um *patch* (grupo de elementos adjacentes). Uma formulação desta técnica é dada através da equação (3.3), que é a forma genérica de uma função Laplaciana ponderada (Fole *et al.*, 1990):

$$X_0^{n+1} = X_0^n + \phi \frac{\sum_{i=1}^m w_{i0} (X_i^n - X_0^n)}{\sum_{i=1}^m w_{i0}}. \quad (3.3)$$

Na equação (3.3), m é o número de nós conectados ao nó 0 , X_0^{n+1} é a posição do nó 0 com $n+1$ interações de suavização, w_{i0} é a função ponderada entre o nó i e 0 , e ϕ é o parâmetro de relaxação que é normalmente entre 0 e 1 . Neste trabalho, adotou-se $\phi = 1$ e $w_{i0} = 1$ resultando assim em uma média simples das coordenadas dos nós adjacentes. Essa média é repetida quatro ou cinco vezes para dar resultado final satisfatório e afeta somente os vértices interiores, já que não se pode alterar as coordenadas dos vértices originais do contorno.

3.1.3.2 Avaliação da Qualidade e Procedimento de “volta-passo” Local

Em alguns casos a suavização Laplaciana não é suficiente para garantir a qualidade da malha. Neste trabalho, um procedimento de “volta-passo” é adotado para melhorar a qualidade da malha gerada.

O procedimento “volta-passo” é uma proposta de Cavalcante Neto (1998) utilizado na melhoria de malhas volumétricas e que se aplica neste trabalho. O procedimento consiste em remover um elemento que é classificado como de forma ruim e um grupo de elementos na sua vizinhança. A classificação de triângulos ruins é baseada em uma métrica especificada, que neste trabalho é a razão γ/γ^* descrita anteriormente, embora qualquer outra métrica representativa da qualidade dos elementos pudesse ser usada. Para cada elemento da malha gerada, a métrica de qualidade γ/γ^* é avaliada. Se o valor desta métrica está acima de um limite pré-definido, o elemento é classificado como um elemento de forma ruim. O limite a ser adotado é definido empiricamente. Neste trabalho o valor do limite adotado foi $1,50$.

O objetivo do procedimento de “volta-passo” é eliminar arestas de elementos envolvendo um elemento ruim para criar um polígono local que poder ser reconstruído com elementos de melhor forma. Um polígono local para reconstrução é criado por um algoritmo

cujos passos são detalhados a seguir:

- Passo 1: criação de uma lista inicial de nós adjacentes. Para cada elemento que é adjacente a um elemento de forma ruim (Figura 3.9), seus nós são armazenados em uma lista. Um elemento adjacente é aquele que compartilha dois nós (uma aresta) com o elemento ruim.
- Passo 2: criação de uma lista ampliada de nós adjacentes. Para cada elemento do modelo que tem dois nós na lista inicial de nós adjacentes, o terceiro nó do elemento é adicionado na lista se este nó pertence a pelo menos dois destes elementos. Esse passo foi incorporado pois mostrou que resulta em polígonos com melhor forma para reconstrução.
- Passo 3: criação de uma lista de elementos a serem removidos. Estes elementos são definidos como tendo três nós na lista de nós adjacentes ampliada.
- Passo 4: criação do polígono local de reconstrução da malha. Para cada elemento a ser removido, cada uma de suas arestas é inserida no contorno do polígono se ela não pertencer ao contorno, caso contrário a aresta é eliminada.

O algoritmo de contração de contorno descrito nas seções anteriores é aplicado no polígono local, gerando elementos de melhor qualidade em relação aos elementos anteriores ao procedimento de “volta-passo”. A Figura 3.9 ilustra o procedimento para um exemplo.

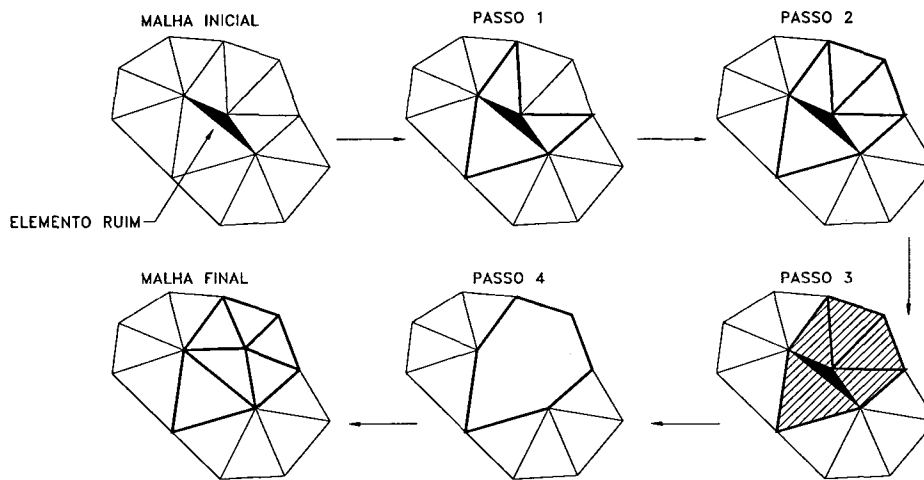


Figura 3.9 - Exemplo do procedimento “volta-passo”.

O procedimento de “volta-passo” seguido pela re-geração dos elementos, analogamente à técnica de suavização, é aplicado uma série de vezes (5 vezes). Neste trabalho, a técnica de suavização foi usada em conjunto com o procedimento de “volta-passo”. A Figura 3.10 ilustra o modelo adotado como exemplo após a melhoria local da malha.

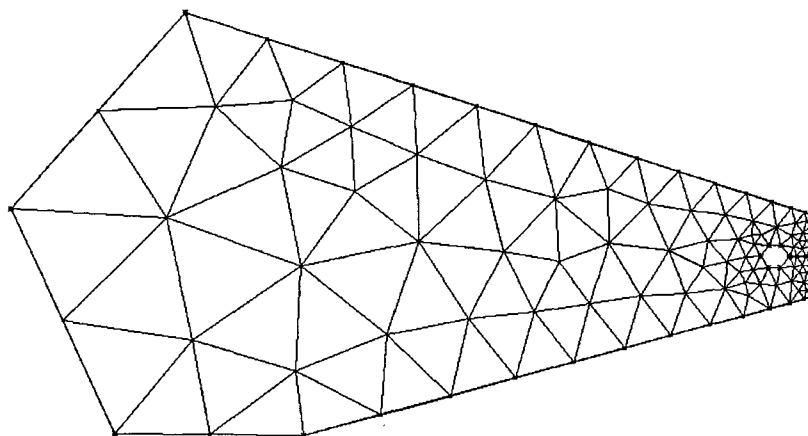


Figura 3.10 - Malha gerada após melhoria local da malha.

3.2 Comparação com Outros Algoritmos Baseados em Quadtree

Nesta seção são feitas comparações entre o algoritmo proposto com outros algoritmos baseados em *quadtree* em relação à qualidade das malhas triangulares geradas e ao tempo de processamento. Os algoritmos utilizados para essa comparação são os de Vianna (1992) e Cavalcante Neto (1994). Esses algoritmos foram desenvolvidos dentro da linha de pesquisa e encontram-se implementados no modelador geométrico *MTool* (1992). O algoritmo proposto também foi implementado no *MTool* para se ter as mesmas condições para avaliação do tempo de processamento.

O tempo medido para cada algoritmo foi feito quando da chamada da função que gera os elementos triangulares e logo após a sua saída. O tempo medido é estritamente do algoritmo, sem computar o processamento que o modelador geométrico gasta para construir a estrutura de dados de entrada e o tempo de restauração dos dados de saída. O computador utilizado nesta operação foi um computador da linha PC (*Personal Computer*) com processador Intel de 200 Megahertz, memória RAM de 32 Megabytes, sistema Windows 95 mono-usuário.

Para avaliar a qualidade das malhas triangulares geradas pelo algoritmo descrito, uma análise pode ser feita baseada na métrica normalizada γ/γ^* adotada. Como explicado no Apêndice A, esta métrica tem um intervalo de valores válidos que variam de 1,0 a infinito ($[1, \infty)$) e o valor ótimo para o triângulo (equilátero) é igual a 1,0. É desejável ter a maioria dos elementos da malha com valores de γ/γ^* próximos a 1,0.

As comparações de forma e de tempo foram feitas para os seguintes modelos: um

quadrado discretizado em 60 arestas com igual comprimento em cada lado, totalizando 240 arestas; um círculo discretizado uniformemente em 120 arestas; uma arquibancada discretizada conforme ilustra a Figura 3.11(a), em que os números mostrados são as divisões do contorno; e um pórtico discretizado uniformemente conforme ilustra a Figura 3.11(b). O número de elementos gerados para cada modelo está apresentado na Tabela 3.1. As Figuras 3.12 a 3.15 mostram as malhas triangulares geradas para os modelos descritos anteriormente.

A Tabela 3.2 apresenta os tempos de processamento e a Tabela 3.3 apresenta o número de elementos gerados por segundo em cada modelo. Pode-se verificar através dessas tabelas que o algoritmo proposto é mais rápido que os outros algoritmos.

Os resultados da análise de forma dos elementos da malha de elementos são apresentados na forma de histogramas através das Figuras 3.16 a 3.19. Nos histogramas, o eixo horizontal representa a métrica normalizada em intervalos de 0,1. O eixo vertical representa o percentual de elementos dentro dos intervalos da métrica.

Em todos os modelos, o algoritmo proposto consegue gerar elementos triangulares com maior qualidade em relação aos algoritmos de Vianna e Cavalcante Neto. Um fato importante a ser mencionado é que o procedimento de “volta-passo” em nenhum dos modelos foi acionado. Verificando os histogramas apresentados, todas as métricas ficaram abaixo de 1,30 (o procedimento de “volta-passo” é feito para os elementos com métrica maior que 1,50) e para o modelo de um quadrado todos os elementos ficaram abaixo da métrica de 1,10. Isto não significa que o procedimento de “volta-passo” é desnecessário, pois foi observado para modelos com maior complexidade geométrica que foram gerados elementos com métrica maior que 1,50.

O algoritmo de triangulação proposto foi também incorporado a um modelador de malhas de superfícies e volumétricas. No Capítulo 5 são mostrados exemplos de malhas volumétricas geradas tendo como ponto de partida malhas de superfície geradas pelo algoritmo proposto.

As Figuras 3.20 a 3.22 mostram as malhas geradas pelos algoritmos utilizando os recursos desenvolvidos no Capítulo 2, para o exemplo do círculo. A medida de distorção utilizada na visualização é a razão com a escala de cores mostrado na Figura 2.14. Verifica-se visualmente que o algoritmo proposto apresenta elementos com menor distorção.

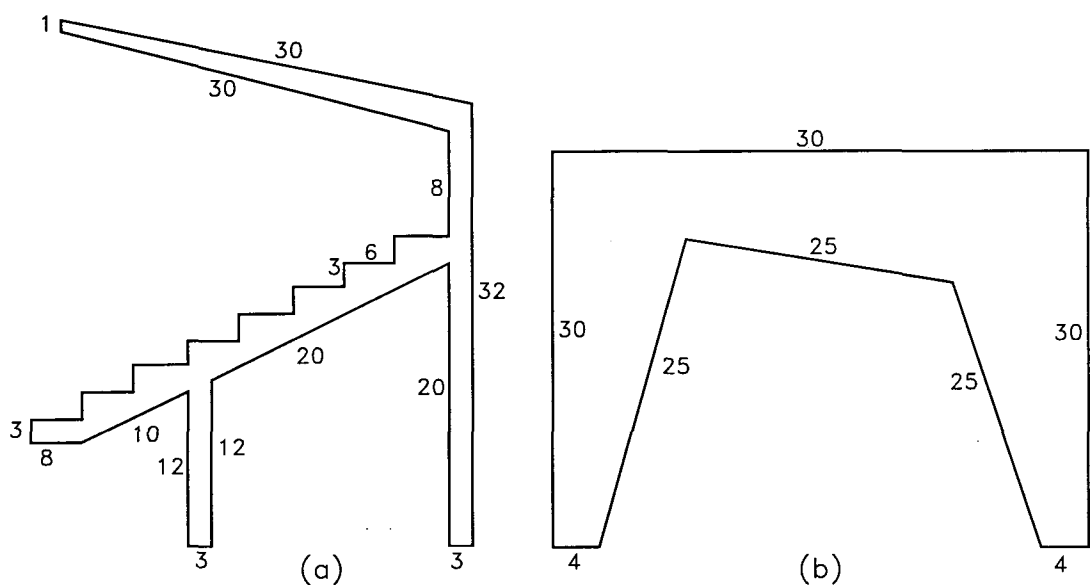


Figura 3.11- Modelos: (a) arquibancada e (b) pórtico.

Tabela 3.1 - Número de elementos gerados para os modelos comparados.

Exemplo	Algoritmo		
	Vianna	Cavalcante Neto	Proposto
Quadrado	8070	1560	6330
Círculo	1670	600	4994
Arquibancada	499	788	751
Pórtico	979	1207	1061

Tabela 3.2 - Tempo de processamento para os modelos comparados (segundos).

Exemplo	Algoritmo		
	Vianna	Cavalcante Neto	Proposto
Quadrado	67,61	4,83	13,62
Círculo	4,34	2,64	8,29
Arquibancada	2,75	13,73	1,70
Pórtico	2,58	9,40	1,59

Tabela 3.3 – Número de elementos gerados por segundo.

Exemplo	Algoritmo		
	Vianna	Cavalcante Neto	Proposto
Quadrado	119.3	323.0	464.7
Círculo	384.8	227.3	602.4
Arquibancada	181.5	57.4	441.8
Pórtico	379.5	128.4	667.3

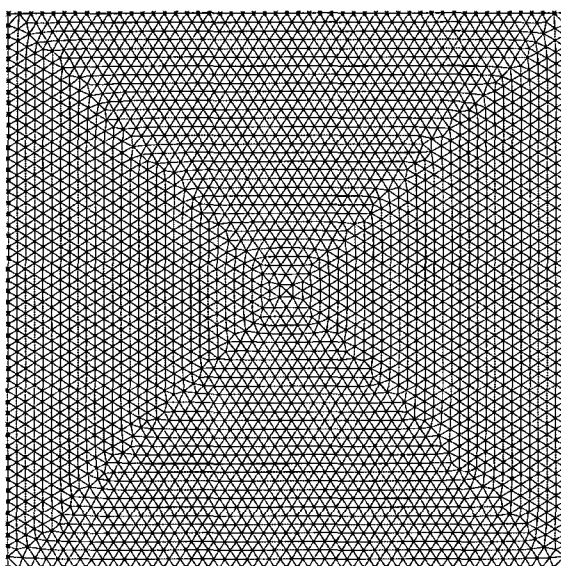


Figura 3.12 - Malha triangular gerada para o exemplo de um quadrado.

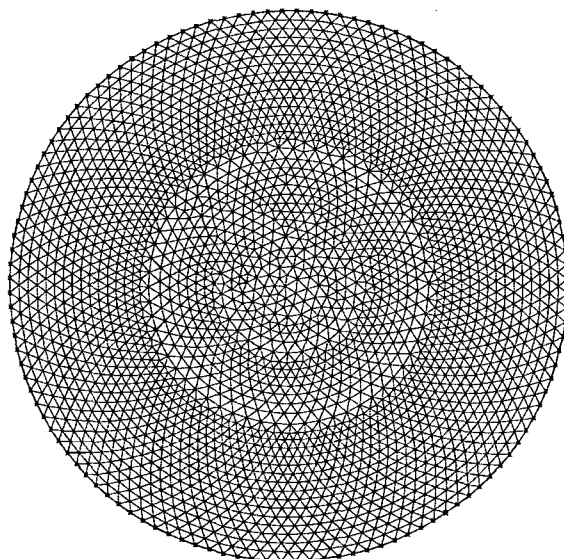


Figura 3.13 - Malha triangular gerada para o exemplo de um círculo.

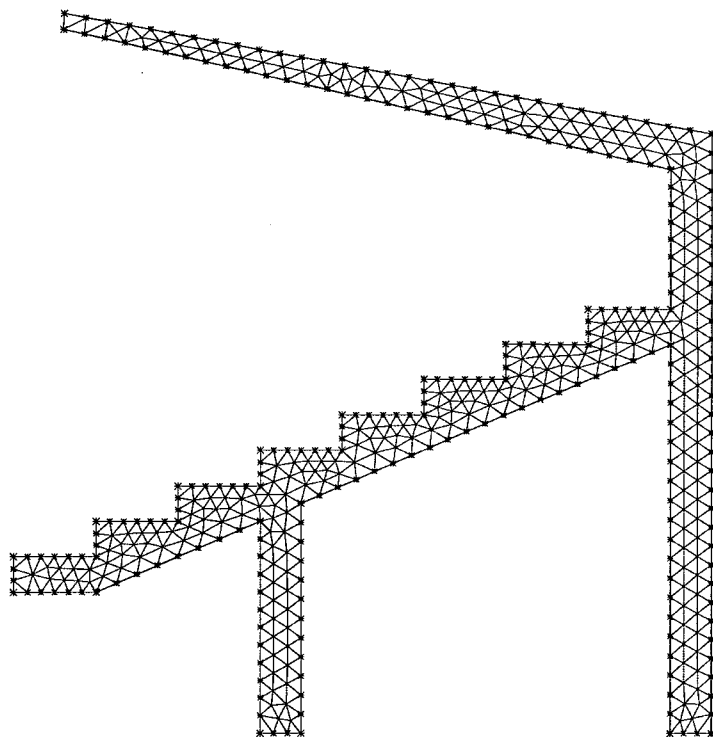


Figura 3.14 - Malha triangular gerada para o exemplo de um arquivancada.

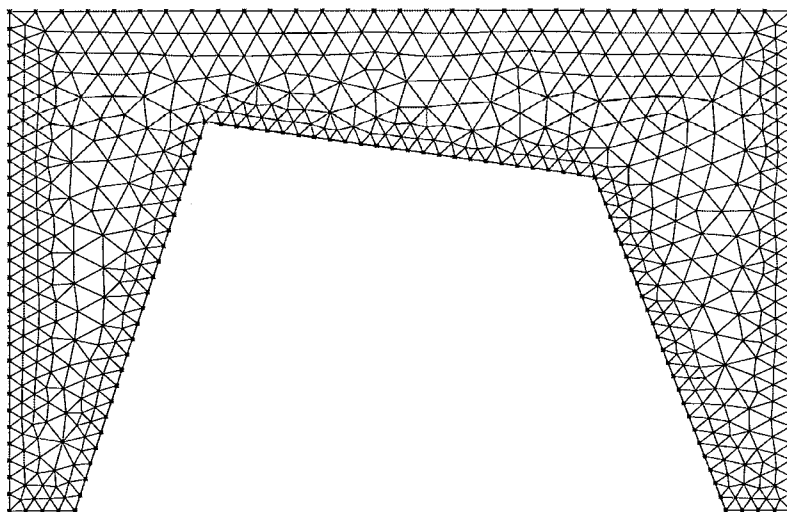


Figura 3.15 - Malha triangular gerada para o exemplo de um pórtico.

