Universidade Federal de Minas Gerais
Departamento de Engenharia Elétrica
Programa de Pós-Graduação em Engenharia Elétrica

Refinamento de Malha Superficial Baseado na Aproximação Suave da Superfície do Modelo

Cássia Regina Santos Nunes

Orientador: *Prof. Dr. Renato Cardoso Mesquita (UFMG)*
Co-Orientador: *Prof. Ph.D. David Alister Lowther (McGill University)*

Tese apresentada ao Curso de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais, como requisito parcial para obtenção do título de Doutor em Engenharia Elétrica.

Belo Horizonte, Dezembro de 2006

Universidade Federal de Minas Gerais
Departamento de Engenharia Elétrica
Programa de Pós-Graduação em Engenharia Elétrica

Remeshing Driven by Smooth Approximation of Model Surface

Cássia Regina Santos Nunes

Advisor: *Prof. Dr. Renato Cardoso Mesquita (UFMG)*
Co-Advisor: *Prof. Ph.D. David Alister Lowther (McGill University)*

Thesis presented to the Graduate Program in Electrical Engineer of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Electrical Engineer.

Belo Horizonte, December 2006

Aos meus pais, Sebastião Daniel e Maria Fátima,
meus primeiros mestres.

To my parents, Sebastião Daniel e Maria Fátima,
my first masters.

Agradecimentos

Ao professor Renato Cardoso Mesquita por ter compartilhado comigo grandes idéias e permitido que eu realizasse este trabalho. Sua paciência, disponibilidade e exigência foram essenciais para o meu crescimento intelectual e pessoal.

Às minhas queridas irmãs, Danielle Fátima e Maria Cláudia, pelo carinho e incentivo.

Aos alunos de iniciação científica, em especial, Felipe Terra e Raphael Chaves, pelo empenho na execução deste trabalho.

Aos professores Jaime Arturo Ramírez, Rodney Rezende Saldanha e Elson José da Silva pelos ensinamentos e auxílios para apresentações de trabalhos em Congressos.

Aos colegas do GOPAC Ricardo, Alexandre, Douglas e Luciano pelas brincadeiras e ajuda na solução de problemas.

Aos colegas de caminhada, Lane, Carla, Marcelo, Anderson, Jose Hissa, Ademir, Julio e Elizabeth pelas trocas de experiências e conversas agradáveis na hora do café.

Ao meu amigo, Frederico Bruno Ribas Soares, pelas palavras reconfortantes e incentivadoras.

Às minhas amigas Larissa, Maíra, Ana Paula e Letícia pelas nossas conversas intermináveis sobre tudo.

Ao Geraldo, meu osteopata, fisioterapeuta, acupunturista e grande amigo.

Aos mestrandos, doutorandos, professores e funcionários do CPDEE/UFMG pela convivência e cooperação.

À CAPES, pela bolsa de estudos durante a realização deste trabalho.

À CAPES/PDEE, pela concessão da bolsa de doutorado sanduíche para a McGill University, que possibilitou a minha viagem e estadia de 1 ano em Montreal/Canadá.

Acknowledgments

I would like to also thank professor David Alister Lowther for receiving me and coaching me during my stay in Montreal/Canadá.

Many thanks to my CADLAB/McGill friends, Dileep Nair and Linda Wang, for sharing amazing moments with me.

Thanks a lot to Kate, Sue, Gustavo, Nadia, Marcia and, specially, Raquel for been my family when I was in Montreal.

Resumo

O método de elementos finitos é uma poderosa ferramenta para a comunidade de engenharia. Uma das barreiras para automatização de análises pelo método de elementos finitos é a geração automática de malhas de boa qualidade. Uma malha com boa distribuição de nós e elementos bem formados contribui na geração de sistemas bem condicionados, minimizando erros numéricos e singularidades. Contudo, a maioria das malhas não é considerada satisfatória sem a aplicação de alguma etapa de pós-processamento para melhorar suas propriedades. As malhas que representam modelos resultantes da aplicação das operações Booleanas e de montagem à modelos pré-existentes possuem um grande número de elementos mal formados. O problema de baixa qualidade também ocorre em malhas geradas por técnicas de reconstrução de superfícies.

Este trabalho apresenta um algoritmo efetivo de refinamento de malhas superficiais para melhorar a forma dos elementos, a distribuição nodal e suas conexões. Este processo consiste na aplicação de séries de operadores de modificação local para: mover, retirar e inserir nós na malha superficial. Para evitar a perda das características geométricas do modelo, uma nova técnica de aproximação da geometria é apresentada. Como, na maioria das vezes, apenas a configuração da malha é conhecida, a aproximação suave do modelo é calculada por partes a partir dos nós que compõem a malha. Os movimentos dos nós são direcionados pela aproximação, o que assegura que eles permaneçam sobre a superfície original durante o aprimoramento da malha. Alguns exemplos são apresentados para ilustrar os resultados alcançados após a aplicação do nosso esquema de refinamento da malha.