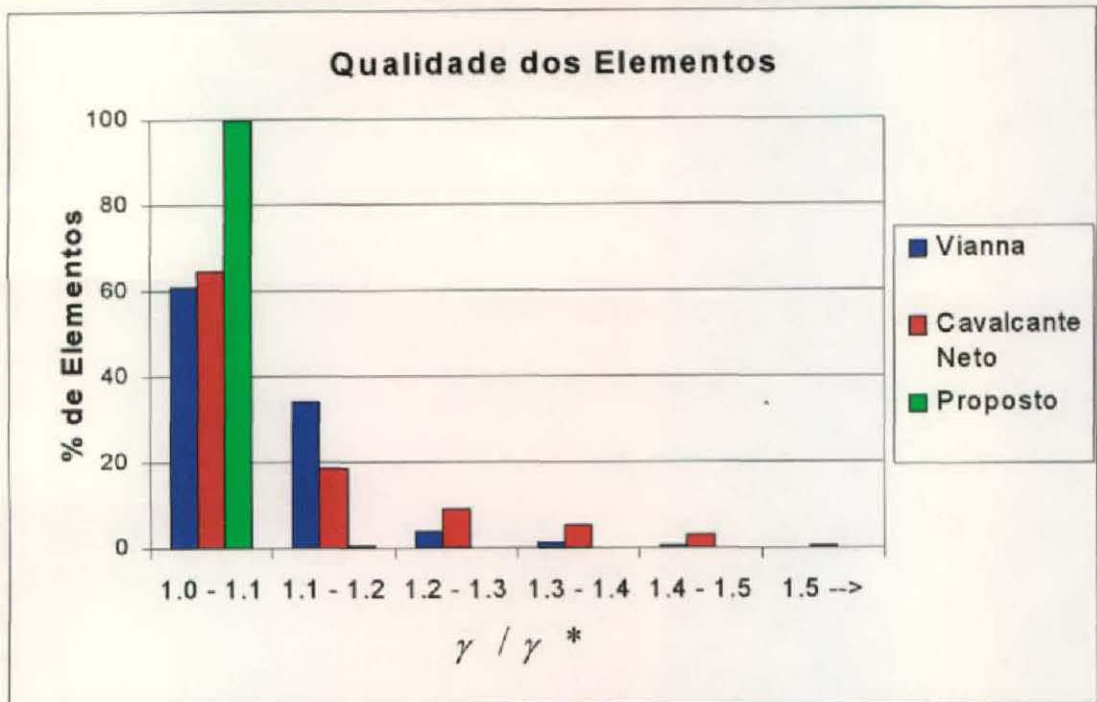


Figura 3.16 - Histograma de análise de forma para o exemplo de um quadrado.

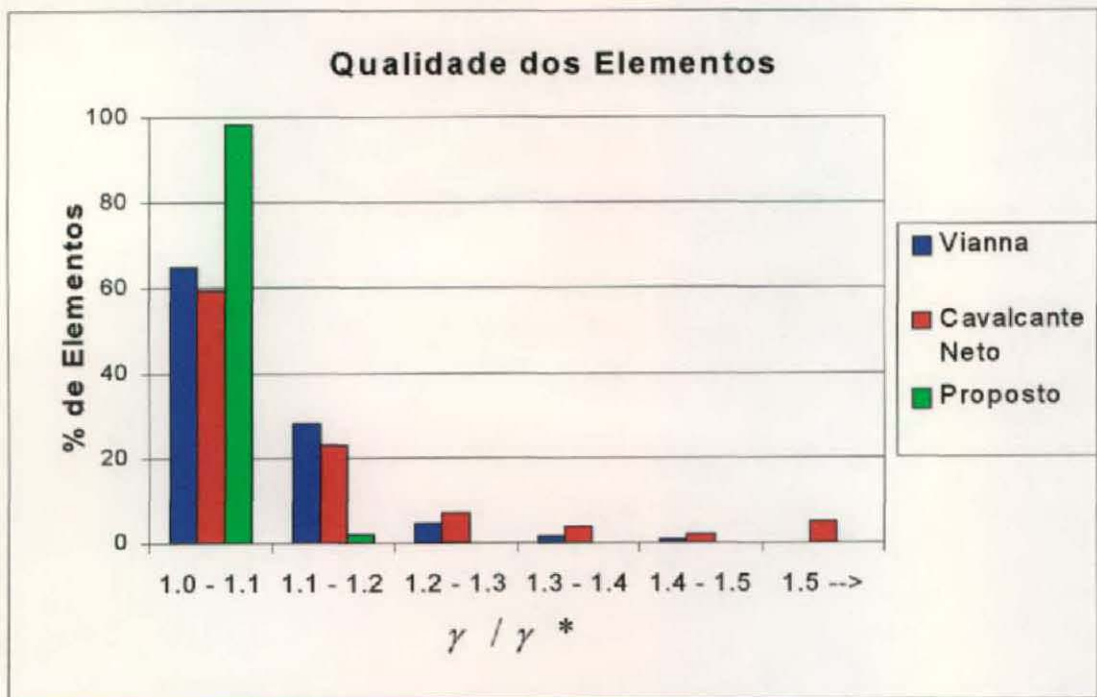


Figura 3.17 - Histograma de análise de forma para o exemplo de um círculo.

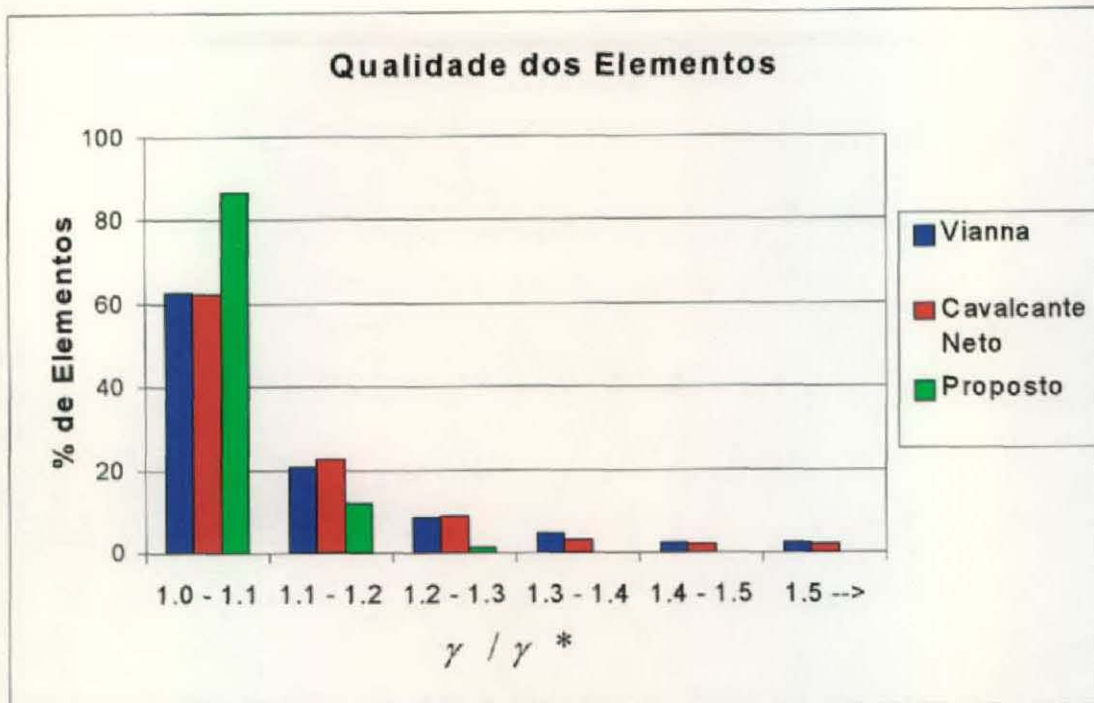


Figura 3.18 - Histograma de análise de forma para o exemplo de uma arquibancada.

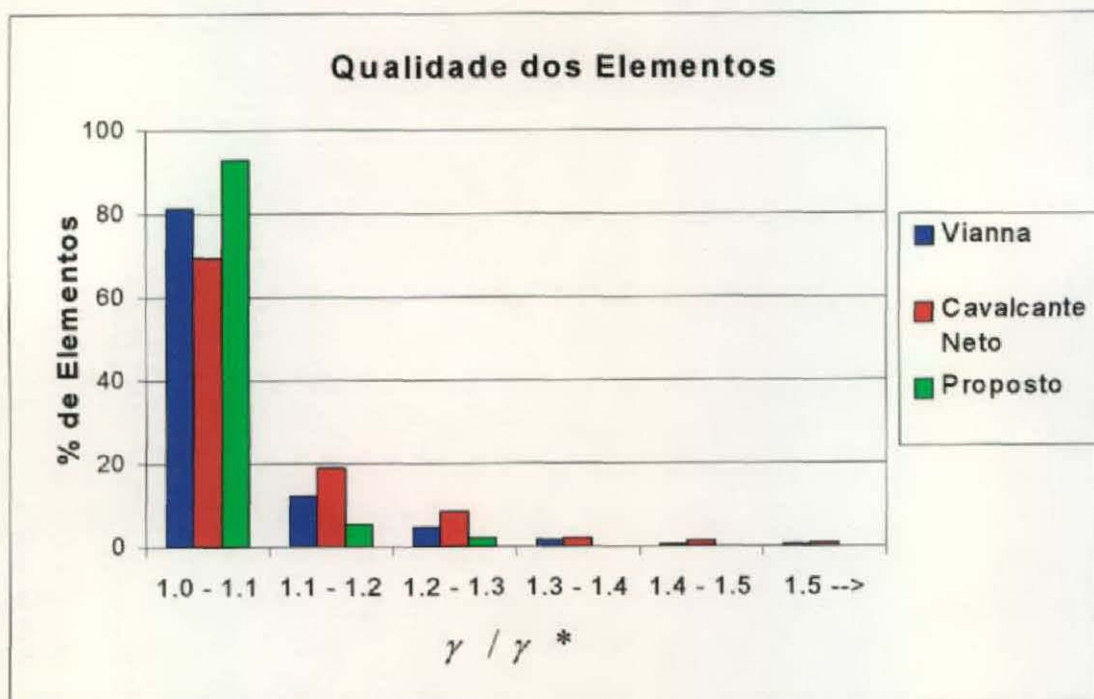


Figura 3.19 - Histograma de análise de forma para o exemplo de um pórtico.

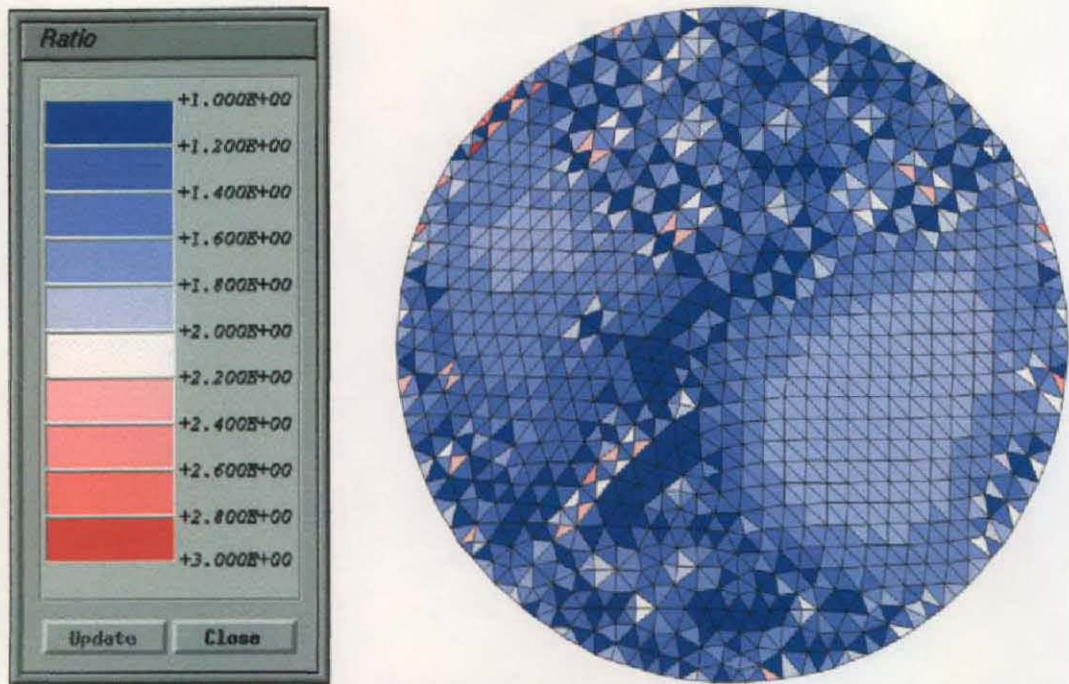


Figura 3.20 - Malha gerada pelo algoritmo de Vianna com representação da distorção razão.

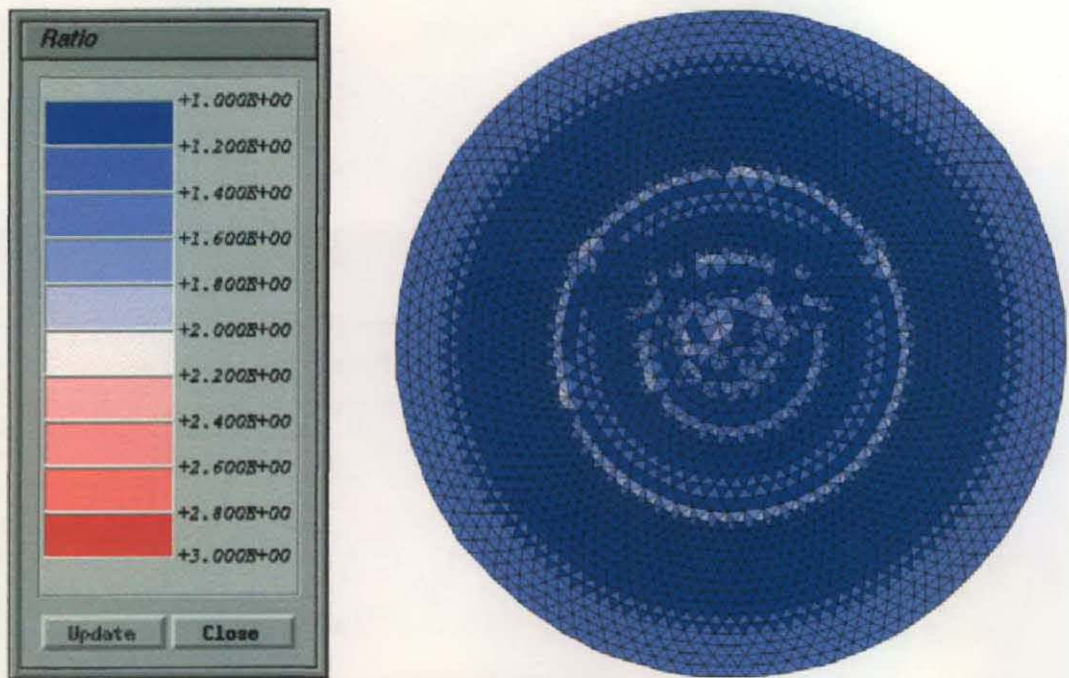


Figura 3.21 - Malha gerada pelo algoritmo proposto com representação da distorção razão.

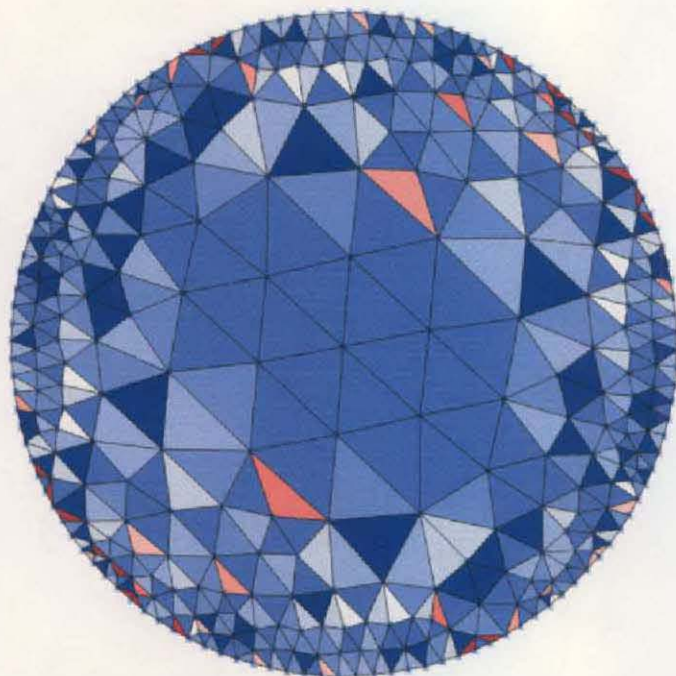
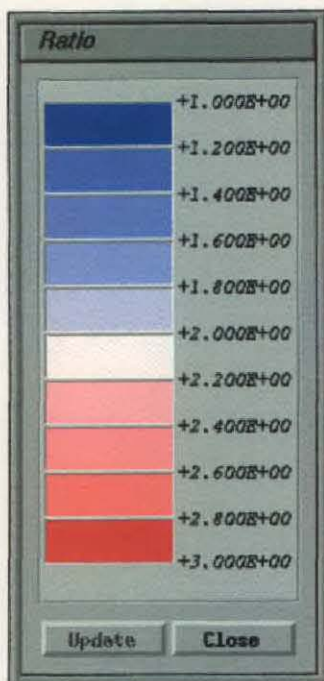


Figura 3.22 - Malha gerada pelo algoritmo de Cavalcante Neto com representação da distorção razão.

4 Malhas Estruturadas em 3D

Técnicas de construção de malhas de elementos finitos podem ser classificadas como malhas estruturadas e malhas não-estruturadas. Malhas estruturadas são aquelas que apresentam uma ordenação espacial uniforme com um arranjo topológico pré-determinado. Por outro lado, as malhas não-estruturadas têm uma ocupação espacial arbitrária. Dentro da construção de malhas estruturadas existem as técnicas de mapeamento que possuem diversas aplicações na modelagem de superfícies e sólidos e na geração de malhas de elementos finitos. Uma das técnicas de mapeamento muito utilizada é o *sweep*. Nesta técnica, a geração de superfícies ou sólidos é feita através do “arrasto” de uma seção transversal bidimensional (ou contorno no caso da geração de superfícies) ao longo de uma trajetória no espaço. O *sweep* mais simples é o translacional, também chamado de extrusão, em que o “arrasto” da área é feito ao longo de uma linha reta. De modo idêntico, o *sweep* rotacional é definido pela rotação de uma seção transversal em torno de um eixo. O *sweep* mais genérico pode ser definido por mudanças de tamanho, forma e orientação da seção transversal combinadas com um “arrasto” ao longo de uma trajetória tridimensional (Hearn e Baker, 1997).

No trabalho de Gomes Coelho (1998) são apresentadas algumas técnicas de construção de malhas de superfícies que podem ser criadas a partir de curvas espaciais, incluindo o *sweep*. No contexto do trabalho de Gomes Coelho, o *sweep* para superfícies é definido por duas curvas: uma define as seções transversais em cada passo e outra a trajetória de arrasto. A idéia deste tipo de *sweep* pode ser estendida para a geração de sólidos.

Um outro tipo mais genérico de geração de malhas de superfícies utilizam as superfícies de Gordon (Farin, 1990) que interpolam um conjunto de curvas em duas direções. Um tipo de abordagem semelhante foi utilizado para gerar malhas sólidas por interpolação entre seções transversais, conforme abordado por Fonseca (1989), ou anteriormente por Peruchio (1982). A interpolação entre seções transversais utiliza uma spline cúbica passando por cada ponto correspondente das seções. Gordon e Hall (1973) definem esse tipo de construção de malha sólida como Mapeamento Transfinito Tridimensional (*MTT*) e que neste trabalho também será tratado com tal.

Com relação as características do trabalho de Fonseca (1989), observa-se que a geração da malha tridimensional é dependente da estrutura de dados do pré-processador gráfico interativo. Essa dependência é indesejável quando há a necessidade de se reutilizar o

código do algoritmo em um outro pré-processador. Com vistas a permitir uma rápida e fácil reutilização do código do algoritmo de *MTT* com diferentes seções transversais, com topologias idênticas, torna-se necessário criar procedimentos que façam reconhecimento dessas seções transversais de modo automático, para posteriormente aplicação da técnica de *MTT*.

Nas seções seguintes são apresentados dois procedimentos para geração de malhas estruturadas de elementos finitos tridimensionais: (1) o *MTT* definido por seções transversais, utilizando-se de uma spline cúbica para interpolação das seções intermediárias; (2) o *sweep* definido por uma seção transversal e uma trajetória no espaço. Para o primeiro procedimento apresenta-se um algoritmo capaz de fazer o reconhecimento automático entre seções transversais, de modo a torná-lo independente da criação das malhas das seções transversais e aplicável em diferentes pré-processadores. O segundo procedimento é uma extensão, para a geração de elementos finitos sólidos, do procedimento de geração de superfícies por *sweep* (Rogers e Adams, 1990; Coelho, 1998). Os procedimentos apresentados fazem parte da Biblioteca Computacional para Geração de Malhas de Elementos Finitos, apresentada no Apêndice B.

Os elementos das seções transversais devem ser triangulares ou quadrilaterais, gerando respectivamente, elementos pentaedrais e hexaedrais. Exemplos de geração de malhas sólidas utilizando essas técnicas são mostradas no Capítulo 5. Apesar dos algoritmos testarem a validade dos dados de entrada, é possível que os elementos gerados sejam inválidos devido a entrada de dados errada para algoritmos. A validade dos elementos deve ser verificadas por um teste *a posteriori*.

4.1 Mapeamento Transfinito Tridimensional

Em muitas aplicações de modelagem são conhecidas seqüências de seções transversais do objeto que se quer modelar. Este aspecto motiva a criação de procedimentos de construção do modelo a partir de suas seções transversais. Esta seção apresenta um procedimento de construção de malhas sólidas através de *MTT* partindo-se de uma seqüência de superfícies discretizadas que podem ter tamanho, forma e orientação distintas, mas que devem ser topologicamente idênticas. O termo “topologicamente idêntica” significa que as seções devem apresentar uma mesma ordenação espacial e ter o mesmo número de elementos, pontos, arestas internas e externas.

A construção da malha sólida se resume em: dado um conjunto de seções transversais discretizadas, criar seções transversais intermediárias e depois relacioná-las para a geração dos elementos sólidos. Entre cada par de seções transversais fornecidas pode-se ou não gerar um número específico de seções transversais intermediárias.

Caso as seções transversais sejam topologicamente idênticas, o problema principal da construção da malha sólida consiste na identificação e relacionamento dos pontos em cada seção transversal para que seja aplicada a técnica de *MTT*. A solução desenvolvida neste trabalho foi feita selecionando um ponto base em cada seção transversal, pertencente a esta, que permite o relacionamento dos outros pontos restantes.

4.1.1 Reconhecimento dos Pontos entre Seções Transversais

Nesta seção descreve-se o algoritmo para fazer o relacionamento entre os pontos das seções transversais. Os seguintes dados são indispensáveis ao algoritmo (vide Figura 4.1):

- Uma série de seções transversais que não se interceptam, discretizadas em elementos triangulares ou quadrilaterais.
- Pontos bases conhecidos em cada seção transversal no qual uma curva *spline* irá percorrer. O ponto base deve se situar no contorno da seção transversal.

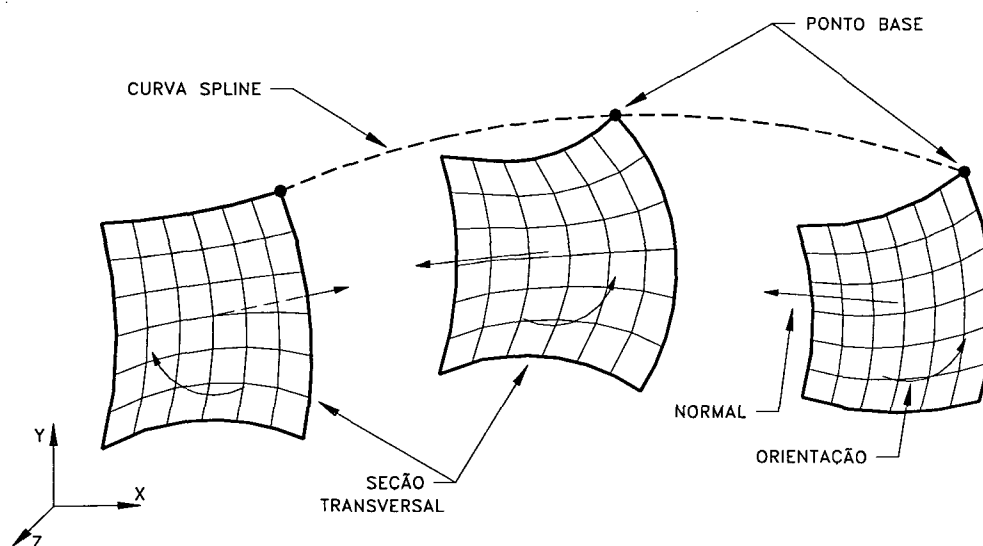


Figura 4.1 - Seções transversais de um modelo sólido.

O primeiro passo do algoritmo consiste em orientar todas as seções transversais em uma mesma direção com respeito à curva *spline*. A idéia básica do algoritmo é criar *listas de arestas* guardando os índices inicial e final dos pontos (nós) das arestas para cada seção

transversal. Se as *listas de arestas* forem construídas de modo idêntico para todas as seções transversais é possível fazer a correlação dos pontos entre seções, quando as *listas de arestas* forem percorridas de forma linear. A construção de cada *lista de arestas* segue os seguintes passos: (a) identificar o elemento que inicia a construção da *lista de arestas*; (b) adicionar à *lista de arestas* as arestas do elemento inicial; e (c) construir o restante da *lista de arestas* percorrendo os elementos adjacentes ao elemento inicial, seguindo recursivamente para os outros elementos adjacentes.

4.1.1.1 Escolha do Elemento Inicial

A partir do ponto base é feita a escolha de um elemento que inicia a construção da *lista de arestas*. Quando o ponto base possui apenas um elemento adjacente, esse elemento é escolhido automaticamente. Caso o ponto base tenha mais de um elemento adjacente escolhe-se o elemento que possui uma de suas arestas pertencentes ao contorno da seção com o ponto base identificado como inicial (“i”) nessa aresta. A Figura 4.2 ilustra um exemplo para a escolha do elemento inicial. Há três elementos adjacentes ao ponto base: “a”, “b” e “c”. Dois elementos têm arestas de contorno (“a” e “c”). Apenas o elemento “c” tem o ponto base identificado como ponto inicial na aresta de contorno do elemento. O elemento “c” é então escolhido como inicial.

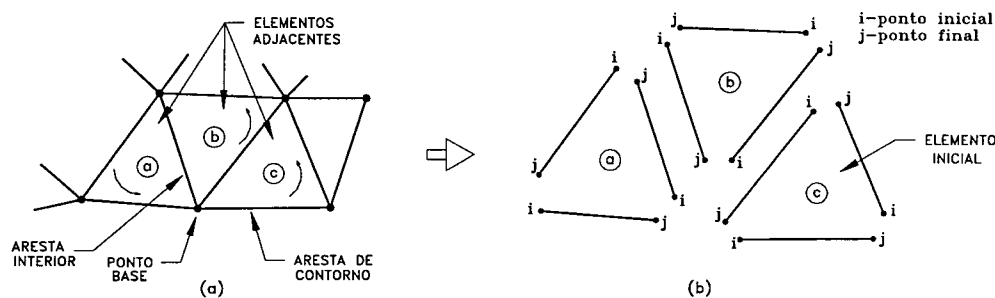


Figura 4.2 - Escolha do elemento inicial para lista de arestas.

Em seções transversais que são topologicamente idênticas, mas com os pontos base identificados em posições incoerentes geometricamente, é possível que os elementos gerados sejam inválidos. Neste trabalho é feito um teste para diminuir a possibilidade de elementos inválidos comparando-se o número de elementos adjacentes ao ponto base em cada seção transversal. Caso estes sejam diferentes o processo não segue. A Figura 4.3 ilustra esse caso. Mesmo considerando seções transversais idênticas topologicamente, os elementos gerados segundo o *MTT* podem ser inválidos geometricamente (auto-interseção, por exemplo). Tais

situações deverão ser identificada com testes *a posteriori*.

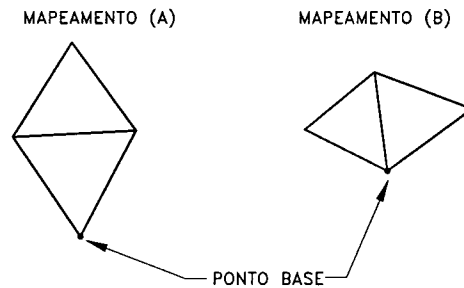


Figura 4.3 – Pontos base selecionados em posições incoerentes.

4.1.1.2 Construção da Lista de Arestas

A partir do elemento inicial escolhido em cada seção transversal faz-se a construção da *lista de arestas* para cada seção. A *lista de arestas* é inicializada com as arestas do elemento inicial, segundo a conectividade deste elemento. A aresta que fica no topo da lista deve pertencer ao contorno e possuir o ponto base identificado como inicial. Esta escolha garante a validade topológica dos elementos gerados, conforme descrito mais adiante. A Figura 4.4 ilustra a construção de uma *lista de arestas* para um elemento inicial.

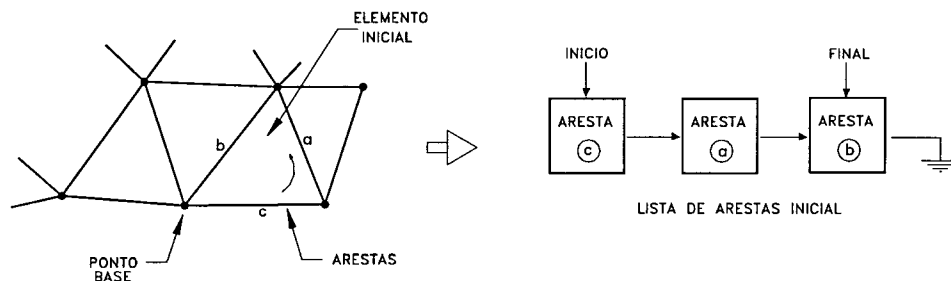


Figura 4.4 - Lista de arestas para o elemento inicial.

As arestas incluídas na lista são “marcadas” para evitar repetições. As inserções seguintes na *lista de arestas* obedecem os seguintes passos:

- (a) A aresta no topo da lista de aresta se torna a aresta corrente.
- (b) O elemento adjacente a aresta corrente é o próximo a ser visitado.
- (c) As arestas adjacentes não marcadas por elementos já visitados são incluídas no final da *lista de arestas*.
- (d) A aresta corrente passa a ser a próxima aresta na *lista de arestas* e o processo é retomado a partir de (b). Quando todos os elementos tiverem sido visitados, e todas as arestas incluídas

na *lista de arestas*, o processo termina.

O processo é repetido para todas as seções transversais produzindo assim um conjunto de *listas de arestas* construídas de modo idêntico.

A Figura 4.5 ilustra uma seção discretizada em quatro elementos quadrilaterais. O ponto base é o ponto identificado como 7. O elemento 3 é o inicial. A seqüência de arestas obtida segundo os passos descritos é a seguinte: *k, i, f, h, l, j, g, d, a, c, e e b*.

Com a seqüência de arestas obtida para todas as seções transversais, identificando também se a aresta é de contorno ou interna, é possível fazer um teste de consistência entre seções. Esse teste é feito comparando o número de arestas de contorno e internas entre as seções transversais. Caso não ocorra igualdade entre os números de arestas significa que as seções não são topologicamente idênticas, não sendo possível continuar com o algoritmo.

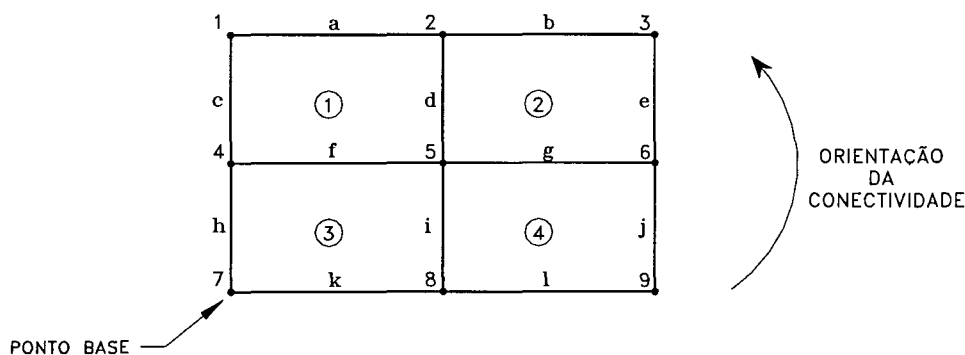


Figura 4.5 - Exemplo para construção da lista de arestas.

4.1.1.3 Obtenção dos Pontos a partir da Lista de Arestas

Os índices dos vértices de cada aresta descrita na seção anterior contêm a informação topológica necessária à obtenção da seqüência de pontos de cada seção transversal. A seqüência de pontos pode ser obtida percorrendo-se a *lista de arestas* do início ao fim, pesquisando-se os pontos da aresta corrente que ainda não foram inseridos em um vetor de pontos (ou em outra estrutura que permita armazenar os pontos seqüencialmente). No fim do processo tem-se uma seqüência de pontos para uma determinada seção transversal. Para o exemplo da Figura 4.5 tem-se a seguinte seqüência de pontos: 7, 8, 5, 4, 9, 6, 2, 1 e 3.

Para todas as seções transversais são obtidos vetores de pontos que são seqüências de pontos montados a partir da listas de arestas. Desta forma, a correlação entre os pontos em cada seção transversal é dada pelo índice no vetor.

4.1.2 Interpolação e Exemplo

Cada conjunto de vértices identificados pelo índice define uma trajetória poligonal usada para interpolar uma curva spline cúbica por trechos (Gomes Coelho, 1998; Foley, 1989). Esta representação de curva contínua permite calcular os pontos das seções transversais intermediárias. Esta formulação também permite que se subdivida as curvas da forma desejada, com trechos igualmente espaçados ou com uma razão qualquer definida *a priori*.

A Figura 4.6 ilustra as seções transversais de um modelo onde a fronteira externa passa da forma quadrada, na seção inicial, para a forma circular, na última seção. A Figura 4.7 ilustra o modelo após aplicada a técnica do mapeamento transfinito.

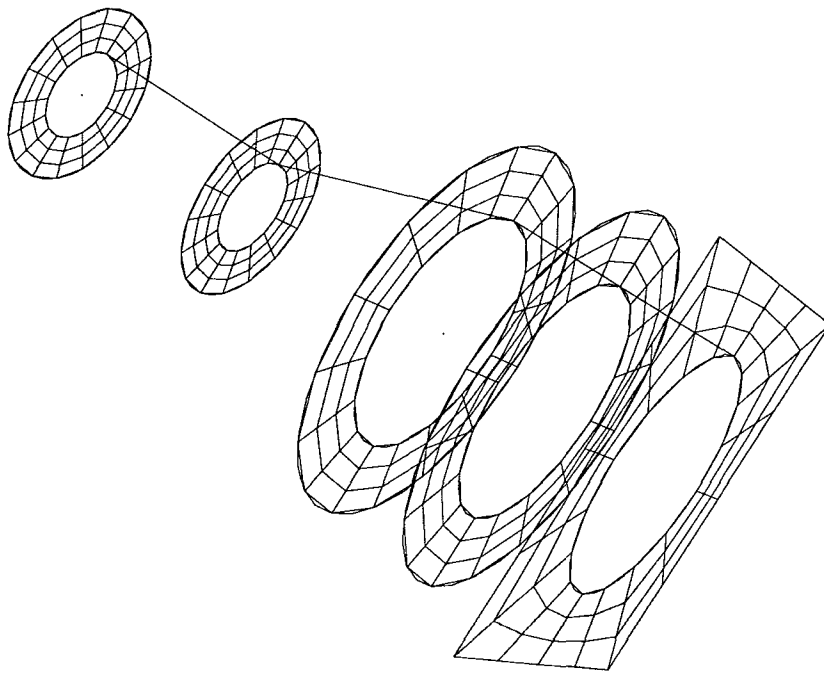


Figura 4.6 - Seções transversais de um exemplo para gerar elementos através de *MTT*.

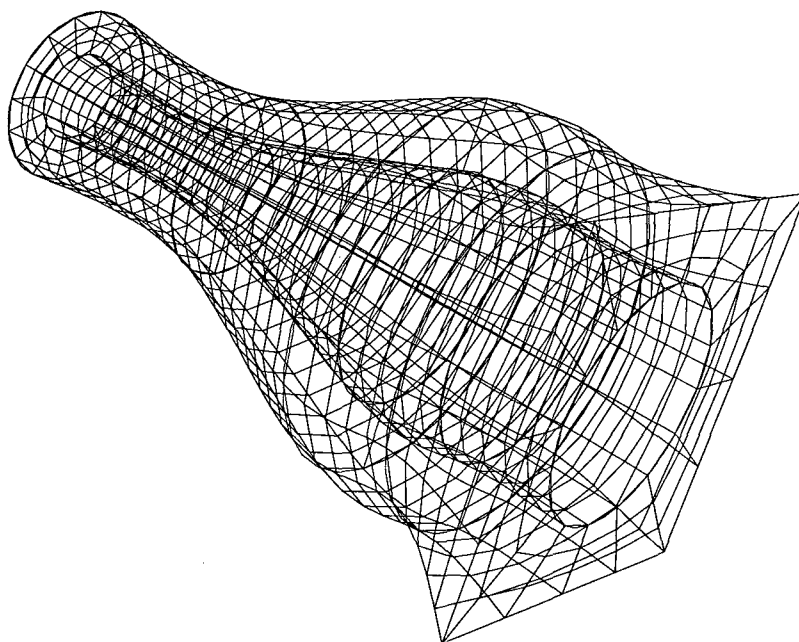


Figura 4.7 - Modelo gerado a partir de suas seções transversais.

4.2 Técnica de Sweep

A outra técnica de geração de malhas sólidas utilizada neste trabalho foi o *sweep* definido por uma seção transversal e uma trajetória no espaço. Partindo-se de uma seção transversal discretizada em triângulos ou quadriláteros e uma curva que define a trajetória do *sweep* (Figura 4.8), deseja-se obter os pontos intermediários dos elementos finitos sólidos (pentaedra ou hexaedra) a partir do “arrasto” da seção transversal ao longo da trajetória. Os pontos intermediários do modelo sólido são obtidos através de translações e rotações, definidos pela curva, aplicados aos pontos da seção transversal discretizada e posteriormente feita a conectividade para cada elemento sólido. Esse procedimento pode ser entendido com a aplicação da técnica de Bloomenthal (Bloomenthal, 1990) para a geração de EF sólidos.

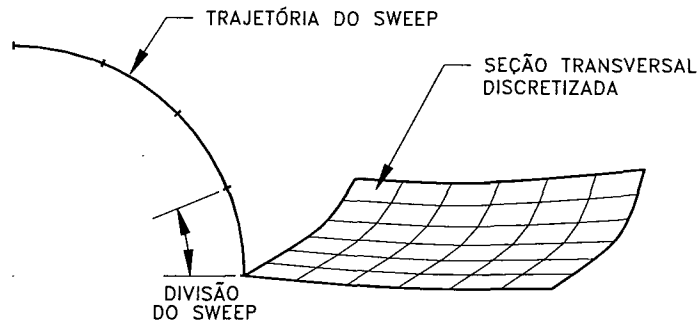


Figura 4.8 - Sweep definido por uma seção transversal e uma trajetória no espaço.

A malha sólida é obtida com os seguintes passos, que são descritos ao longo desta seção:

- Obtenção da orientação de cada seção intermediária.
- Transformação dos pontos da seção transversal para o sistema local de coordenadas.
- Cálculo dos pontos do modelo sólido.