Abstract

The finite element method is a powerful tool for the engineering community. One of the barriers for automating finite element analysis is the automatic generation of high-quality meshes. Meshes with good nodes distribution and well-shaped elements provides good system conditioning, which minimizes errors and singularities that might arise. However, most meshes can hardly be called satisfactory without any kind of post-processing to improve their qualities. The meshes representing models generated by the application of the Boolean and assembly operations to predefined primitives have a large number of badly shaped elements. The quality problem also raises in models obtained by the acquisition process, like scanning devices.

This work presents an effective remeshing algorithm to improve the shape, distribution and connectivity of the surface mesh elements, while keeping the mesh geometrically close to the model surface. Our post-processing scheme applies series of local mesh modifications operators to the input mesh. The local operators are able to improve, refine and simplify the mesh. To ensure fidelity, a novel technique was developed to approximate the model surface by smooth surface patches. Since, most of the time, only the mesh configuration is available, a smooth surface approximation is evaluated by pieces from the mesh nodes. The approximation is used to drive the nodes movements and assure that they stay on top of the original model surface during the application of the local mesh modifications. Many examples are presented to illustrate the accomplished results of this work.

Resumo Estendido

O texto a seguir consiste em um resumo estendido sobre o trabalho desenvolvido nesta tese. Primeiramente, este texto introduz o problema abordado, a principal motivação para solucioná-lo e alguns dos principais métodos relacionados. Em seguida, uma breve descrição da principal metodologia desenvolvida é apresentada. Finalmente, as conclusões são apresentadas.

Introdução

O método de elementos finitos (FEM - *Finite Element Method*) é uma boa escolha para solucionar equações diferenciais parciais (PDEs - *Partial Differential Equations*) em domínios complexos. FEM aproxima numericamente a solução de PDEs lineares e não-lineares pela substituição do sistema contínuo de equações por um número finito de equações algébricas lineares e não-lineares acopladas. O domínio do problema deve ser dividido em partes menores de forma simples, construindo uma malha de elementos finitos. Esta malha deve aproximar o domínio de estudo e seus elementos devem satisfazer restrições de tamanho e forma.

Em análise pelo FEM, a qualidade da aproximação da superfície e a qualidade da forma dos elementos da malha volumétrica afetam a precisão dos resultados numéricos. Por exemplo, se os ângulos internos dos elementos são próximos a 180°, o erro de discretização da solução aumenta; se os ângulos são próximos a zero, o número de condição da matriz do elemento aumenta. Resumindo, uma malha com boa distribuição de nós e elementos bem formados contribui na geração de sistemas bem condicionados, o que

minimiza erros numéricos e singularidades que possam ocorrer.

A qualidade da malha volumétrica está intimamente ligada à qualidade da malha superficial. Uma malha superficial de baixa qualidade pode inviabilizar a geração da malha volumétrica correspondente, ou os elementos obtidos são de baixa qualidade. Em geral, os programas geradores de malhas volumétricas têm autonomia para inserir novos pontos na malha superficial, mas não podem decidir sobre a retirada ou modificação da localização de um vértice da malha inicial. A malha superficial inicial deve estar presente na malha volumétrica resultante. Assim, malhas superficiais de boa qualidade facilitam a obtenção de malhas volumétricas também de boa qualidade.

A princípio, modeladores de sólidos podem produzir malhas superficiais de alta qualidade, mas após a aplicação de algumas operações Booleanas e de operações montagem a modelos pré-existentes, a qualidade da malha resultante diminui drasticamente. O problema de baixa qualidade também ocorre em malhas geradas por técnicas de reconstrução de superfícies. Estas técnicas reconstroem superfícies a partir de um conjunto de pontos, os quais podem ser obtidos de vários tipos de fontes, como digitalizadores tridimensionais a laser. As técnicas de reconstrução enfatizam a aproximação da geometria e topologia, mas não garantem a qualidade da malha superficial gerada.

Neste contexto, refinar a malha superficial torna-se muito importante para maximizar a qualidade da forma de seus elementos, reduzindo o número de ângulos agudos, melhorando a distribuição nodal e suas conexões.

O objetivo deste projeto é a geração de malhas superficiais de alta qualidade para modelos obtidos através de processos de aquisição de dados ou modelos resultantes da aplicação das operações Booleanas e/ou de montagem sobre primitivas pré-definidas. Uma malha de alta qualidade viabiliza um sistema de elementos finitos bem condicionado, o que aumenta a precisão dos resultados obtidos.