4.2.1 Obtenção da Orientação de cada Seção Intermediária

Os pontos intermediários dos elementos finitos sólidos são obtidos a partir dos pontos da seção transversal discretizada, fazendo-se transformações de translação e rotação nesses pontos. Os elementos necessários para essas transformações são os pontos base, os eixos de orientação da seção transversal fornecida e as seções intermediárias ao longo da trajetória do *sweep* (Figura 4.9). Em cada ponto base são definidos eixos locais, referentes a cada seção transversal intermediária, que juntamente com os pontos bases permitirão a transformação de rotação e translação dos pontos da seção transversal.

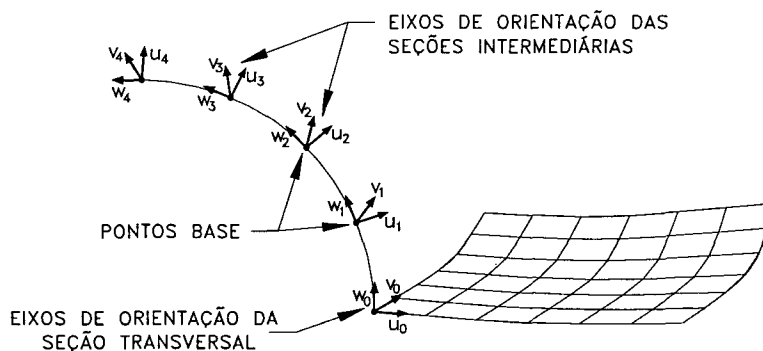


Figura 4.9 - Pontos bases e eixos orientação das seções transversais.

A orientação dos eixos locais das seções intermediárias é definida pelos triedros de Frenet-Serret da curva trajetória (Bloomenthal, 1990), que são definidos de acordo com forma

do objeto que se deseja modelar. No Capítulo 5 é comentada a obtenção dos triedros no contexto de um modelador geométrico. Os vetores tangentes a trajetória do *sweep*, w_i na Figura 4.9, pertencentes aos eixos locais, devem seguir um mesmo sentido da trajetória. A orientação dos outros vetores, u_i e v_i , ortogonais entre si e a w_i , dependem do modelo sólido a ser discretizado. Por exemplo, seja uma seção quadrada no plano xy . Os vetores tangentes à trajetória do *sweep* apontam na direção do eixo global z . Os vetores u e v giram passo a passo em torno do eixo global z em cada seção intermediária. O resultado deste exemplo é uma barra com a seção transversal torcendo-se em torno do eixo z .

4.2.2 Transformação dos Pontos da Seção Transversal para o Sistema Local de Coordenadas

A transformação dos pontos da superfície da seção transversal para o sistema local de coordenadas (mudança de base) possibilita posteriormente a transformação dos pontos para todas as orientações do *sweep*. Essa transformação para o sistema local de coordenadas é feita pela aplicação da matriz de transformação composta de uma translação para a origem do sistema local e de uma rotação para alinhar os eixos globais aos eixos locais.

Em coordenadas homogêneas, onde um ponto i é definido por $[P_i] = [x_i, y_i, z_i, 1]$, a transformação de translação é dada pela matriz:

$$[T] = \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4.1)$$

onde (x_0, y_0, z_0) é a coordenada do ponto base para seção transversal. A matriz de rotação é dada por:

$$[R] = \begin{bmatrix} u_x & u_y & u_z & 0 \\ v_x & v_y & v_z & 0 \\ w_x & w_y & w_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4.2)$$

onde os valores apresentados na matriz são as coordenadas dos vetores definidores do sistema de eixos local no ponto (x_0, y_0, z_0) . A mudança de base pode ser agrupada em uma única matriz de transformação:

$$[M] = [R] [T]. \quad (4.3)$$

Essa transformação é então aplicada para todas as coordenadas que descrevem a seção transversal fornecida no sistema global.

4.2.3 Cálculo dos Pontos da Malha Sólida

Com os pontos da seção transversal no sistema local de coordenadas deve-se fazer uma transformação inversa, isto é, pontos do sistema local para o sistema cartesiano global considerando-se os sistemas locais de cada seção intermediária a ser criada. A transformação aplicada é uma rotação seguida de uma translação para o sistema de global. Para cada seção intermediária da malha sólida, tem-se uma matriz de rotação dada por:

$$[R_i] = \begin{bmatrix} u_{ix} & v_{ix} & w_{ix} & 0 \\ u_{iy} & v_{iy} & w_{iy} & 0 \\ u_{iz} & v_{iz} & w_{iz} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4.4)$$

onde i representa cada seção transversal intermediária. Os valores apresentados na matriz são as coordenadas dos vetores pertencentes ao sistema local. A matriz de translação é dada por:

$$[T_i] = \begin{bmatrix} 1 & 0 & 0 & x_i \\ 0 & 1 & 0 & y_i \\ 0 & 0 & 1 & z_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (4.5)$$

onde $(x_i, y_i$ e $z_i)$ são as coordenadas dos pontos base na trajetória do *sweep* para cada seção transversal. A matriz de transformação pode ser grupada em uma única matriz:

$$[M_i] = [T_i] [R_i]. \quad (4.6)$$

Calcula-se os pontos do modelo sólido pela aplicação das matrizes de transformação aos pontos da seção transversal expressos no sistema local, como ilustra a Figura 4.10. Fazendo-se a devida conectividade dos pontos para geração dos elementos sólidos tem-se o modelo completo como mostra a Figura 4.11.

As Figuras 4.12 e 4.13 mostram a aplicação deste tipo de construção de malhas sólidas por *sweep* para um modelo de uma mola onde a trajetória do *sweep* é uma espiral. A seção transversal é discretizada em elementos triangulares.

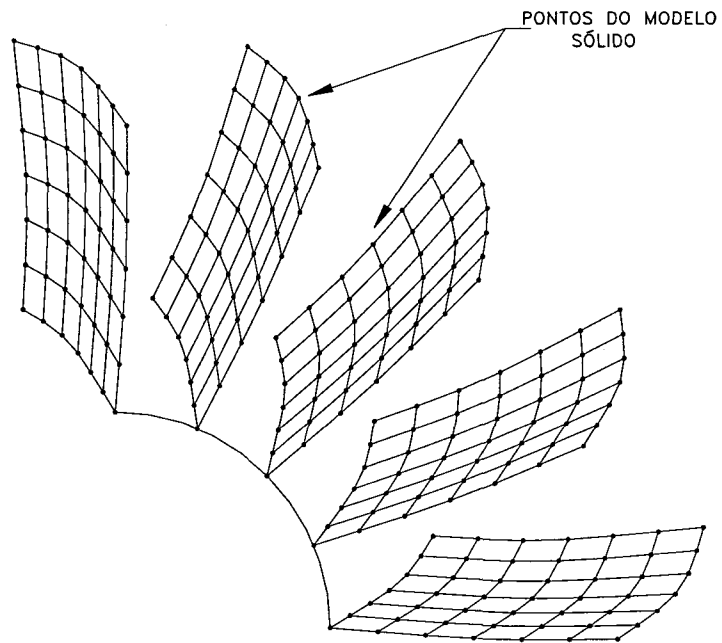


Figura 4.10 - Pontos gerados do modelo sólido.

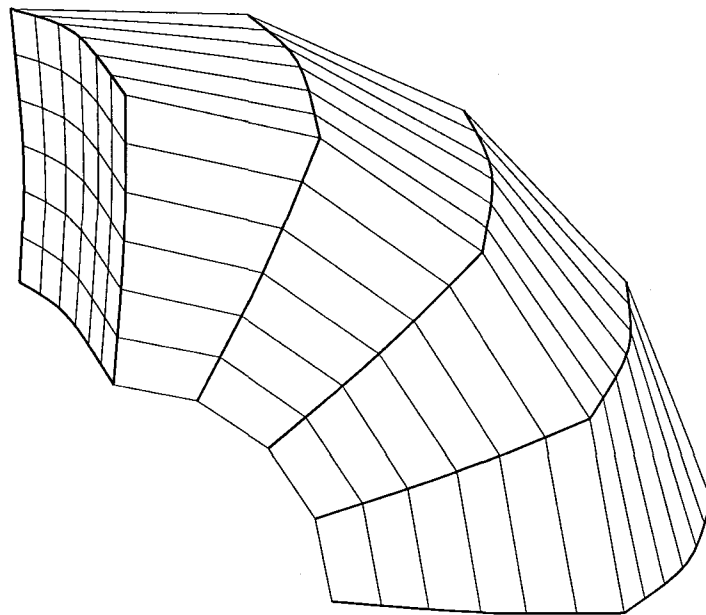


Figura 4.11 - Modelo gerado co *sweep*.

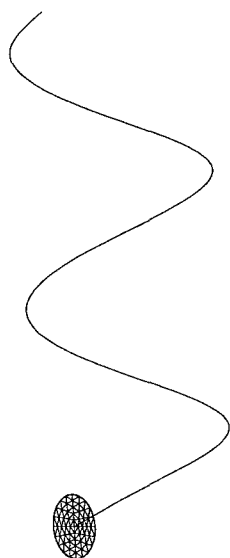


Figura 4.12 - Seção transversal e trajetória do *sweep* para um modelo de uma mola.

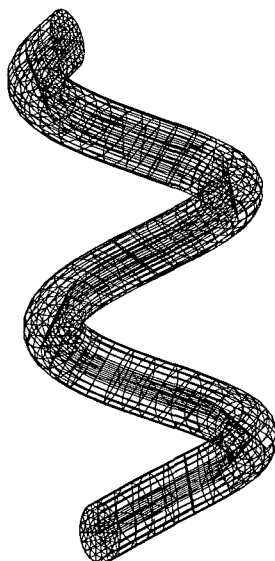


Figura 4.13 - Modelo de uma mola gerada por *sweep*.

5 Modelagem Tridimensional de Elementos Finitos

A idéia principal do Método dos Elementos Finitos (MEF) é subdividir o domínio do problema em pequenas regiões (elementos), onde o comportamento do campo de interesse possa ser aproximado por funções simples, tais como polinômios ou funções harmônicas. Essas funções são expressas pelos valores do campo nos vértices (nós) destes elementos; estes valores, as incógnitas do problema discreto, são determinados pela minimização de um funcional associado às equações diferenciais que governam o problema, resultando na resolução de um sistema de equações. Em geral, o domínio de interesse é tridimensional para a maioria dos problemas, mas várias vezes a simulação é simplificada para duas dimensões, e algumas considerações são necessárias. Uma das razões dessa simplificação é a redução das incógnitas no sistema de equações e conseqüentemente redução do tempo computacional. Outro motivo é a dificuldade de criar um modelo discreto adequado para descrever a geometria e a topologia do domínio tridimensional.

Com o desenvolvimento de novas tecnologias na computação gráfica, como bibliotecas de visualização e manipulação de objetos tridimensionais, os programas de pré e pós-processamento têm permitido modelar com relativa facilidade os modelos tridimensionais em elementos finitos. Essas tecnologias não garantem que as questões de modelagem e simulações tridimensionais estejam resolvidas, pois além disso é necessário que os programas de pré e pós-processamento sejam suficientemente capazes de permitir uma interação com o usuário de modo rápido e confiável. Neste contexto, adotou-se um pré-processador para modelagem tridimensional de elementos finitos que tem demonstrado ao longo dos anos (inclusive remodelado a partir de vários testes-pilotos e de uso em simulações reais) ser capaz de atender eficientemente grande parte das necessidades da modelagem tridimensional: o MG (*Mesh Generator*, 1996).

O MG é um gerador de malhas para modelos tridimensionais de elementos finitos implementado por Gomes Coelho (1998) e proveniente do Convênio PUC-Petrobrás. Neste trabalho, o MG foi estendido para incorporar a geração de malhas sólidas de elementos finitos.

O MG foi desenvolvido usando linguagem C padrão ANSI seguindo uma disciplina de programação orientada a objetos. O programa utiliza as bibliotecas gráficas G3D e V3D (Martha, 1994) e IUP/LED, um *toolkit* portátil para interface com o usuário (Levy, 1993). O

modelo de interface utilizado no MG é resultado de um trabalho de avaliação feito com testes-pilotos de modelagem, discutidas por Gomes Coelho e de Souza (1995). Detalhes sobre as técnicas desenvolvidas e uso do MG encontram-se no trabalho de Gomes Coelho (1998) e no manual do MG (*Mesh Generator*, 1996).

Neste capítulo atenção será dada para a modelagem de elementos sólidos tridimensionais dentro do ambiente do MG, incluindo a geração de malhas estruturadas vista no Capítulo 4 e malhas volumétricas não-estruturadas em volume arbitrário geradas pelo algoritmo de Cavalcante Neto (1998). Esses algoritmos fazem parte da Biblioteca Computacional para Geração de Malhas de Elementos Finitos descrita no Apêndice B.

5.1 Modelagem de Malhas Sólidas

Nesta seção serão descritas as alterações e inclusões feitas no modelador geométrico para suportar de modo eficiente a modelagem de malhas de elementos finitos sólidos. Não é o objetivo desta seção descrever procedimentos do MG já documentados em outros trabalhos. Os seguintes métodos de geração são apresentados:

- Extrusão, que é o *sweep* de uma área ao longo de um segmento reto;
- *Sweep* Curvo, que é o *sweep* de uma área ao longo de uma curva no espaço tridimensional;
- Mapeamento Transfinito Tridimensional, onde o sólido é gerado por interpolação de seções transversais;
- Malha volumétrica genérica, onde o sólido é construído com geração de tetraedros para domínios tridimensionais de forma arbitrária.

Os três primeiros métodos geram elementos pentaedrais e hexaedrais, enquanto que o último método gera elementos tetraédricos. Com o objetivo de incluir os métodos no ambiente do MG, a sua estrutura de dados foi estendida para suportar objetos sólidos. Como o MG se baseia na filosofia de programação orientada a objetos, para incluir os métodos novos bastou acrescentar uma nova classe, chamada de *Sólido*, à estrutura de dados do MG, como ilustra a Figura 5.1 em destaque.

A classe *Sólido* contém as subclasses *Extrus*, *Csweep*, *Mapp* e *Ctetra* que representam os métodos apresentados anteriormente. Assim como todas as outras entidades do MG, onde se tem informações de adjacência entre as entidades *Vértice* e *Curva*, e *Curva* e *Superfície*, também incluiu-se informações de adjacência entre as entidades *Superfície* e *Sólido*. Essas

relações de adjacências são representadas pelas linhas mais espessas na Figura 5.1. Cada superfície possui uma lista de “usos” que contém referências para todos os sólidos adjacentes.

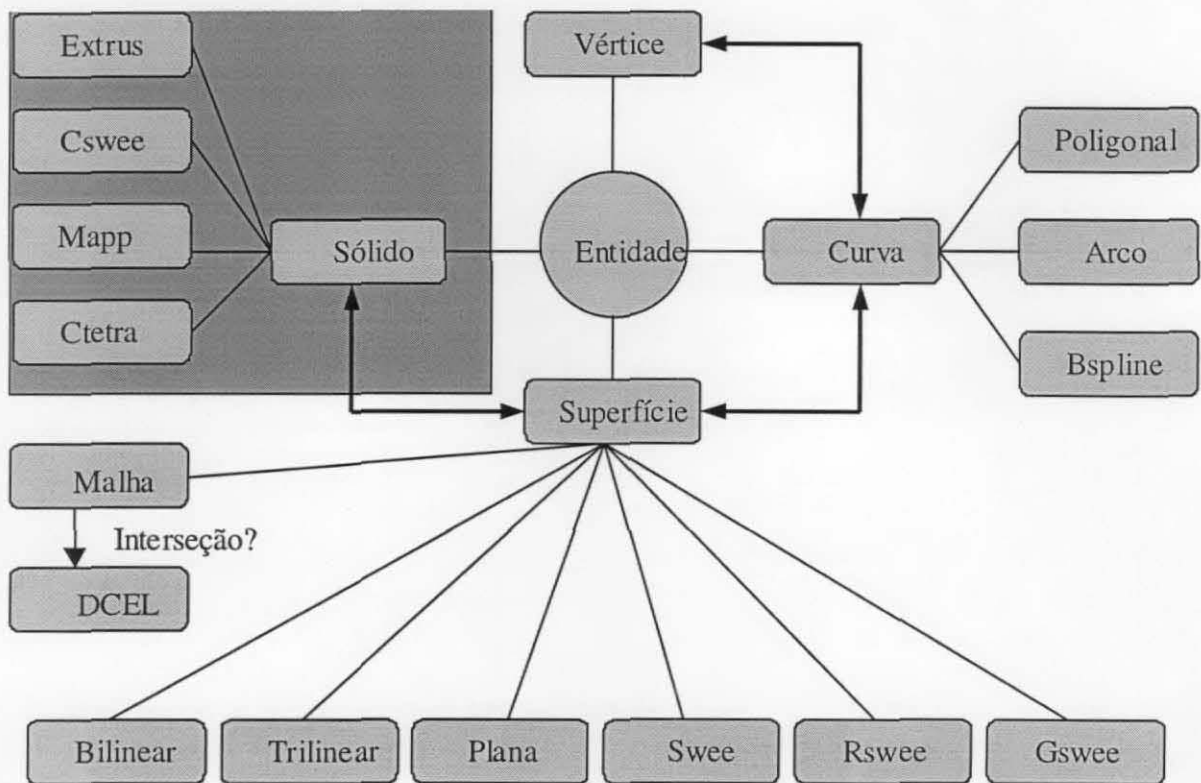


Figura 5.1 - Módulos da estrutura de dados do MG.

Os métodos de construção de sólidos são realizados a partir de entidades da classe *Superfície* e, dependendo da subclasse utilizada para a construção do sólido, poderá também ser necessário entidades da classe *Curva*. Para aplicá-los na classe *Sólido* dois procedimentos tornam-se fundamentais. O primeiro é identificar conjuntos de superfícies adjacentes e suas orientações no espaço tridimensional. O segundo é tornar as superfícies adjacentes em uma única ou várias superfícies desconexas que serão o dado de entrada para as subclasses da classe *Sólido*, sem que haja a repetição dos pontos situados no encontro das superfícies adjacentes. O motivo para realizar esses procedimentos é que o MG trabalha com modelagem de superfícies compostas, enquanto que os algoritmos de geração de malhas sólidas trabalham com superfícies únicas (conjunto de superfícies adjacentes do MG).

O primeiro procedimento é realizado com auxílio das informações de adjacência *Curva-Superfície*. A partir de uma superfície pode-se identificar suas curvas adjacentes. Com essas curvas adjacentes identifica-se suas superfícies adjacentes e o procedimento é repetido

recursivamente até encontrar todas as superfícies adjacentes. Caso se tenha superfícies desconexas o procedimento também é feito para outras porções conexas. Ao mesmo tempo em que é feita a identificação das superfícies adjacentes, são verificadas as orientações entre superfícies adjacentes. Isto permite gerar uma única superfície ou várias superfícies desconexas com uma orientação consistente dos seus elementos, conforme será visto adiante.

O segundo procedimento é resolvido com uso de uma árvore de ordenação espacial B-Tree (Comer, 1978). Essa estrutura permite fazer uma busca rápida de um determinado dado que já se encontra na árvore de ordenação, permitindo assim armazenar somente dados não repetidos na sua estrutura com eficiência. Para o caso das superfícies adjacentes, o que se pretende é não repetir pontos no encontro das superfícies para que se possa obter uma única superfície do conjunto de superfícies adjacentes. A partir da realização dos dois procedimentos tem-se uma única ou várias superfícies desconexas orientadas em um mesmo sentido que podem ser utilizadas para gerar as malhas volumétricas. As sub-seções seguintes resumem os algoritmos de geração de malha sólida.

5.1.1 Extrusão

Extrusão ou *sweep* translacional é o *sweep* de uma área (superfície) ao longo de um segmento reto. Neste tipo de procedimento é necessário fornecer um vetor que indique a direção, comprimento de arrasto, o número de divisões do segmento reto e a malha de superfície que representa a área do *sweep*.