Para melhorar a qualidade da malha superficial, diferentes métodos de pós-processamento podem ser usados. Existem três técnicas básicas para melhoramento da malha: suavização (*smoothing*), simplificação (*clean-up*) e subdivisão (*refining*). Suavização inclui qualquer método que melhora o posicionamento dos nós sem alterar suas conexões. Simplificação, geralmente, refere-se a processos que modificam a conectividade dos elementos, através da retirada de elementos da malha. Enquanto a subdivisão reduz o tamanho dos elementos,

inserindo novos elementos. Na maioria das vezes, a aplicação de apenas uma dessas técnicas não é suficiente para alcançar o nível de qualidade desejado. Então, para garantir a obtenção de uma malha superficial com alto grau de qualidade, duas ou três técnicas são aplicadas em conjunto, o que gera métodos híbridos de pós-processamento.

Em outra classificação, os métodos de refinamento de malhas superficiais são divididos em três grupos. O primeiro grupo é baseado no particionamento das malhas 3D em *patches*(partes) e no tratamento de cada parte em separado. Esta técnica produz resultados razoáveis, mas é muito sensível a estrutura dos *patches*, além da amostragem dos vértices ser de difícil controle. Outro grupo de algoritmos está baseado na parametrização global da malha original. Uma nova triangulação de boa qualidade é gerada sobre o domínio paramétrico e depois projetada de volta ao espaço 3D. Assim, a malha resultante é uma versão melhorada do modelo original. A principal desvantagem dos métodos de parametrização global é a sensibilidade dos resultados à parametrização utilizada. Converter uma estrutura 3D não-trivial num plano paramétrico distorce drasticamente a estrutura e informações importantes, não especificadas claramente, podem ser perdidas. Mesmo que a parametrização minimize as distorções do modelo original, é impossível eliminá-las completamente. Além disso, os métodos de cálculo da parametrização global são lentos, pois eles envolvem a solução de grandes sistemas de equações, muitas vezes não-lineares.

A principal alternativa à parametrização global é trabalhar diretamente sobre a malha superficial e realizar séries de modificações locais para melhorar, enriquecer ou simplificar a malha. Está técnica é conhecida como processo de adaptação da malha ou simplesmente *remeshing*(refinamento da malha) e este método é utilizado neste trabalho. Algoritmos de *remeshing* podem ser considerados métodos híbridos, pois os operadores de modificação local da malha encapsulam as técnicas básicas de suavização, simplificação e subdivisão.

Os operadores de modificações locais permitem mover, retirar e inserir nós na malha superficial. Para evitar a perda das características geométricas do modelo, torna-se essencial o uso de uma aproximação da geometria do modelo. Essa aproximação direciona os movimentos dos nós e assegura que eles permaneçam sobre a superfície original do modelo.

Infelizmente, na maioria das vezes, apenas a configuração da malha é conhecida e não as características geométricas do modelo. Para superar este problema, uma representação suave da superfície do modelo é utilizada para aproximar a sua geometria. A aproxima-

ção geométrica é calculada por partes a partir dos vértices que compõem a malha. Os operadores de modificações locais são aplicados à malha, levando em consideração as informações geométricas provenientes da aproximação da superfície do modelo. A construção da representação suave da superfície do modelo constitui na principal contribuição deste trabalho.

A próxima seção apresenta a técnica introduzida para o cálculo da aproximação suave da superfície do modelo. Em seguida, o método para refinamento de malhas superficiais utilizando esta técnica é discutido.

Aproximação Suave da Superfície do Modelo

A aproximação da superfície do modelo é muito importante durante o processo de refinamento da malha superficial. É através dela que as características geométricas do modelo serão conhecidas e preservadas. Um operador de modificação local da malha é aplicado apenas quando a qualidade da forma do elemento e o erro de aproximação permanecem dentro dos limites pré-estabelecidos.

A aproximação utilizada deve garantir a representação de modelos gerados através da aplicação das operações Booleanas e operação de montagem sobre primitivas pré-definidas, assim como de modelos gerados a partir de técnicas de reconstrução de superfícies. Por definição, os modelos reconstruídos são obtidos a partir de um conjunto desorganizado de pontos P, usualmente denso, amostrados diretamente de uma superfície suave S. Por outro lado, as malha resultantes das aplicações das operações Booleanas e de montagem podem ser formadas por superfícies planares e curvas. Seu conjunto de vértices é normalmente reduzido. Para representar os dois tipos de modelos, a técnica de aproximação da superfície do modelo por um conjunto de *patches*(partes) suaves é introduzida. A cada face da malha é associado um *patch* suave, que pode ser plano ou curvo. Aproximar a geometria do modelo por um conjunto de *patches* diminui o tempo de processamento e minimiza erros de aproximação, pois a parametrização global é evitada.

Para calcular a aproximação de uma face da malha, os vértices da mesma e os vértices vizinhos são utilizados. O uso dos vértices vizinhos é importante para melhorar a qualidade da aproximação, pois leva em consideração o comportamento da curvatura lo-

cal do modelo. Se o conjunto de vértices é coplanar, a aproximação da face é um *patch* plano; senão, a face é aproximada por um *patch* curvo. Outro aspecto considerado na identificação do tipo de *patch* é o erro associado à aproximação calculada. Após o cálculo do *patch* curvo, o erro médio entre o *patch* e os vértices utilizado no cálculo é avaliado. Se o erro for maior que um valor limite especificado (por exemplo: o erro máximo pode ser limitado em 2%), a aproximação é descartada e a face é então aproximada por um plano. A aproximação de toda a superfície do modelo é formada por um conjunto de *patches* suaves curvos ou planos. Os *patches* curvos são *patches* B-splines calculados através da técnica de mínimos quadrados, onde os vértices da face a ser aproximada e seus vértices vizinhos são as entradas.

As superfícies paramétricas B-splines são frequentemente usadas em sistemas CAD (*Computer Aided Design* - Projeto Assistido por Computador) para aproximar pontos distribuídos irregularmente no espaço. Entre as vantagens na utilização das B-splines, destacam-se: i) garantia de boa aproximação de uma grande variedade de sólidos; ii) preservação de altos graus de continuidade em superfícies complexas; e iii) controle local, o que significa que uma modificação local da forma não é propagada para toda a superfície.

O problema consiste em encontrar um *patch* de superfície B-spline $s(u, v)$, que aproxima um conjunto de vértices, com representação da forma:

$$s(u, v) = \sum_{\omega=0}^{\nu} \mathbf{M}_{\omega}(u, v) c_{\omega}, \tag{1}$$

onde, c_{ω} são pontos de controle e $\mathbf{M}_{\omega}(u, v)$ são funções base.

Seja p_{τ}, $\tau = 1, ..., \mu$, o conjunto de vértices da face e seus vérticies vizinhos e s_{τ} suas aproximações. A aproximação é calculada como um mínimo da função F:

$$F = \sum_{\tau=0}^{\mu} \|s(u_{\tau}, v_{\tau}) - p_{\tau}\|^2 = \sum_{\tau=0}^{\mu} \left[\sum_{\omega=0}^{\nu} \mathbf{M}_{\omega}(u_{\tau}, v_{\tau}) c_{\omega} - p_{\tau} \right]^2. \tag{2}$$

Ou seja, assumindo que as funções base, $\mathbf{M}_{\omega}$, são dadas ou pré-calculadas, precisa-se encontrar o conjunto de pontos de controle, c_{ω}, que minimize F.

Este problema clássico de aproximação por mínimos quadrados sempre possui solução. Contudo, essa solução não é única e também, a superfície resultante, $s(u, v)$, pode não

ser suficientemente suave. Por isso, acrescenta-se à Equação 2 um termo quadrático de regularização (F_s), denominado termo de suavização ou termo de penalidade. Assim, garante-se que a solução seja única e com suavização controlada. Frequentemente, esse termo é obtido a partir da aproximação de *thin-plate energy*, uma função quadrática nas derivadas parciais de segunda ordem:

$$F_s = \int\int (s_{uu}^2 + 2s_{uv}^2 + s_{vv}^2) dudv. \tag{3}$$

ou a partir de *membrane energy* que é quadrática nas derivadas parciais de primeira ordem:

$$F_s = \int\int (s_u^2 + s_v^2) dudv. \tag{4}$$

Acrescentando o termo de suavização a Equação 2, a função a ser minimizada torna-se:

$$F = \sum_{\tau=0}^{\mu} \left[\sum_{\omega=0}^{\nu} \mathbf{M}_\omega(u_\tau, v_\tau) c_\omega - p_\tau \right]^2 + \lambda F_s, \tag{5}$$

onde λ é um valor real maior ou igual a zero, denominado parâmetro de suavização. Como F e qualquer um dos termos de suavização (F_s) são quadráticos em c_ω, a Equação 5 pode ser reescrita na forma matricial:

$$\mathbf{F} = \|\mathbf{Bc} - \mathbf{p}\|_2^2 + \lambda \mathbf{c}^T \mathbf{E} \mathbf{c}, \tag{6}$$

onde, $\mathbf{c} = (c_1, ..., c_\nu)^T$, $\mathbf{p} = (p_1, ..., p_\mu)^T$, $\mathbf{E}$ é uma matriz $\nu \times \nu$, simétrica e definida positiva, e a matriz $\mathbf{B}$ é $\mu \times \nu$, da forma:

$$\mathbf{B} = \begin{pmatrix} \mathbf{M}_1(u_1, v_1) & \dots & \mathbf{M}_\nu(u_1, v_1) \\ \mathbf{M}_1(u_2, v_2) & \dots & \mathbf{M}_\nu(u_2, v_2) \\ \vdots & \dots & \vdots \\ \mathbf{M}_1(u_\mu, v_\mu) & \dots & \mathbf{M}_\nu(u_\mu, v_\mu) \end{pmatrix} \tag{7}$$

Tomando-se o gradiente de F e igualando-o a zero, tem-se:

$$\left(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{E}\right)\mathbf{c} = \mathbf{B}^T\mathbf{p}, \tag{8}$$

A matriz $B^T B$ é de ordem $n \times n$, simétrica e semi-definida positiva, então a solução da Equação 8 com $\lambda = 0$ não é necessariamente única. Como $\mathbf{E}$ é positiva definida, a solução do sistema matricial $(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{E})$ para $\lambda > 0$ é também definida positiva e desse modo não singular, o que implica que a Equação 8 tem solução única.