A Figura 5.2 ilustra um chumbador rotulador construído utilizando a técnica de extrusão. A base do modelo, o chumbador, contém 4 furos para parafusos e os rotuladores contêm um furo cada um, que permitem fazer uma ligação rotulada. A Figura 5.3 mostra as superfícies utilizadas para a construção do modelo. As malhas foram criadas com as sub-classes de superfície *Bilinear* e *Plana*. Os elementos sólidos foram criados com a extrusão na direção perpendicular às malhas de superfície.

5.1.2 Sweep Curvo

Este método é o *sweep* em que a superfície geradora faz o “arrasto” ao longo de uma curva no espaço tridimensional. Os dados necessários ao procedimento são: superfície ou superfícies adjacentes e uma curva que representa o caminho do *sweep* ligada a alguma superfície. Com a curva do *sweep* o modelador faz a identificação automática dos triedros de Frebet-Serret (Gomes Coelho, 1998), a partir dos pontos nas divisões da curva, os quais serão necessários para o cálculo dos pontos do modelo tridimensional. O cálculo dos triedros de

Frebet-Serret é feito utilizando os mesmos procedimentos usados no trabalho de Gomes Coelho para o *sweep* de curvas que geram superfícies.

A Figura 5.4 ilustra um bocal gerado por *sweep* curvo. O bocal tem uma parte inicial tronco-cônica com espessura variável seguida de uma parte cilíndrica de espessura constante. Na saída tem-se o frange de conexão à tubulação e na extremidade oposta a chapa de reforço do casco do vaso na abertura para o bocal. Esse modelo é um exemplo mostrado no trabalho de Fonseca (1989) e é construído a partir de mapeamento transfinito tridimensional. No trabalho de Fonseca gastou-se aproximadamente 2 horas de iteração com o usuário para a geração de todo o modelo, enquanto que utilizando o *sweep*, gastou-se aproximadamente 15 minutos. A Figura 5.5(a) mostra a curva de “arrasto” e a superfície geradora. A discretização da seção transversal do bocal foi obtida com um conjunto de superfícies adjacentes geradas pela classe *Bilinear*. Essa mesma seção transversal poderia ser obtida mais rapidamente utilizando uma superfície por triangulação através da classe *Plana*, como mostra a Figura 5.5(b), pois é necessário somente a definição do contorno do bocal para geração da superfície.

5.1.3 Mapeamento Transfinito Tridimensional

O mapeamento transfinito tridimensional é a construção de malha sólida a partir das seções transversais de um modelo a ser discretizado em elementos finitos. As seções transversais devem estar ligadas por uma curva que permite identificar os pontos bases de cada seção transversal e o número de seções intermediárias entre seções, que são necessários para o algoritmo descrito no Capítulo 4. Os pontos bases são os pontos onde as curvas e seções transversais se interceptam. O número de seções intermediárias entre seções transversais é o número de divisões da curva que liga essas seções. O reconhecimento da ordem das seções transversais é feito de modo automático, sem que haja necessidade do usuário informar seqüencialmente as seções.

A Figura 5.6 ilustra um gancho construído por mapeamento. A Figura 5.6 mostra as seções transversais. O modelo tem uma seção transversal inicial circular que muda longitudinalmente de tamanho e forma. A seção circular inicial se torna menor e depois muda para uma seção retangular. Essa seção retangular segue constante e na ponta do gancho a seção transversal torna-se quadrada. Tem-se no total 10 seções transversais que estão ligadas por curvas que definem as seções transversais intermediárias e os pontos bases.

5.1.4 Malha Volumétrica Genérica

Diferente dos métodos apresentados anteriormente que geram malhas estruturadas, a geração da malha volumétrica descrita nesta seção é feita com a aplicação do algoritmo de Cavalcante Neto (1998) que gera malhas não-estruturadas com elementos tetraedricos. A geração da malha sólida é feita a partir da representação da fronteira do sólido, isto é, representação da superfície que define o contorno do modelo discretizada em faces triangulares. Três testes são feitos para validar o contorno do modelo: verificação de superfície fechada, identificação de superfícies exterior e interiores (ocos), e orientação das superfícies.

A verificação se uma dada superfície está fechada é feita considerando que cada aresta do contorno do modelo é adjacente a apenas duas faces. Caso o número de faces adjacentes a uma aresta seja diferente de dois, o MG indica graficamente a aresta inválida, ou arestas inválidas, e não prossegue com a geração da malha. Este teste verifica a validade topológica de uma porção conexa da superfície de um sólido, isto é, é verificado se a porção conexa corresponde a uma variedade de dimensão dois (*2-manifold* - Mäntyla, 1988; Hoffmann, 1989). Entretanto, nenhum teste geométrico é feito para verificar a validade do ponto de vista geométrico (auto-interseção, por exemplo). O usuário faz este teste visualmente.

O problema da identificação de superfícies exteriores e interiores é resolvido calculando o volume de cada sólido a partir da sua superfície de contorno. A superfície de contorno que apresentar maior volume em valor absoluto é considerada como superfície exterior e as demais consideradas como superfícies interiores (ocos). O volume de uma superfície fechada é calculada pela seguinte formula:

$$V = \sum_{i=1}^n \frac{1}{6} (a_i \times b_i) \cdot c_i, \quad (5.1)$$

onde a_i , b_i e c_i são os vetores de uma face i do contorno, definidos pelos pontos desta face com origem em (0,0,0). A operação entre esses vetores é o produto misto. n é o número de faces da superfície fechada. Se as faces da superfícies tiverem orientações que permitam a normais das faces apontarem para o interior do modelo, o valor do volume é negativo. Em caso contrário o valor do volume é positivo.

O algoritmo usado na geração de malhas parte do princípio que as normais das faces, obtidas pela orientação das faces, são direcionadas para o interior do modelo. Para um caso de um modelo que contenha um oco, as faces são direcionadas para o interior do modelo mas

direcionadas para o exterior de seu próprio volume. A orientação das superfícies é feita considerando que o volume da superfície exterior deve ter valor negativo (devido a orientação das faces em que as normais apontam para dentro do modelo) e as superfícies interiores devem ter volume com valor positivo, e caso não atenda essas condições as orientações são corrigidas.

A Figura 5.8 ilustra um fecho de porta de avião com as malhas volumétricas criadas no MG, o qual foi apresentado por Cavalcante Neto (1998) utilizando-se do programa FRANC3D (Martha, 1989; Wawrzynek, 1991). A Figura 5.9 mostra o contorno do sólido que foi criado com superfícies *Bilinear*, *Rsweep* e *Plana*.

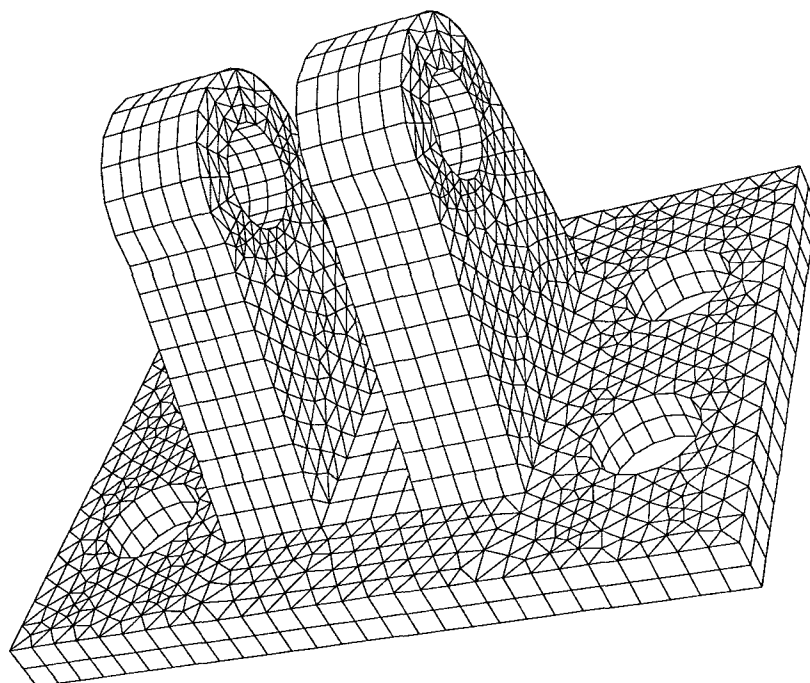


Figura 5.2 - Chumbador rotulador construído através de extrusão.

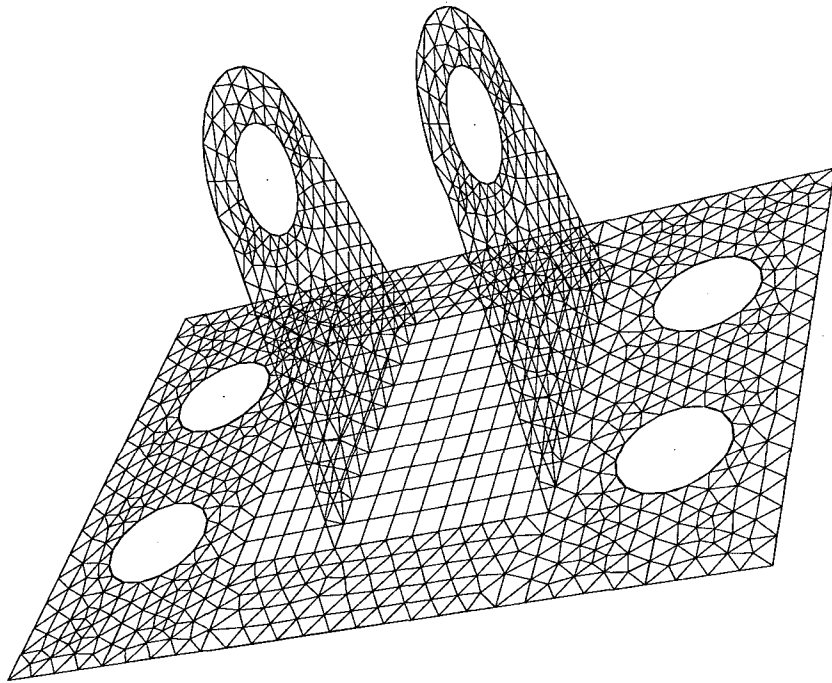


Figura 5.3 - Superfícies que geram o chumbador rotulador.

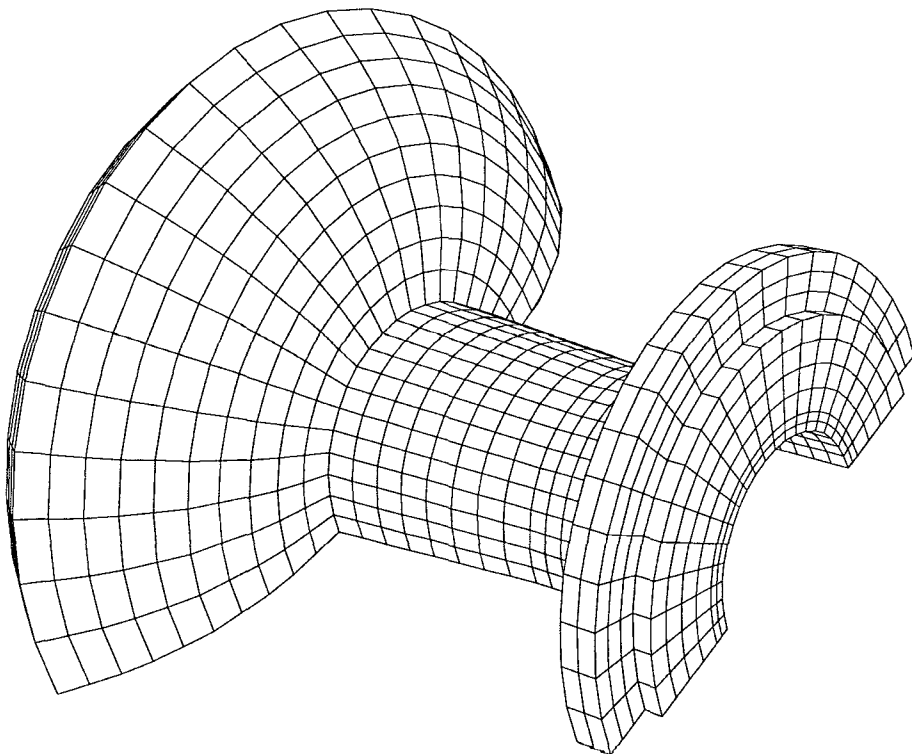


Figura 5.4 - Bocal construído através de *sweep* curvo.

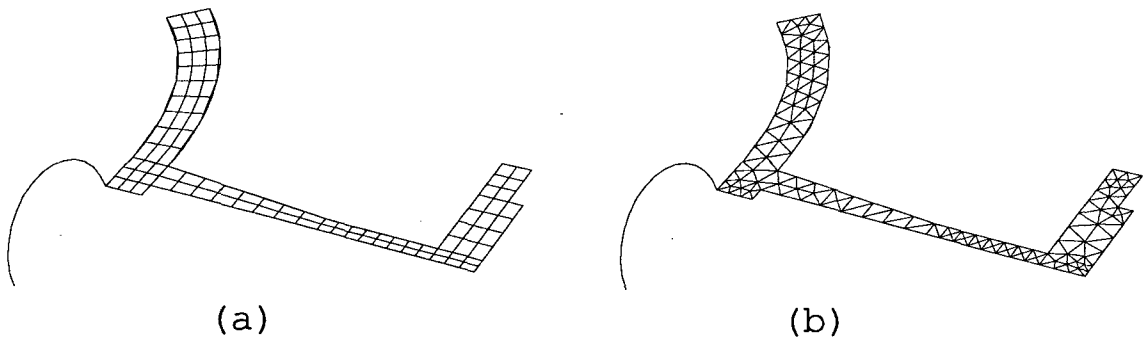


Figura 5.5 - Superfície e caminho do *sweep* para o bocal.

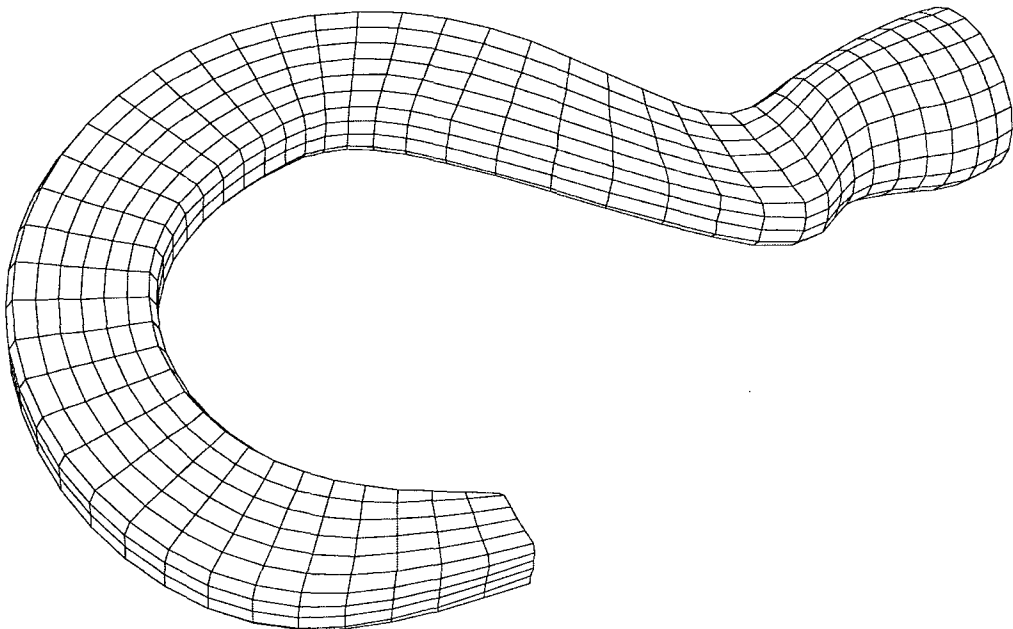


Figura 5.6 - Gancho construído por mapeamento transfinito tridimensional.

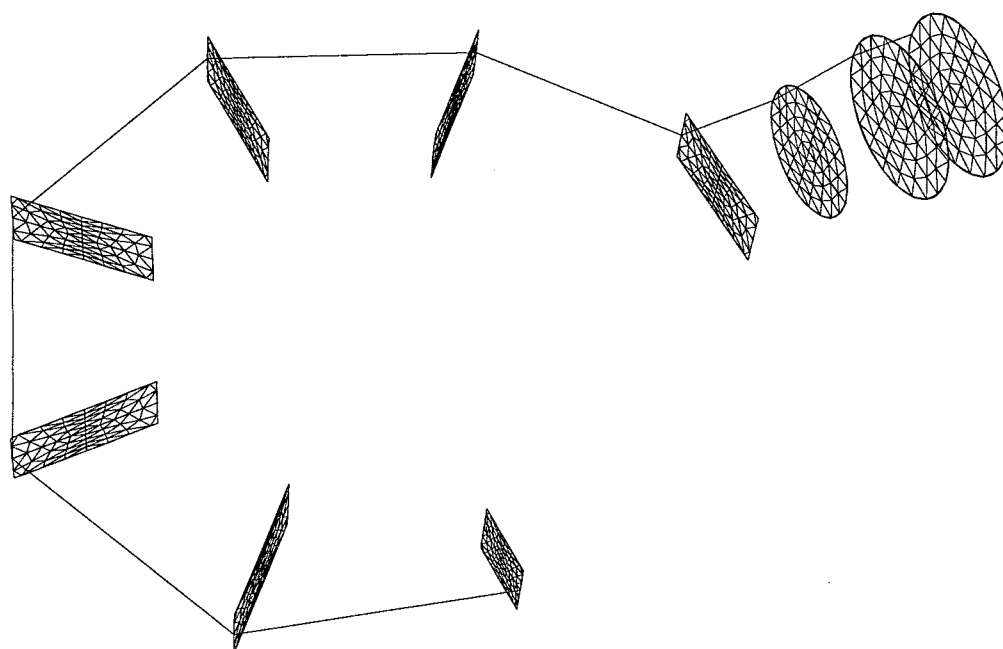


Figura 5.7 - Seções transversais do gancho.

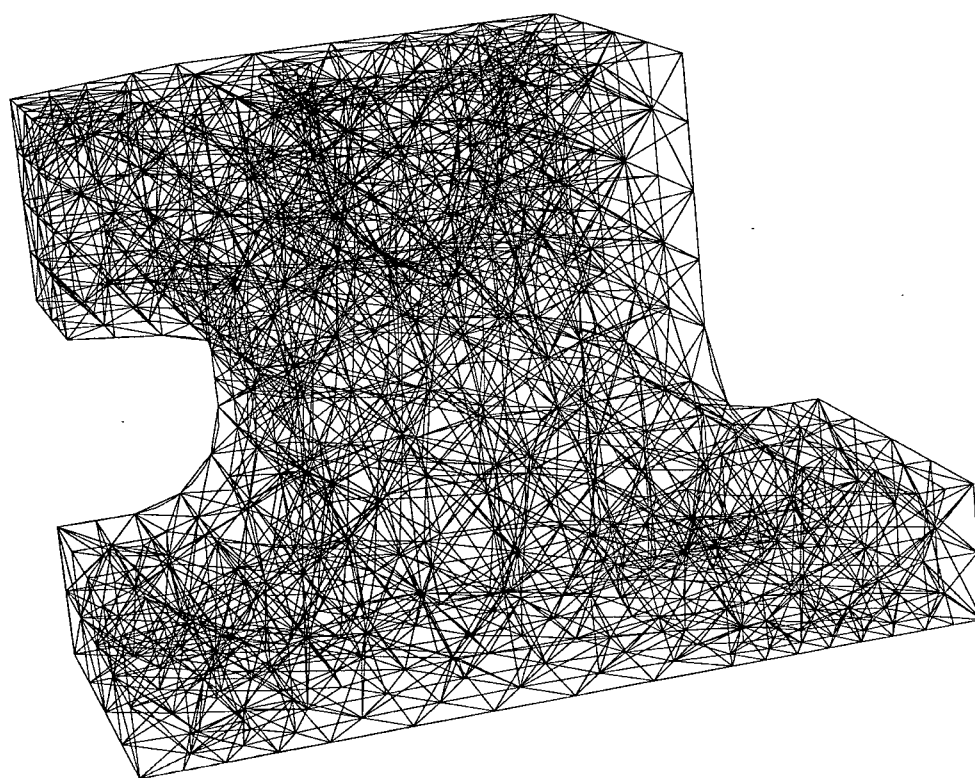


Figura 5.8 - Malha volumétrica em tetraedros de um fecho de porta de avião.

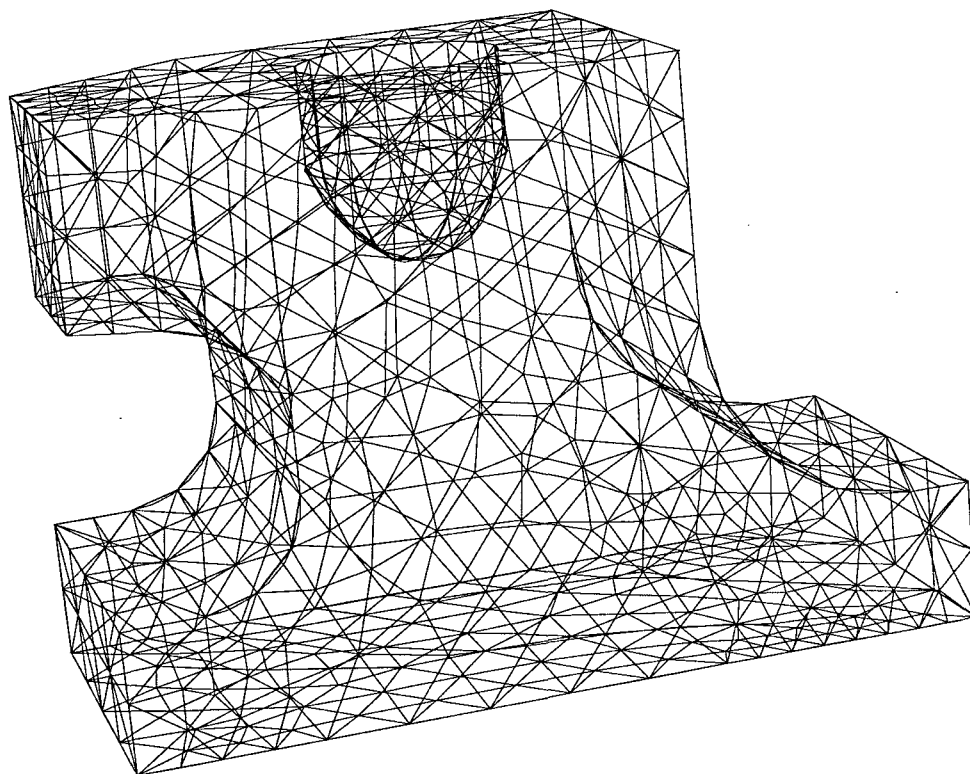


Figura 5.9 - Superfície de contorno do fecho de porta de avião.

5.2 Exemplo de Modelagem Composta

Assim como podem ser feitas construções de superfícies compostas com o MG (tratado por Gomes Coelho, 1998), também podem ser feitas construções de modelos sólidos compostos usando-se superfícies espaciais (cascas) definidos pelo usuário.

Esta seção mostra dois exemplos de modelos tridimensionais criados com malhas de elementos finitos pelo MG. Os modelos foram exportados para arquivo neutro (*Neutral File Format*, 1998) e os atributos foram inseridos manualmente no arquivo, já que o estado atual do MG não suporta a aplicação de atributos. Os modelos foram analisados pelo programa de análise FEMOOP (Martha *et al.*, 1996) para que se pudesse avaliar a validade dos elementos gerados. Entretanto, não é o objetivo aqui avaliar a qualidade dos resultados. A análise gera resultados que podem ser visualizados pelo pós-processador POS3D (Celes Filho *et al.*, 1991).

5.2.1 Corpo-de-Prova

A Figura 5.10 mostra um modelo de corpo de prova retangular muito utilizado em livros de Resistência dos Materiais e Teoria da Elasticidade, para exemplificar zonas de

concentração de tensões. O modelo é constituído de uma barra prismática (cilindro) onde há uma diminuição brusca da seção transversal no meio da barra, e é aplicada uma força de tração nas extremidades. As superfícies utilizadas na construção do modelo estão mostradas na Figura 5.11: são três superfícies da classe *Bilinear*. As extremidades da barra foram construídas com *extrusão* e o afinamento da seção transversal através do *mapeamento transfinito tridimensional*. A Figura 5.12 mostra os elementos gerados internamente. Nesse modelo foram criados apenas elementos hexaédricos (*brick*).

A Figura 5.13 mostra o modelo cortado longitudinalmente com os campos de tensões normais à seção transversal representados graficamente por faixa de cores. É possível observar a alta concentração de tensões, representado em vermelho, estando aparentemente de acordo com a literatura da Teoria da Elasticidade.

5.2.2 Meso-Estrutura

Um outro exemplo de modelagem composta é mostrado na Figura 5.14. Este modelo representa uma meso-estrutura de uma passarela constituída de uma viga superior, um pilar e uma base de apoio. A viga superior suporta um carregamento indicado na figura, aplicado na região sombreada. Ao longo do pilar a seção transversal é variável. A base de apoio tem a sua face inferior considerada engastada na fundação (sombreado).

A Figura 5.15 mostra as superfícies utilizadas para a construção do modelo em elemento finitos. Devido a forma da viga superior ser irregular, adotou-se a geração genérica de elementos tetraédricos no seu interior, bastando para isso definir o contorno da viga com superfícies triangulares. O pilar com seção variável foi construído com mapeamento transfinito tridimensional a partir da definição das seções transversais inicial e final, gerando elementos finitos pentaédricos. A base de apoio foi construído por extrusão das superfícies, gerando elementos pentaédricos para as faces triangulares e elementos hexaédricos para as faces quadrilaterais. Em resumo, tem-se nesse modelo elementos tetraédricos, pentaédricos e hexaédricos, gerados por mapeamento transfinito tridimensional, extrusão e volume arbitrário.

A Figura 5.16 mostra o modelo com as malhas geradas e a Figura 5.17 mostra o campo de tensões normais na direção do eixo x .

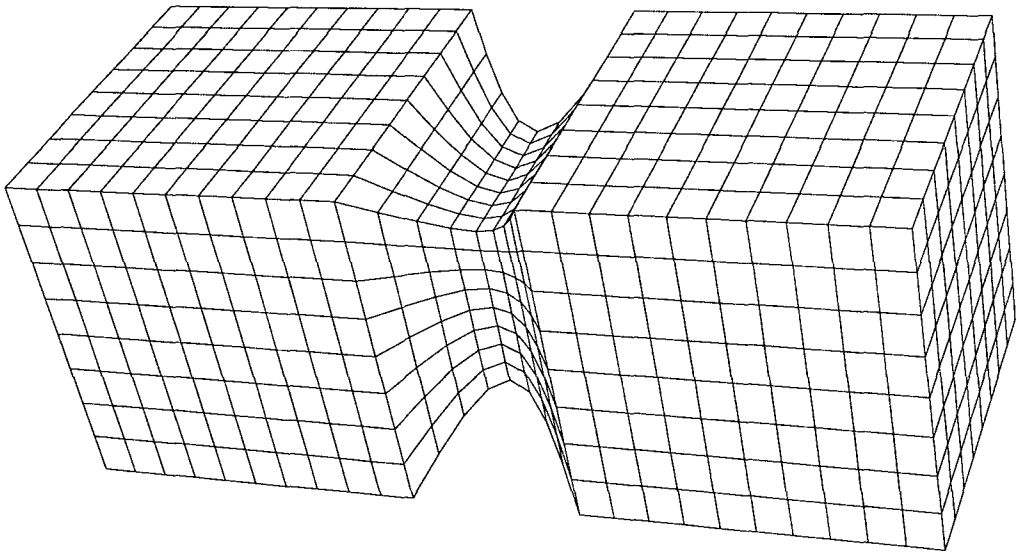


Figura 5.10 - Modelo de um corpo-de-prova.

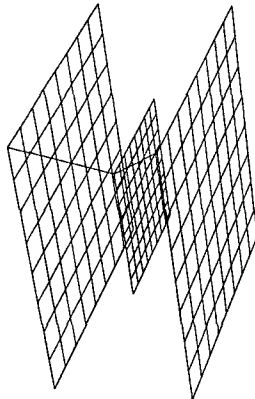


Figura 5.11 - Superfícies bases para o corpo-de-prova.

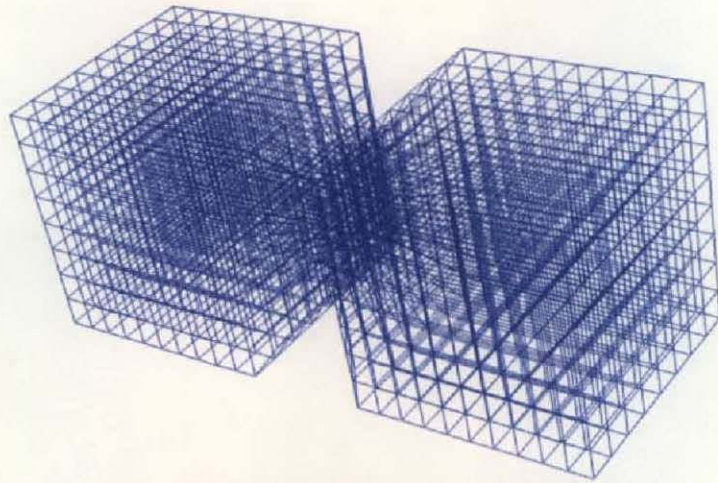


Figura 5.12 - Malhas geradas para o corpo-de-prova.

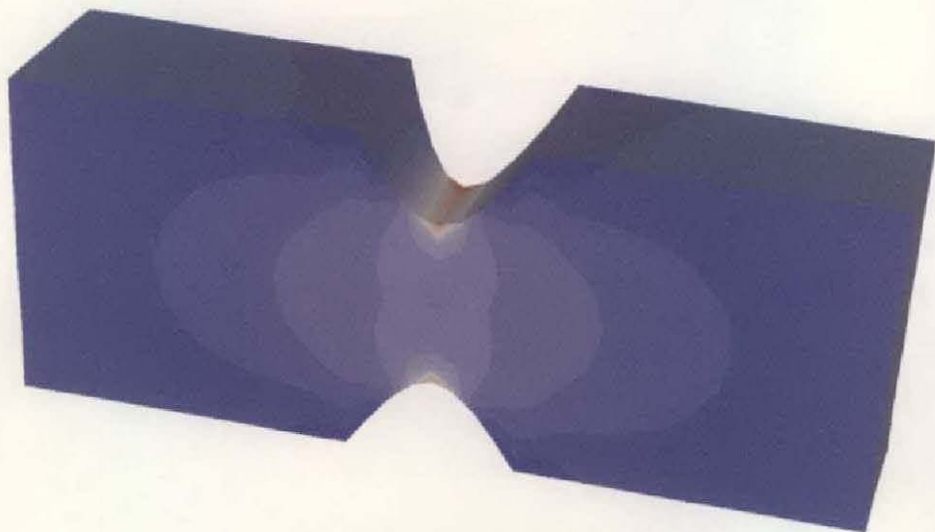


Figura 5.13 - Campos de tensões normais à seção transversal do corpo-de-prova.

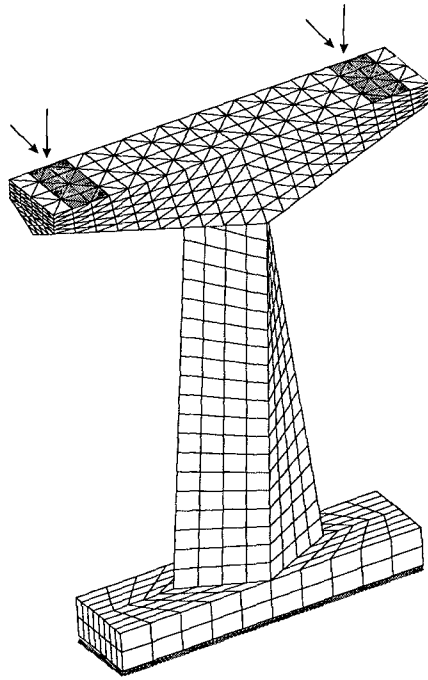


Figura 5.14 - Modelo de uma meso-estrutura de passarela.

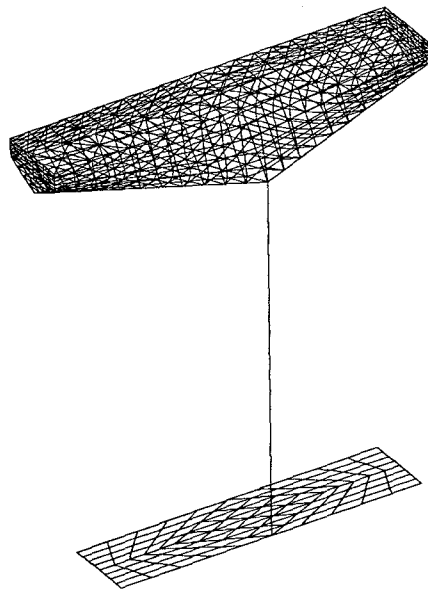


Figura 5.15 - Superfícies utilizadas para gerar a meso-estrutura.



Figura 5.16 - Malhas geradas para a meso-estrutura.

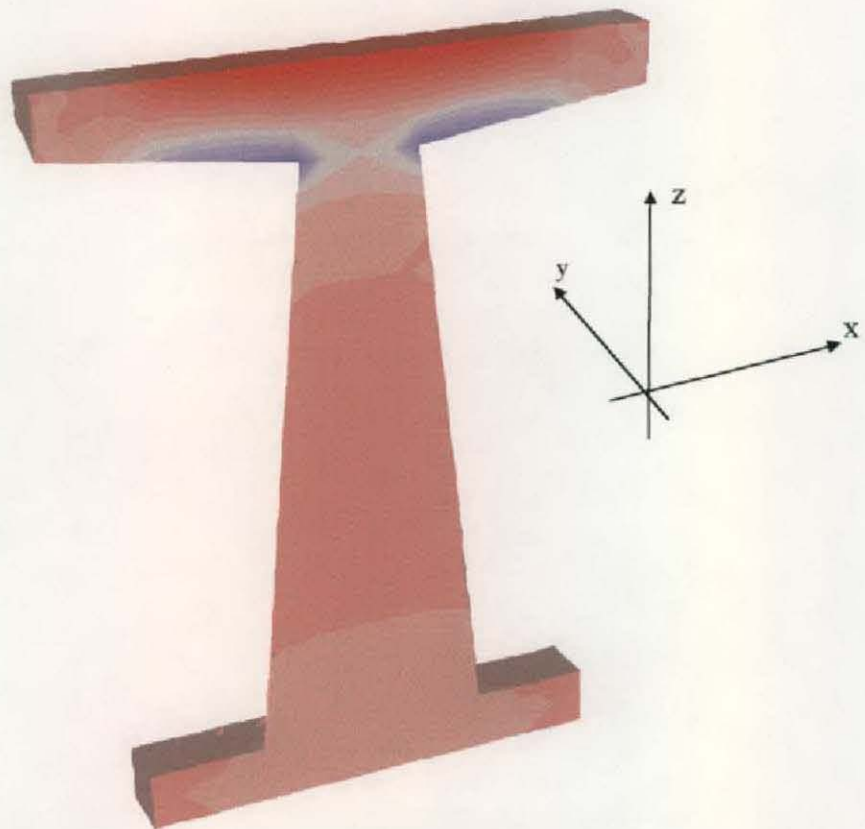


Figura 5.17 - Campos de tensões na meso-estrutura.

6 Conclusão

Este trabalho apresenta algoritmos de geração automática de malhas de elementos finitos bidimensionais e tridimensionais que, juntamente com outros algoritmos desenvolvidos dentro da linha de pesquisa ao qual ele está inserido, são integrados consistentemente em um pacote computacional. Os algoritmos geradores de malhas foram modificados para serem independentes dos pré-processadores, podendo assim serem incorporados em diferentes programas. Um novo algoritmo bidimensional de geração de elementos triangulares é proposto. Este algoritmo incorpora novas idéias, adaptadas de um algoritmo de geração tridimensional de malhas em volume arbitrário, e procura gerar elementos de melhor qualidade geométrica possível. Duas técnicas de geração de malhas sólidas são apresentadas: uma de mapeamento transfinito tridimensional, onde a malha sólida é construída a partir das seções transversais do modelo; e uma técnica de *sweep*, onde a malha sólida é construída pelo “arrasto” de uma seção transversal ao longo de um curva no espaço.

Os algoritmos são incorporados em um modelador geométrico tridimensional, o MG (*Mesh Generator*), e alguns exemplos são feitos para demonstrar a capacidade de modelagem do ambiente criado. Em particular são mostrados exemplos para modelagem composta com diferentes métodos de construção de malhas e diferentes tipos de elementos finitos em um mesmo modelo.

Também é feito um estudo de algumas medidas de distorção para elementos finitos planos. O cálculo dessas medidas de distorção é implementado em um modelador bidimensional gráfico interativo. A visualização das medidas de distorção é feita através de uma escala de cores.

6.1 Principais Contribuições

Os algoritmos de geração de malhas bidimensionais por triangulação desenvolvidos pela linha de pesquisa utilizam-se da técnica de *Enumeração Espacial Recursiva*, através do uso de árvores quartenárias (*quadtree*), para gerar os elementos triangulares. O algoritmo de Vianna (1992) gera pontos no centro de cada célula da *quadtree* e depois faz uma contração do contorno, construindo os elementos. O algoritmo de Cavalcante Neto (1994) constrói os

elementos do interior do domínio utilizando-se de padrões de elementos para cada configuração apresentada pela célula da *quadtree*; para a faixa do contorno gera os elementos por uma técnica de contração de contorno. Os elementos gerados por esses algoritmos demonstram ter uma grande dependência da árvore *quadtree*. No caso do algoritmo proposto no Capítulo 2, os elementos não estão necessariamente atrelados à árvore *quadtree*, pois esta é utilizada apenas para dar uma gradação de transição na geração de “elementos ótimos”. Isto permite uma maior flexibilidade e controle na forma dos elementos. Mesmo que se tenha um elemento de forma não desejável pelo algoritmo é provável que este venha a ser corrigido pelo procedimento de “volta-passos”. O algoritmo de triangulação proposto procura gerar os melhores elementos possíveis no que se refere à sua forma. Logo, uma das primeiras contribuições deste trabalho é o próprio algoritmo em si, como também a idéia de que a *quadtree* não deve ser usada diretamente para gerar os nós. Com a utilização das ferramentas para visualizar a qualidade de uma malha gerada pelos algoritmos de triangulação, o usuário pode decidir qual algoritmo é mais conveniente para suas exigências, ou então fazer as mudanças que julgue necessárias nos parâmetros que definem a geração da malha.

O algoritmo de mapeamento transfinito tridimensional para geração de malhas estruturadas, juntamente com o reconhecimento automático dos pontos das seções transversais usando a topologia de suas malhas, permite a independência requerida pelo algoritmo em relação ao pré-processador. O algoritmo que faz o reconhecimento das malhas das seções transversais independe de como estas malhas são criadas, desde que as seções sejam topologicamente equivalentes. Uma consequência do algoritmo, já que trabalha com a topologia, é a possibilidade de se trabalhar com seções transversais não planas. Entretanto, ainda é necessário desenvolver testes para verificar a validade dos elementos gerados pois as curvas splines podem sofrer interseção e os elementos podem resultar distorcidos.