Uma boa aproximação dos nós da malha é muito importante para preservação das características geométricas do modelo durante o processo de melhoramento da malha superficial. A técnica de aproximação suave da superfície do modelo apresentada neste trabalho garante isto. Ela reduz erros de aproximação e pode ser utilizada para aproximar uma grande variedade de sólidos.

A seguir o método de refinamento de malha superficial introduzido neste trabalho e algumas características de implementação são discutidos.

Remeshing

O método de refinamento proposto deve garantir a melhoria de malhas de modelos gerados pela aplicação das operações Booleanas e de montagem a modelos pré-existentes, bem como de modelos obtidos a partir de um conjunto de pontos por métodos de reconstrução. Os elementos da malha resultante devem possuir boa qualidade; ângulos internos próximos a 60° (elementos triangulares) ou superior (demais elementos); boa distribuição dos vértices e boa aproximação da superfície do modelo. A forma e distribuição dos elementos são requisitos dos geradores de malha volumétrica. A qualidade da aproximação geométrica é necessária para que os resultados das simulações eletromagnéticas sejam válidos e compatíveis com a realidade.

Estudos recentes indicam o processo de adaptação da malha superficial como excelente alternativa para o refinamento de malhas superficiais de baixa qualidade. Ele evita as desvantagens dos métodos que utilizam parametrizações globais, pois trabalha diretamente sobre a malha que representa o modelo.

O método apresentado neste trabalho é baseado nos operadores de modificações locais, que encapsulam os métodos básicos de suavização, simplificação e subdivisão da malha. Para evitar a perda de informações geométricas do modelo, calcula-se uma aproximação suave da superfície do modelo através da técnica apresentada na seção anterior.

Os operadores de modificação local são aplicados sempre que a qualidade dos elementos da malha é aprimorada. Eles permitem simplificar, enriquecer e movimentar elementos da malha. Para garantir que as características geométricas do modelo não sejam perdidas, a execução dos operadores está condicionada ao desvio entre os elementos da malha e a aproximação suave. Se este desvio ultrapassa os limites pré-estabelecidos, a operação não pode ser executada.

Os operadores de modificações locais utilizados neste trabalho são: união de aresta (*edge-collapsing*), divisão de aresta (*edge-splitting*), realocação de vértice (*vertex relocation*) e troca de arestas (*edge-swapping*). Estes operadores são aplicados sequencialmente para alcançar as características de malha desejadas. Os operadores de união de aresta e divisão de aresta são usados para melhorar a forma dos elementos e também a taxa de amostragem, a qual varia de acordo com a curvatura local. Regiões de maior curvatura conterão elementos menores e maior número de vértices, enquanto regiões planas conterão vértices mais esparsos e elementos maiores. A aproximação do modelo fornece informações sobre a curvatura local. As operações de troca de elementos e realocação de vértices melhoram apenas a qualidade da forma dos elementos. A qualidade da aproximação geométrica é medida *a priori*. Operações que diminuem o ângulo mínimo dos elementos envolvidos ou aumentam o desvio entre o elemento e a aproximação geométrica não são aplicadas. A aplicação de uma modificação local modifica a configuração dos elementos.

Os operadores de modificação local são detalhados a seguir. Seja uma aresta AB, a operação de união de aresta consiste em unir os vértices A e B num único vértice. Esta operação é executada se a exclusão da aresta AB não faz com que as arestas conectadas ao vértice B extrapolem o desvio permitido.

A operação de divisão de aresta introduz um novo vértice na aresta AB. O novo vértice é inserido quando se aumenta o valor do ângulo mínimo das faces que compartilham a aresta longa, ou quando a distância da aresta à superfície supera o desvio máximo permitido. O novo vértice é calculado sobre a superfície aproximada. Caso as faces que compartilham essa aresta sejam triangulares, os triângulos existentes são divididos produzindo quatro novos triângulos.

O procedimento de realocação de vértices consiste em redefinir todas as faces que compartilham o vértice A. Todas essas faces são mapeadas num mesmo plano e o melhor posicionamento de A para cada face é calculado. A seguir, a média dos pontos ótimos é

obtida e o resultado é mapeado sobre a superfície aproximada, obtendo-se A'.

Por último, tem-se o operador de troca de arestas. Este operador é aplicado quando os ângulos internos dos triângulos que compartilham a aresta testada crescem. Os triângulos que compartilham a aresta devem ser coplanares ou praticamente coplanares. Esta operação introduz modificações na curvatura local.

O processo de refinamento da malha superficial pode ser resumido pelo algoritmo:

1: **Definições preliminares**:
2: Obtém-se o conjunto de *patches* associados às faces a serem refinadas
3: Calcula o desvio entre a malha e o modelo aproximado
4: Preenche o vetor E de arestas a serem testadas
5: **Refinamento da malha**:
6: **enquanto** a malha sofre alterações
7: **para** cada aresta $e \in E$
8: **se** a união de vértices aumenta ângulos internos
9: **e** não perde características geométricas
10: une e
11: **senão se** a troca de arestas aumenta ângulos internos
12: **e** não perde características geométricas
13: troca e
14: **senão se** a divisão de arestas melhora a geometria **ou** a qualidade da malha
15: divide e
16: **fim se**
17: realoca os vértices da aresta e
18: **fim para**
19: **fim enquanto**

Tabela 1: Algoritmo para Refinamento da Malha

Características de Implementação

O método foi implementado no GSM (Gopac Solid Modeler). O GSM é um Modelador de Sólidos Voltado para Aplicações em Eletromagnetismo, que se encontra em desenvolvimento pelo GOPAC (Grupo de Otimização e Projeto Assistido por Computador) da UFMG (Universidade Federal de Minas Gerais). Este modelador garante a interpretação única das interfaces entre componentes; assegura a compatibilidade da malha de elementos

finitos superficial; facilita e agiliza os processos de criação, edição, visualização e acesso à representação computacional de sólidos. O modelador tem como objetivo gerar uma malha de elementos finitos de boa qualidade, capaz de fornecer as informações necessárias para simulação eletromagnética do modelo. O GSM está dividido em quatro subsistemas, sendo que cada um pode ser desenvolvido e operado individualmente, mesmo que estejam interligados. Estes subsistemas são:

- subsistema de Interface, responsável direto pela interação do usuário com o GSM e pela visualização do modelo construído. As funções que o GSM realiza são captadas e descritas por procedimentos da interface, que ativam procedimentos dos outros subsistemas. Os resultados das funções são recebidos e apresentados pela Interface;

- subsistema de Modelagem, responsável pelas operações realizadas na representação interna. Possui funções que avaliam a descrição de um novo modelo ou de uma alteração e, a seguir, acionam as funções do subsistema de Representação para criar ou alterar uma representação. Também é responsável pelas questões topológicas e geométricas solicitadas pelo sistema. As operações Booleanas e operação de montagem estão contidas neste subsistema;

- subsistema de Representação, ou Núcleo: responsável pela manipulação, gerenciamento e acesso à representação interna. As funções para armazenamento da descrição dos objetos e composições em bases de dados permanentes são também incluídas nesse subsistema. Os dados das representações internas podem ser lidos por funções de outros subsistemas, mas as criações e modificações são realizadas pelo subsistema de representação;

- subsistema de Geração de Malha, garante a geração da malha de elementos finitos a partir da descrição geométrica da fronteira, obtida da representação B-rep.

A biblioteca SINTEF LSMG (versão 1.0) foi utilizada para calcular e tratar os *patches* de superfície B-splines.

Todo projeto foi realizado utilizando técnicas de orientação a objeto. A superfície suave aproximada foi encapsulada numa classe que é responsável por fornecer a interface necessária aos operadores de modificação local. Assim, facilita-se a troca do tipo superfície ou o método de aproximação utilizado para construí-la.

Cada face da malha está associada a uma aproximação da superfície do modelo. A aproximação pode ser um *patch* B-spline ou um plano. Admite-se um erro de 2% para o *patch* B-spline em relação aos pontos utilizados para seu cálculo. As aproximações geradas com erro maior que 2% são descartadas e as faces são aproximadas por planos. Regiões com curvatura acentuada precisam de malhas mais densas para que a aproximação obtida seja válida.

É importante relembrar que todas as novas faces geradas pelos operadores de modificação local são testadas antes da aplicação dos mesmos. Os testes estão relacionados à qualidade da forma e à qualidade da aproximação geométrica. Operações que degradariam a qualidade da forma dos elementos ou acrescentariam à malha um desvio maior que o permitido não são realizadas.

Para os modelos resultantes das operações Booleanas e de montagem, apenas as faces da região de interface participam do processo de refinamento. Num modelo gerado a partir de uma nuvem de pontos todas as faces são utilizadas pelo método. Os *patches* das faces das regiões de interface devem ser calculados antes da aplicação das operações Booleanas e de montagem, pois os vértices das faces que serão destruídas auxiliam no cálculo da aproximação suave da superfície do modelo.

Conclusões

A geração de malhas volumétricas de alta qualidade é essencial para garantir soluções eletromagnéticas precisas e de alta qualidade. A qualidade da aproximação do modelo ao sólido real estudado torna as simulações compatíveis com a realidade.