Os algoritmos que geram malhas tridimensionais estruturadas e não-estruturadas dentro do MG permitem que se tenha uma ferramenta poderosa em auxílio aos problemas de modelagem. A grande contribuição deste trabalho está em disponibilizar essa ferramenta de modelagem tridimensional e uma biblioteca computacional com os algoritmos de geração de malhas bidimensionais e tridimensionais, com uma *API* genérica, possibilitando seu uso em outros modeladores.

6.2 Sugestões para Trabalhos Futuros

Dentre as várias possibilidades de trabalhos futuros com a utilização dos recursos de geração automática de malhas bidimensionais e tridimensionais de elementos finitos, os seguintes trabalhos são sugeridos:

- Inclusão dos procedimentos de “volta-passos” nos algoritmos de triangulação desenvolvidos anteriormente na linha de pesquisa, para que seja possível avaliar o quanto este procedimento é importante na melhoria da malha e depois comparar com os algoritmo proposto no Capítulo 2.
- Aplicação imediata do algoritmo de triangulação em uma estratégia adaptativa onde o tamanho das células da *quadtree* também seriam ditadas pela análise numérica de erros.
- Implementação do algoritmo de triangulação visto no Capítulo 2 em superfícies não planas.
- Implementação de procedimentos de verificação da validade dos elementos gerados pelos algoritmos de mapeamento transfinito tridimensional e *sweep* genérico.
- Estudo das medidas de distorção para elementos tridimensionais e sua implementação em modeladores de elementos finitos tridimensionais. A partir deste estudo uma medida de distorção poderá ser implementada como teste *a posteriori* nos geradores de malhas estruturadas.
- Estudo do efeito da distorção nos resultados obtidos pela formulação isoparamétrica dos elementos finitos em modelos tridimensionais.
- Comparação dos resultados de análise utilizando os diferentes elementos finitos tridimensionais.
- Expansão do modelador tridimensional para análises automáticas de propagação de trincas em problemas de fraturamento tridimensional e refinamento adaptativo tridimensional de forma automática, com intervenção mínima do usuário.

Apêndice A - Sensibilidade da métrica γ/γ^* em Triângulos

No trabalho de Cavalcante Neto (1998) usou-se uma métrica proveniente de um estudo aprofundado de qualidade de forma de elementos finitos tetraédricos. Essa métrica foi utilizada para decidir na reconstrução de malhas com elementos considerados *ruins*. Com o objetivo de aplicar a métrica normalizada γ/γ^* para o caso bidimensional, faz-se aqui um estudo de sensibilidade para esta métrica. Os seguintes testes de sensibilidade foram realizados:

- (a) Para o elemento triangular com coordenadas $(-\sqrt{3}/3,0)$, $(\sqrt{3}/3,0)$ e $(0,1)$, um triângulo equilátero de altura 1, varia-se a posição do nó com coordenada $(0,1)$ de $y = 0,2$ à $y = 3$. Os resultados estão na Figura A.1.
- (b) Para o elemento triangular com coordenadas $(-1,0)$, $(1,0)$, $(0, \sqrt{3}/2)$, varia-se a posição do nó com coordenada $(0, \sqrt{3}/2)$ de $x = 0$ à $x = 3$. Os resultados estão na Figura A.2.

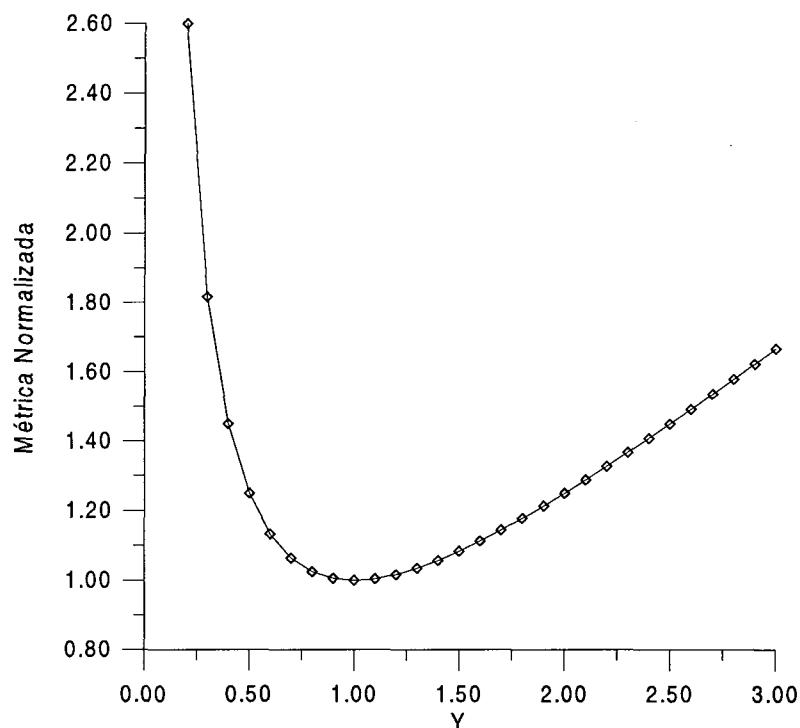


Figura A.1- Sensibilidade - teste (a).

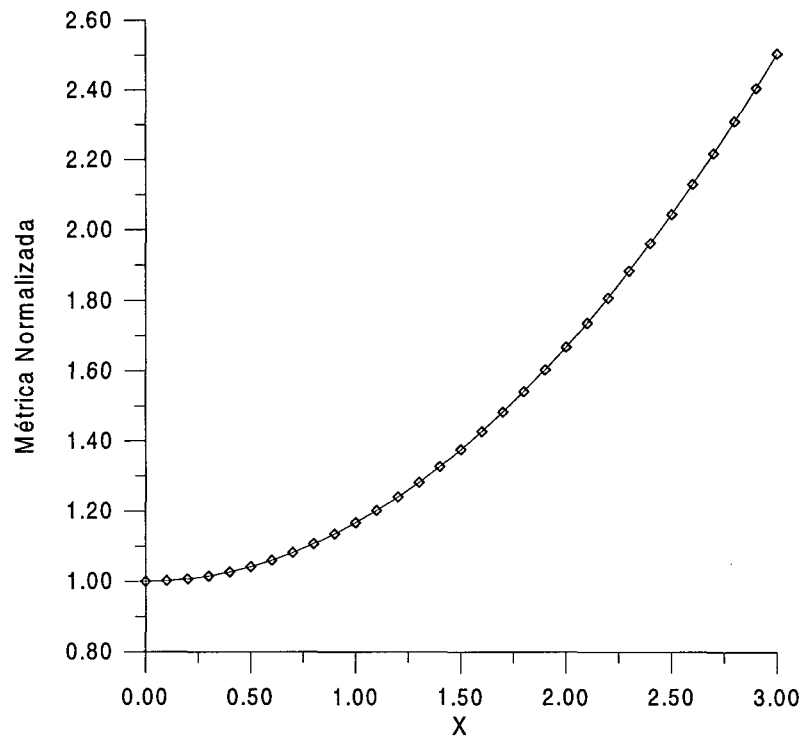


Figura A.2 - Sensibilidade - teste (b)

Os resultados ilustrados nas Figuras A.1 e A.2 demonstram seguir um intervalo de um até o infinito ($[1, \infty)$), tendo assim comportamentos idênticos em relação aos testes de sensibilidade feitos por Parthasarath *et al.* (1993) para os elementos tetraédricos. Logo, essa métrica é aplicável para o caso bidimensional.

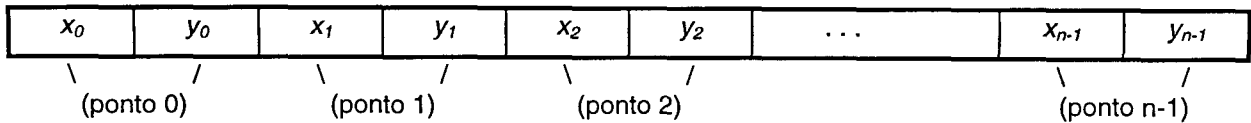
Apêndice B - Biblioteca Computacional para Geração de Malhas de Elementos Finitos

Este apêndice descreve uma biblioteca computacional incluindo alguns algoritmos geradores de malhas desenvolvidos anteriormente na linha de pesquisa. Os objetivos no desenvolvimento desta biblioteca foram:

- disponibilizar os algoritmos para programas que necessitem de malhas de elemento finitos;
- tornar os algoritmos independentes dos programas de pré-processamento;
- permitir a programação dos pré-processadores sem preocupação com os algoritmos específicos;
- padronizar os argumentos de entrada e saída dos algoritmos.

Os argumentos de entrada e saída dos pontos foram padronizados seguindo uma representação por um vetor, como mostra a Figura B.1(a). As coordenadas bidimensionais seguem em duplas (x,y) e os tridimensionais em seguem triplas (x,y,z) . A conectividade dos elementos é representada por um vetor, Figura B.1(b), onde o primeiro argumento representa o numero de nós (nn) do elemento e os argumentos seguintes indicam os índices dos nós, no vetor de pontos, que representa o elemento. E assim seqüencialmente para todos os elementos.

Vetor de coordenadas dos pontos:



Vetor de conectividade dos elementos:

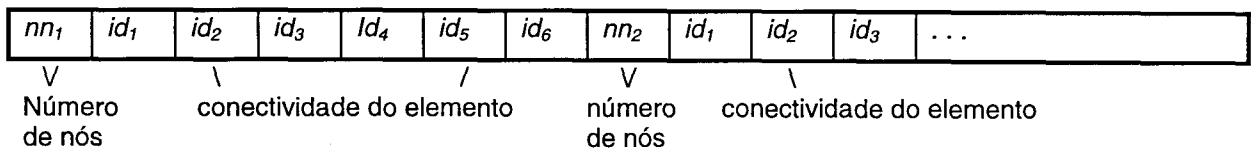


Figura B.1 - Vetores de coordenadas e conectividade.

Com respeito à memória computacional exigida pelos algoritmos, todas as funções da biblioteca fazem os armazenamentos necessários. Existem funções auxiliares para liberar a memória armazenada e que são descritas mais adiante.

A seguir faz-se uma descrição das funções que geram as malhas de elementos finitos. Os argumentos destas funções são apresentadas na forma de um pseudo-código similar à linguagem C.

Geração de Malha Bidimensional

Msh2DBilinear

Argumentos de entrada:

```
double *bry,      /* vetor de coordenadas do contorno          */
int m,           /* número de nós na primeira direção        */
int n,           /* número de nós na outra direção          */
int elem_type,   /* Tipo de elemento:                        */
                /* T3 (3), T6 (6), Q4 (4) ou Q8 (8)        */
int diagtype,    /* Opção de triangulação                    */
                /* = 1 --> diagonal orientada para direita  */
                /* = 2 --> diagonal orientada para esquerda */
                /* = 3 --> modo cruzado alternado          */
                /* = 4 --> melhor diagonal                 */
```

Argumentos de saída:

```
int *nno,        /* número de nós gerados                    */
int *nel,        /* número de elementos gerados              */
double **pt,     /* coordenadas dos nós da malha             */
int **conn       /* conectividade dos elementos               */
```

Descrição:

Mapeamento transfinito bilinear para regiões quadrilaterais com qualquer forma para os lados. O número de segmentos em lados opostos devem ser iguais (Campos, 1991).

Msh2DCollBilinear

Argumentos de entrada:

```
double *bry,      /* vetor de coordenadas do contorno          */
int m,           /* número de nós no lado oposto para o nó colapsado */
int n,           /* número de nós na outra direção          */
int elem_type,   /* Tipo de elemento:                        */
                /* T3 (3), T6 (6), Q4 (4) ou Q8 (8)        */
int diagtype,    /* Opção de triangulação                    */
                /* = 1 --> diagonal orientada para direita  */
                /* = 2 --> diagonal orientada para esquerda */
                /* = 3 --> modo cruzado alternado          */
                /* = 4 --> melhor diagonal                 */
```

Argumentos de saída:

```
int *nno,        /* número de nós gerados                    */
int *nel,        /* número de elementos gerados              */
double **pt,     /* coordenadas dos nós da malha             */
int **conn       /* conectividade dos elementos               */
```

Descrição:

Mapeamento transfinito bilinear para regiões triangulares que conceitualmente é formado por uma região quadrilateral com um dos lados colapsado para um ponto. Os três lados podem ter qualquer forma e os lados adjacentes ao lado colapsado devem ter o mesmo número de segmentos (Campos, 1991).

Msh2DLoft

Argumentos de entrada:

```
double *bry,      /* vetor de coordenadas do contorno          */
int m,           /* número de nós no lado oposto para o nó colapsado */
int n,          /* número de nós na outra direção           */
int dir,        /* direção do lofting:                       */
                /* (0 => primeira direção, 1 => outra direção) */
double weigth    /* peso aplicável a direção do lofting      */
int elem_type,  /* Tipo de elemento:                          */
                /* T3 (3), T6 (6), Q4 (4) ou Q8 (8)         */
int diagtype,   /* Opção de triangulação                     */
                /* = 1 --> diagonal orientada para direita  */
                /* = 2 --> diagonal orientada para esquerda */
                /* = 3 --> modo cruzado alternado          */
                /* = 4 --> melhor diagonal                 */
```

Argumentos de saída:

```
int *nno,        /* número de nós gerados                      */
int *nel,        /* número de elementos gerados                */
double **pt,    /* coordenadas dos nós da malha              */
int **conn       /* conectividade dos elementos                */
```

Descrição:

Mapeamento transfinito linear entre dois lados opostos de uma região quadrilateral. Os outros dois lados devem ser linhas retas. O número de segmentos nos lados opostos devem ser iguais (Campos, 1991).

Msh2DCollLoft

Argumentos de entrada:

```
double *bry,      /* vetor de coordenadas do contorno          */
int m,           /* número de nós no lado oposto para o nó colapsado */
int n,          /* número de nós na outra direção           */
double weigth    /* peso aplicável a direção do lofting      */
int elem_type,  /* Tipo de elemento:                          */
                /* T3 (3), T6 (6), Q4 (4) ou Q8 (8)         */
int diagtype,   /* Opção de triangulação                     */
                /* = 1 --> diagonal orientada para direita  */
                /* = 2 --> diagonal orientada para esquerda */
                /* = 3 --> modo cruzado alternado          */
                /* = 4 --> melhor diagonal                 */
```

Argumentos de saída:

```
int *nno,        /* número de nós gerados                      */
int *nel,        /* número de elementos gerados                */
double **pt,    /* coordenadas dos nós da malha              */
int **conn       /* conectividade dos elementos                */
```

Descrição:

Mapeamento transfinito linear entre um ponto e uma curva, que forma uma região triangular. Os dois lados da região que são adjacentes ao ponto de *lofting* tem de ser linhas retas e ter o mesmo número de pontos (Campos, 1991).

Msh2DTrilinear

Argumentos de entrada:

```
double *bry,      /* vetor de coordenadas do contorno          */
int m,           /* número de nós em um lado do contorno      */
int elem_type,   /* Tipo de elemento:                          */
                /* T3 (3), T6 (6), Q4 (4) ou Q8 (8)          */
int diagtype,    /* Opção de triangulação                      */
                /* = 1 --> diagonal orientada para direita  */
                /* = 2 --> diagonal orientada para esquerda */
                /* = 3 --> modo cruzado alternado           */
                /* = 4 --> melhor diagonal                  */
```

Argumentos de saída:

```
int *nno,        /* número de nós gerados                      */
int *nel,        /* número de elementos gerados                */
double **pt,    /* coordenadas dos nós da malha               */
int **conn       /* conectividade dos elementos                 */
```

Descrição:

Mapeamento transfinito trilinear para uma região triangular com qualquer forma para os lados. O número de segmentos em todos os lados devem ser iguais (Campos, 1991).

Msh2DBoundContraction

Argumentos de entrada:

```
int n_loops,     /* número de circuitos                        */
int *loop_segs, /* vetor com o número de segmentos em cada   */
                /* circuito                                   */
double *bdry_pts, /* vetor de coordenadas do contorno          */
int gen_intpts,  /* indicador para gerar pontos internos      */
                /* ( 0 --> nao gera, 1 --> gera )           */
int n_add_pts,  /* número de nós adicionais ao interior do modelo */
int qt_flag,    /* indicador de como gerar os pontos internos */
                /* ( 0 --> grade, 1 - quadtree )           */
int type_mesh,  /* Tipo de elemento: T3 (3) ou T6 (6)        */
double *add_coords, /* vetor de coordenadas do pontos adicionais */
                /* ao interior do modelo                     */
```

Argumentos de saída:

```
int *n_node,    /* número de nós gerados                      */
double **coords, /* coordenadas dos nós da malha               */
int *n_elem,    /* número de elementos gerados                */
int **conn       /* conectividade dos elementos                 */
```

Descrição: Triangulação genérica aplicável a contornos fechados sem auto-interseção. Os pontos do circuito externo devem estar orientados em sentido horário e circuitos internos (furos) em sentido anti-horário. A triangulação é feita usando uma técnica de contração de contorno (Vianna, 1992). Os pontos interiores da triangulação podem ser gerados pelo algoritmo e/ou fornecidos. Dois procedimentos são utilizados para gerar os pontos interiores. O primeiro procedimento utiliza-se de uma estrutura *quadtree* para gerar pontos no centro de cada célula e depois faz-se uma contração do contorno. O outro procedimento gera os pontos interiores em uma grade regular.

Msh2DQuadTree

Argumentos de entrada:

```
int n_loops,           /* número de circuitos */
int *loop_segs,       /* vetor com o número de segmentos em cada */
                        /* circuito */
double *bdry_pts,     /* vetor de coordenadas do contorno */
int type_mesh,        /* Tipo de elemento: */
                        /* T3 (3), T6 (6), Q4 (4) ou Q8 (8) */
```

Argumentos de saída:

```
int *n_node,          /* número de nós gerados */
double **coords,      /* coordenadas dos nós da malha */
int *n_elem,          /* número de elementos gerados */
int **conn            /* conectividade dos elementos */
```

Descrição: Triangulação genérica aplicável a contornos fechados sem auto-interseção. Os pontos do circuito externo devem estar orientados em sentido horário e circuitos internos (furos) em sentido anti-horário. Para elementos Q4 e Q8, o número de segmentos (lados de elementos) deve ser par em cada circuito. O algoritmo utiliza-se de uma estrutura *quadtree* e constrói os elementos do interior do domínio utilizando padrões de elementos para cada configuração da *quadtree*; para a faixa do contorno gera os elementos por uma técnica de contração de contorno (Cavalcante Neto, 1994).

Msh2DShape

Argumentos de entrada:

```
int n_loops,           /* número de circuitos */
int *loop_segs,       /* vetor com o número de segmentos em cada */
                        /* circuito */
double *bdry_pts,     /* vetor de coordenadas do contorno */
int type_mesh,        /* Tipo de elemento: T3 (3) ou T6 (6) */
```

Argumentos de saída:

```
int *n_node,          /* número de nós gerados */
double **coords,      /* coordenadas dos nós da malha */
int *n_elem,          /* número de elementos gerados */
int **conn            /* conectividade dos elementos */
```

Descrição: Triangulação genérica aplicável a contornos fechados sem auto-interseção. Os pontos do circuito externo devem estar orientados em sentido horário e circuitos internos (furos) em sentido anti-horário. O algoritmo utiliza uma estrutura de *quadtree* para dar uma gradação de transição na geração dos nós interiores. Os pontos do domínio são gerados conjuntamente com o avanço da fronteira e de acordo com o gradiente dado pela árvore *quadtree*. O algoritmo contém adicionalmente um procedimento de melhoria local da malha (veja Capítulo 3).