A qualidade da malha volumétrica está intimamente ligada à qualidade da malha superficial. Uma malha superficial de baixa qualidade pode inviabilizar a geração da malha volumétrica correspondente, ou os elementos obtidos são de baixa qualidade. Neste contexto, garantir a obtenção de malhas superficiais de boa qualidade facilita a geração de malhas volumétricas também de boa qualidade.

Este trabalho apresenta um método de refinamento baseado na aplicação de séries de operadores de modificação local a malha superficial. Essas operações são direcionadas por uma aproximação suave da superfície do modelo. Essa aproximação suave é calculada

considerando as coordenadas dos vértices da malha e suas conexões. A aproximação garante as informações geométricas necessárias à manutenção das características originais do modelo. O método discutido é eficiente na melhoria da qualidade da malha superficial de modelos obtidos através da aplicação de operações Booleanas e de montagem a modelos pré-existentes ou de modelos obtidos através de processos de aquisição de dados.

Contents

1 Introduction **1**

1.1 Motivation . . . 2

1.2 The Problem Origin . . . 3

1.3 Approach . . . 4

1.4 Contributions . . . 5

1.5 Outline of this work . . . 6

2 General Concepts **7**

2.1 Finite Element Method . . . 7

2.2 Geometric Modeling . . . 9

2.2.1 Constructive Solid Modeler . . . 11

2.2.2 Sweep Representation . . . 12

2.2.3 Boundary Representation . . . 12

2.2.3.1 Euler Operators . . . 14

2.2.4 Boolean and Assembly Operations . . . 15

2.2.4.1 Primitive Mesh Generation . . . 17

2.2.4.2 Intersecting Process . . . 17

2.2.4.3 Elements Classification . . . 18

2.2.4.4 Boolean Evaluation and Elimination of Undesired Elements 20

2.3 Finite Element Mesh . . . 21

2.3.1 Delaunay Triangulation . . . 24

2.3.2 Delaunay Tetrahedralization . . . 25

2.3.3 Surface Mesh Generation . . . 27

2.3.3.1 Sphere Discretization . . . 28
2.3.3.2 Swept Primitives Discretization . . . 29
2.3.3.3 Surface Reconstruction . . . 29
2.4 Mesh Quality Measures . . . 31
2.4.1 Triangular Elements . . . 36
2.4.1.1 The Minimum/Maximum Angle . . . 36
2.4.1.2 The Aspect Ratio . . . 36
2.4.1.3 The Distortion Metrics . . . 36
2.4.2 Tetrahedral Elements . . . 36
2.4.2.1 Radius-edge ratio . . . 37
2.4.2.2 Aspect ratio . . . 38
2.4.2.3 Dihedral Angle . . . 38
2.5 Surfaces . . . 38
2.5.1 Surface Representation . . . 39
2.5.2 Bezier Surface . . . 42
2.5.3 B-Spline Surface . . . 44
2.6 Smooth Approximation of a Points Cloud . . . 46
2.7 Discussion . . . 49

3 Surface Mesh Post-Processing Methods 51
3.1 Smoothing Methods . . . 52
3.1.1 Averaging Methods . . . 53
3.1.2 Optimization-Based Methods . . . 53
3.1.3 Physically-Based Methods . . . 54
3.2 Clean-up Methods . . . 54
3.2.1 Shape Improvement . . . 54
3.2.2 Topological Improvement . . . 55
3.3 Refinement Methods . . . 55
3.3.1 Edge Bisection . . . 55
3.3.2 Point Insertion . . . 56
3.4 Local Mesh Modification Operators . . . 56
3.5 Hybrid Methods . . . 58
3.5.1 Frey's Method . . . 59

3.5.1.1 Geometric Surface Mesh 60
3.5.1.2 The Geometric Support 61
3.5.1.3 Unit Surface Mesh Construction 62
3.5.2 Surazhsky and Gotsman's Method 63
3.5.2.1 Geometric Background 64
3.5.2.2 Remeshing 65
3.6 Discussion 67

4 Remeshing Driven by Smooth Approximation of Model Surface 69
4.1 Avoiding the Planar Facets Triangulation 71
4.2 Remeshing 73
4.2.1 Approximation of the Model Surface 75
4.2.2 Algorithm 77
4.3 Implementation Characteristics 80
4.4 Discussion 83

5 Results 85
5.1 Models Generated by Boolean and Assembly Operations 85
5.1.1 Results when the planar facet triangulation is avoided 85
5.1.2 Results Applying the New Remeshing Scheme 89
5.1.3 Discussion 98
5.2 Models Generated by Acquisition Process 100
5.3 Discussion 113

6 Conclusions 115
6.1 Summary of the Accomplished Work 120
6.2 Future Work 123

Bibliography 124

Index 129

Chapter

1

Introduction

Numerical methods based on spatial discretizations, as the finite element method, the boundary element method, the finite volume method, and others, play an important role in computer aided design (CAD), engineering (CAE), and manufacturing (CAM). Namely, the finite element method (FEM) is probably the most widespread analysis technique in the engineering community. This technique is capable of solving field problems governed by partial differential equations for very complex geometries. However, successful and efficient use of FEM still requires significant expertise, time, and cost.