Geração de Malha Tridimensional

Msh3DExtrusion

Argumentos de entrada:

```
double *d,           /* componentes do vetor direção (u,v,w)           */
double m,           /* comprimento de arrasto                           */
int steps,          /* número de divisões do segmento reto             */
int np_sup,         /* número de pontos na superfície geradora         */
double *pts_sup,    /* vetor de coordenadas dos pontos da             */
                    /* superfície geradora                             */
int ne_sup,         /* número de elementos da super. geradora         */
int *conn_sup,      /* conectividade dos elementos da super. geradora  */
```

Argumentos de saída:

```
int *n_node,        /* número de nós gerados                           */
double **coords,    /* coordenadas dos nós da malha                   */
int *n_elem,        /* número de elementos gerados                    */
int **conn          /* conectividade dos elementos                     */
```

Descrição: *Sweep* ao longo de um segmento reto. A superfície geradora deve ter elementos triangulares ou quadriláteros (veja Capítulo 4).

Msh3DCurveSweep

Argumentos de entrada:

```
int n_pts_curve,    /* número de pontos na curva sweep                */
double **pts_curve, /* matrix (n_pts_curve x 3) com as coordenadas    */
                    /* dos pontos na curva sweep                       */
double **u,         /* matrix com as componentes do vetor u dos       */
                    /* triedros de Frenet-Serret                       */
double **v,         /* matrix com as componentes do vetor v dos       */
                    /* triedros de Frenet-Serret                       */
double **w,         /* matrix com as componentes do vetor w dos       */
                    /* triedros de Frenet-Serret                       */
int np_sup,         /* número de pontos na superfície geradora         */
double *pts_sup,    /* vetor de coordenadas dos pontos da             */
                    /* superfície geradora                             */
int ne_sup,         /* número de elementos da super. geradora         */
```

```

int *conn_sup,          /* conectividade dos elementos da super. geradora */
Argumentos de saída:
int *n_node,           /* número de nós gerados */
double **coords,       /* coordenadas dos nós da malha */
int *n_elem,           /* número de elementos gerados */
int **conn             /* conectividade dos elementos */

```

Descrição: *Sweep* ao longo de uma curva no espaço. Os vetores u , v e w dever estar normalizados e ortogonais entre si. O vetores tangentes a trajetória do *sweep*, w_i , devem seguir a mesmo sentido da trajetória do *sweep* (veja Capítulo 4).

Msh3DMapp

Argumentos de entrada:

```

int ns,                /* número de superfícies (seções transversais) */
int *ids,              /* vetor com os índices dos pontos bases em cada */
                      /* superfície */
int *div,              /* vetor com o número de superfícies entre */
                      /* superfícies principais */
int np_sup,           /* número de pontos em uma superfície */
double **pts_sup,     /* matrix (ns x np_sup*3) com as coordenadas dos */
                      /* pontos das superfícies */
int ne_sup,           /* número de elementos em uma superfície */
int **conn_sup,       /* matrix (ns x ...) com as conectividades dos */
                      /* elementos das superfícies */

```

Argumentos de saída:

```

int *n_node,           /* número de nós gerados */
double **coords,       /* coordenadas dos nós da malha */
int *n_elem,           /* número de elementos gerados */
int **conn             /* conectividade dos elementos */

```

Descrição: Mapeamento transfinito tridimensional a partir de superfícies que representam as seções transversais de um modelo. As seções transversais são interpoladas com uso de uma curva *spline* cúbica. As seções transversais devem ter elementos triangulares e quadrilaterais e serem topologicamente idênticas. Os pontos bases são os pontos conhecidos em cada seção transversal onde uma curva *spline* irá percorrer. O ponto base deve situar-se no contorno da seção transversal (veja Capítulo 4).

Msh3DGeneration

Argumentos de entrada:

```

double factor,         /* fator usado no refinamento da octree */
double tol,            /* tolerância usada no modelo */
int np_sup,           /* número de pontos na superfície geradora */
double *pts_sup,      /* vetor de coordenadas dos pontos da */
                      /* superfície geradora */
int ne_sup,           /* número de elementos da super. geradora */
int *conn_sup,        /* conectividade dos elementos da super. geradora */

```

Argumentos de saída:

```

int *n_node,           /* número de nós gerados           */
double **coords,      /* coordenadas dos nós da malha    */
int *n_elem,          /* número de elementos gerados     */
int **conn             /* conectividade dos elementos      */

```

Descrição: Algoritmo para geração de malhas volumétricas não-estruturadas de tetraedros para domínios arbitrários. A entrada de dados deve ser feita por representação de fronteira, por faces triangulares. O modelo deve estar fechado e as orientações das normais das faces da fronteira devem apontar para dentro do modelo (Cavalcante Neto, 1998).

Funções Auxiliares

Msh2DFreeNodes e Msh3DFreeNodes

Argumentos de entrada:

```
double *points          /* vetor de coordenadas dos pontos */

```

Descrição: Libera a memória do vetor de coordenadas dos pontos passada como argumento de saída pelos algoritmos bidimensionais e tridimensionais, respectivamente.

Msh2DFreeConn e Msh3DFreeConn

Argumentos de entrada:

```
int *conn               /* vetor de conectividade dos elementos */

```

Descrição: Libera a memória do vetor de conectividade dos elementos passada como argumento de saída pelos algoritmos bidimensionais e tridimensionais, respectivamente.

Referências Bibliográficas

- (Baehmann *et al.*, 1987) Baehmann, P. L.; Wittchen, S. L.; Shephard, M. S.; *et al.* - "Robust Geometrically Based, Automatic Two-Dimensional Mesh Generation", *International Journal For Numerical Methods in Engineering*, vol. 24, pp. 1043-1078, 1987.
- (Baker, 1987) Baker, T. J. - "Three Dimensional Mesh Generation b Triangulation of Arbitrary Point Sets", *AIAA-87-1124_CP*, 1987.
- (Bloomenthal, 1990) Bloomenthal, J. - "Calculation of Reference Frames Along a Space Curve", *Graphical Gems*, pp. 567-571, 1990.
- (Burrows, 1986) Burrows, D. J. - "A Finite Element Shape Sensitivity Study", in K. J. Bathe and D. R. J. Owen (eds.), *Reliability of Methods for Engineering Analysis*, Peneridge Press, Swansea, pp. 439-456, 1986.
- (Campos, 1991) Campos, J. A. P. - "Geração de Malhas de Elementos Finitos Bidimensionais Baseada em uma Estrutura de Dados Topológica", Dissertação de Mestrado, Departamento de Engenharia Civil, 1991.
- (Cavalcante Neto, 1994) Cavalcante Neto, J. B. - "Simulação Auto-Adaptativa Baseada em Enumeração Espacial Recursiva de Modelos Bidimensionais de Elementos Finitos", Dissertação de Mestrado, Departamento de Engenharia Civil, PUC/Rio, 1994.
- (Cavalcante Neto, 1998) Cavalcante Neto, J. B. - "Geração de Malha e Estimativa de Erro para Modelos Tridimensionais de Elementos Finitos com Trincas", Dissertação de Doutorado, Departamento de Engenharia Civil, PUC/Rio, 1998.
- (CD, 1996) CD, Manual de Referência, versão 2.0, PUC-Rio, 1996.
- (Celes Filho *et al.*, 1991) Celes Filho, W.; Martha, L. F. e Gattass, M. - "An Efficient Data Structure for Solid Finite Element Analysis

- Post-Processor (in Portuguese)", XI Brazilian Congress of
Mechanic Engineering, pp. 105-108, 1991.
- (Chew, 1989) Chew, L. P. - "Constrained Delaunay Triangulation",
Algorithmica, vol. 4, pp. 97-108, 1989.
- (Dumont e Fernandez, 1997) Dumont, N. A. e Fernandez, J. A. L. - "Um Estudo do
Efeito da Distorção de Forma sobre o Desempenho de
Elementos Finitos - Apresentação de uma Família
Alternativa de Elementos", Anais do XVIII CILAMCE,
pp. 1337-1345, 1997.
- (Farin, 1990) Farin, G. - "Curves and Surfaces for Computer Aided
Geometric Design", *Academic Press*, 2nd. Edition, ISBN:
0-12-249051, pp 464, 1990.
- (Field, 1986) Field, D. A. - "Implementing Watson's Algorithm in
Three Dimensions", *Proc. 2nd ACM Symposium on
Computational Geometry*, pp. 246-259, 1986.
- (Figueiredo e Carvalho, 1991) de Figueiredo, L. H. e. de Carvalho, P. C. P - *Introdução à
Geometria Computacional*, 18^o Colóquio Brasileiro de
Matemática - RJ, 1990.
- (de Floriani e Puppo, 1992) de Floriani, L. e Puppo, E. - "An On-line Algorithm for
Constrained Delaunay Triangulation", *Graphical Models
and Image Processing*, vol. 54, pp. 290-300, 1992.
- (Foley et al., 1989) T.A. Foley and G.M. Nielson - "Knot Selection for
Parametric Spline Interpolation", *Mathematical Methods
in CAGD*, L. Schumaker, Academic Press, pp. 445-467,
1989.
- (Foley et al., 1990) Foley, J. D.; van Dam, A.; Feiner, S. K. e Hughes, J. F. -
Computer Graphics (Principles and Practice), 2nd.
Edition, Wesley Publishing Company, Reading - MA,
1990.
- (Fonseca, 1989) Fonseca, G L. - "Editor Gráfico de Malhas Transfinitas
Tridimensionais para Elementos Finitos", Dissertação de
Mestrado, Departamento de Engenharia Civil, PUC/Rio,
1989.

- (George, 1991) George, P. L. - *Automatic Mesh Generation*, 2nd. Edition, Wiley Publishing, 1991.
- (Gomes Coelho e de Souza, 1995) Gomes Coelho, L. C. E de Souza, C. S. - “Comunicação de Problemas e Soluções Geométricas em uma Interface 3D”, *Anais do VIII SIBGRAFI*, pp 233-240, 1995.
- (Gomes Coelho, 1998) Gomes Coelho, L. C. - “Modelagem de Cascas com Interseções Paramétricas”, Dissertação de Doutorado, Departamento de Informática, PUC/Rio, 1998.
- (Guimarães, 1992) Guimarães, L. G. S. - “Disciplina de Programação Orientada a Objetos para Análise e Visualização Bidimensional de Modelos de Elementos Finitos”, Dissertação de Mestrado, Departamento de Engenharia Civil, PUC/Rio, 1992.
- (Gordon e Hall, 1973) Gordon, W. J. e Hall, C. A. - “Construction of Curvilinear Co-ordinate Systems and Application to Mesh Generation”, *International Journal For Numerical Methods in Engineering*, vol. 7, pp. 461-477, 1973.
- (Haber, 1981) Haber, R.; Shephard, M. S.; Greenberg, D. P.; Abel, J. F. and Gallagher, R. H. - “A General Two-dimensional, Graphical Finite Element Preprocessor”, *International Journal For Numerical Methods in Engineering*, vol.17, pp. 1015-1044, 1981.
- (Hearn e Baker, 1997) Hearn, D. e Baker, M. P. - *Computer Graphics - C version*, 2nd. Edition, Prentice Hall, Inc. , 1997.
- (Hoffmann, 1989) Hoffmann, C. M. - *Geometric and Solid Modeling: An Introduction*, Morgan Kaufmann Publishers, Intercorp., 1989.
- (IUP/Lua, 1995) Grupo de Tecnologia em Computação Gráfica - IUP/Lua Uma interface Lua para o sistema IUP, PUC-Rio, 1995.
- (Joe, 1986) Joe, B. - “Delanay Triangular Meshes in Convex Polygons”, *SIAM J. Sci. Stat. Comput.*, vol. 7, pp. 514-539, 1986.

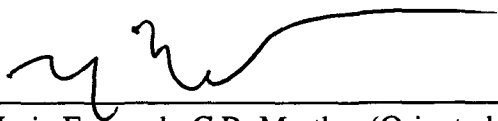
- (Joe e Simpson, 1986) Joe, B. e Simpson, R. B. - "Triangular Meshes for Regions of Complicated Shape", *International Journal For Numerical Methods in Engineering*, vol. 23, pp. 751, 1986.
- (Joe, 1990) Joe, B. - "Construction of Three-Dimensional Triangulations Using Local Transformations", *Comp. Aided Geom. Design*, vol. 8, pp. 123-142, 1990.
- (Lee e Bathe, 1993) Lee, N. S. e Bathe, K. J. - "Effects of Element Distortions on the Performance of Isoparametric Elements", *International Journal For Numerical Methods in Engineering*, vol. 36, pp. 3553-3576, 1993.
- (Lo, 1985) Lo, S. H. - "A New Mesh Generation Scheme for Arbitrar Planar Domains", *International Journal For Numerical Methods in Engineering*, vol. 21, pp. 1403-1426, 1985.
- (Lo, 1989) Lo, S. H. - "Delaunay Triangulation of Non-Convex Planar Domains", *International Journal For Numerical Methods in Engineering*, vol. 28, pp. 2695-2707, 1989.
- (Loher e Parikn, 1988) Loher, R. e Parikn, P. - "Generation of Three-Dimensional Unstructured Grids by the Advancing-Front Method", *International Journal For Numerical Methods in Engineering*, vol. 8, pp. 1135-1149, 1988.
- (Mäntyllä, 1988) Mäntyllä, M. - *An Introduction to Solid Modeling*, Computer Science Press, Rockville, Maryland, 1988.
- (Martha, 1989) Martha, L. F. - "Topological and Geometrical Modeling Approach to Numerical Discretization an Arbitrar Fracture Simulation in Three Dimensions", Ph. D. Thesis, School of Civil Engineering, Cornell University
- (Martha *et al.*, 1996) Martha, L. F.; Menezes, I. F. M.; Lages, E. N.; Parente Jr., E. e Pitangueira, R. L. S. - "An OOP Class Organization for Material Nonlinear Finite Element Analysis", *Anais do XVII CILAMCE*, Congresso Ibero-Latino Americano sobre Métodos Computacionais para Engenharia, Veneza-Itália, pp. 229-232, 1996.

- (MG, 1996) MG, Manual do Usuário, versão 3.0, PUC-Rio, 1996.
- (Moller e Hansbo, 1995) Moller, P. e Hansbo, P. - "On Advancing Front Mesh Generation in Three Dimensional", *International Journal For Numerical Methods in Engineering*, vol. 38, pp. 3551-3569, 1995.
- (Mtool, 1992) Bidimensional Mesh Tool - Manual do Usuário, PUC-Rio, 1992.
- (Mview, 1997) "MVIEW - Bidimensional Mesh View", (Versão 3.0) - PUC-Rio, 1997.
- (Neutral File, 1998) *Neutral File Format*, <http://www.tecgraf.puc-rio.br/~ivan/NeutralFile/nfile.html>, PUC-Rio, 1998.
- (Parthasarath *et al.*, 1993) Parthasarathy, V. N.; Graichen, C. M. e Hathaway, A. F. - "A Comparion of Tethahedron Quality Measures", *Finite Elem. Anal. Des.*, vol 15, pp. 255-261, 1993.
- (Paulino, 1988) Paulino, G. H. - "Pré-Processamento de Estruturas Reticulares Espaciais xxxxxxxx", Dissertação de Mestrado, Departamento de Engenharia Civil, PUC/Rio, 1988.
- (Peruchio *et al.*, 1982) Peruchio, R.; Ingraffea, A. R e Abel, J. F. - "Interat ive Computer Preprocessing for Three-Dimensional Finite Element Analysis", *International Jornal For Numerical Methods in Enginnering*, vol. 18, pp. 909-926, 1982.
- (Peruchio *et al.*, 1989) Peruchio, R.; Saxema, M. e Kela, A.- "Automatic Mesh Generation from Solids Based on Recursive Spatial Decompositions", *International Jornal For Numerical Methods in Enginnering*, vol. 28, pp. 2469-2501, 1989.
- (Potyondy, 1993) Potyondy, D. O. - "A Software Framework for Simulating Curvilinear Crack Growth in Pressurized Thin Shells", Ph.D. Thesis, School of Civil Engineering, Cornell University, 1993.
- (Peraire *et al.*, 1988) Peraire, J.; Peiro, J.; Formaggia, L.; Morgan, K. e Zienkiewicz, O. C. - "Finite Euler Computation in Three-

- Dimensions”, *International Journal For Numerical Methods in Engineering*, vol. 26, pp. 2135-2159, 1988.
- (Robinson, 1987) Robinson, J. - “Some New Distorsion Measures for Quadrilaterals”, *Finite Elements in Analysis and Design*, vol. 3, pp. 183-197, 1987.
- (Robinson, 1988) Robinson, J. - “Distorsion Measures for Quadrilaterals with Curved Boundaries”, *Finite Elements in Analysis and Design*, vol. 4, pp. 115-131, 1988.
- (Roger e Adams, 1990) Roger, D. F. e Adams, J. A. - *Mathematical Elements for Computer Graphics*, McGraw Hill, 1990.
- (Samet, 1984) Samet, H. - “The Quadtree and Related Hierarchical Data Structures”, *ACM Computer Surveys*, vol. 6, no. 2, 1984.
- (Shaw, Pitchen, 1978) Shaw, R. D. e Pitchen, R. G. - “Modifications to the Sahara-Fukuda Method of Network Generation”, *International Journal For Numerical Methods in Engineering*, vol. 12, pp. 93-99, 1978.
- (Shephard e Georges, 1991) Shephard, M. S. e Georges, M. K. - “Automatic Tri-Dimensional Mesh Generation by the Finite Octree Technique”, *International Journal For Numerical Methods in Engineering*, vol. 32, no. 32, pp. 709-749, 1978.
- (Vianna, 1992) Vianna, A. C. - “Modelagem Geométrica Estendida para Modelos Bidimensionais de Elementos Finitos”, Dissertação de Mestrado, Departamento de Engenharia Civil, 1992.
- (Watson, 1981) Watson, D. F. - “Computing the n-Dimensional Delaunay Tessellation with Aplication to Voronoi Polytopes”, *The Computer Journal*, vol. 24, no. 2, pp 167-172, 1981.
- (Wawrzynek, 1991) Wawrzynek, P. A. - “Discrete Modeling of Crack Propagation: Theoretical Aspects and Implementation Issues in Two and Three Dimensions”, Ph.D. Thesis, Cornell University, 1991.
- (Yerry e Shephard, 1984) Yerry, M. A. e Shephard, M. S. - “Automatic Three-Dimensional Mesh Generation by Modified-Octree

Technique”, *International Journal For Numerical Methods
in Engineering*, vol. 20, pp. 1965-1990, 1984.

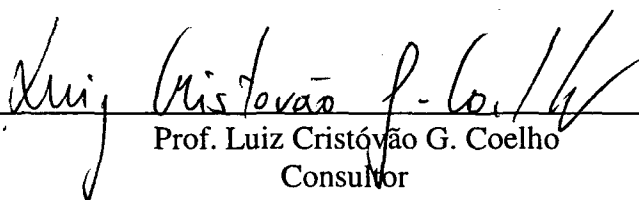
“Integração de Algoritmos de Geração de Malhas de Elementos Finitos”. Dissertação de Mestrado apresentada por ANTONIO CARLOS DE OLIVEIRA MIRANDA em 25 de Fevereiro de 1999 ao Departamento de Engenharia Civil da PUC-Rio e aprovada pela Comissão Julgadora, formada pelos seguintes professores:



Prof. Luiz Fernando C.R. Martha (Orientador)
Departamento de Engenharia Civil / PUC-Rio



Prof. Reynaldo L.N. Taylor de Lima
Departamento de Engenharia Civil / PUC-Rio



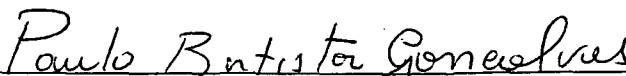
Prof. Luiz Cristóvão G. Coelho
Consultor



Prof. Túlio Nogueira Bittencourt
EPUSP

Visto e permitida a impressão,

Rio de Janeiro, 01/06/99



Prof. Paulo Batista Gonçalves
Coordenador dos Programas de Pós-Graduação
do Centro Técnico Científico