Many researchers are investigating ways to automate FEM, thus allowing improved productivity, more accurate solutions, and use by less trained personnel. Often the most time consuming and experience requiring task faced by an analyst is the discretization of the general geometric definition of a problem into a valid and well conditioned finite element mesh. For complex geometries, the time spent on geometry description and mesh generation are the pacing items in the computational simulation cycle. A particularly complex example given by Mavriplis [Mav00] showed the mesh preparation time to be 45 times that required to compute the solution.

Automatic generation of consistent, reproducible, high quality meshes without user intervention makes the power of the finite element analysis accessible to those not expert in the mesh generation area. Therefore, tools for an automated and efficient mesh generation

are important prerequisites for the complete integration of the FEM with design processes in CAD, CAE, and CAM systems.

Most of the research on development of fully automatic unstructured mesh generators has been concentrated on various triangulation schemes. The advantage of them lies in the fact that simplicial elements (triangles and tetrahedra) are most suitable to discretize domains of arbitrary complexity, particularly when locally graded meshes are needed. Over the past decades, a wide class of algorithms for the generation of triangular and tetrahedral meshes has been established from three basic strategies: tree based approach, advancing front technique, and Delaunay triangulation. While the meshing schemes for the discretization of 2D problems matured into very robust and efficient algorithms, there are still many open issues in 3D, including not only theoretical guarantee of convergence, quality bounds but also implementation aspects as robustness and versatility.

The long-term goal for developers of meshing tools is the generation of high quality meshes directly from CAD models, without user interaction.

1.1 Motivation

The finite element method is a powerful tool for the engineering community. One of the barriers to automating finite element analysis is the automatic generation of high quality meshes.

Triangle meshes are flexible and commonly used as boundary representation for surfaces with complex geometric shapes. In addition to their geometric qualities or topological simplicity, intrinsic qualities such as shape of triangles, their distribution on the surface and the connectivity are essential for many algorithms working on them.

The quality of the volumetric mesh generated from the surface triangulation is directly affected by the quality of the surface mesh. If the surface mesh quality is poor, the volumetric mesh either cannot be generated or a poor quality volumetric mesh results.

In finite element analysis, the quality of the model surface approximation and the mesh quality are very important. For example, if the elements angles become too large, the discretization error in the finite element solution is increased and, if the angles become too small, the condition number of the element matrix is increased. Summarizing, highly

regular meshes are necessary for engineers to perform effective finite element analysis. The accuracy and cost of the analysis directly depends on the size, shape, and number of elements in the mesh.

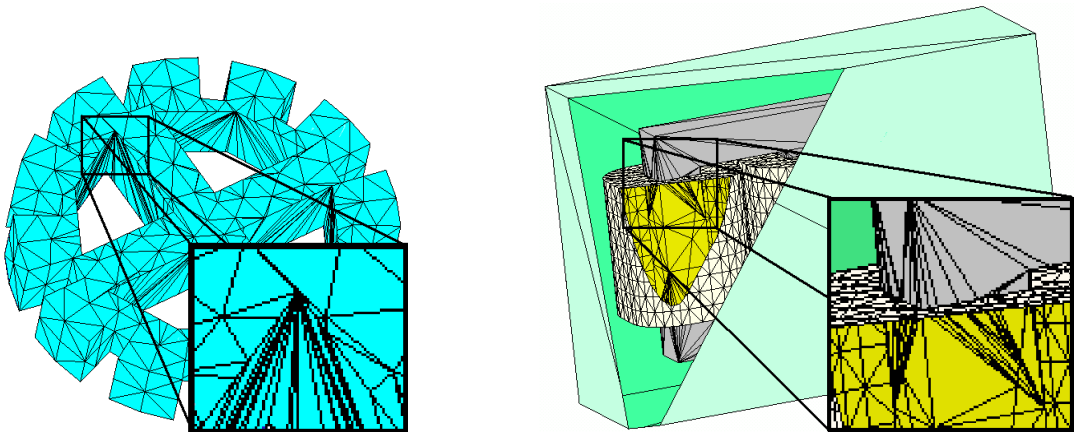
Unfortunately, most of surface meshes can hardly be called satisfactory without any kind of post-processing to improve their intrinsic qualities.

In this context, improving the surface mesh quality is an almost obligatory step for the mesh data in finite element analysis. This step must reduce or eliminate sharp angles, improve the nodes distribution and their interconnections, while preserving the geometric characteristics as much as possible.

1.2 The Problem Origin

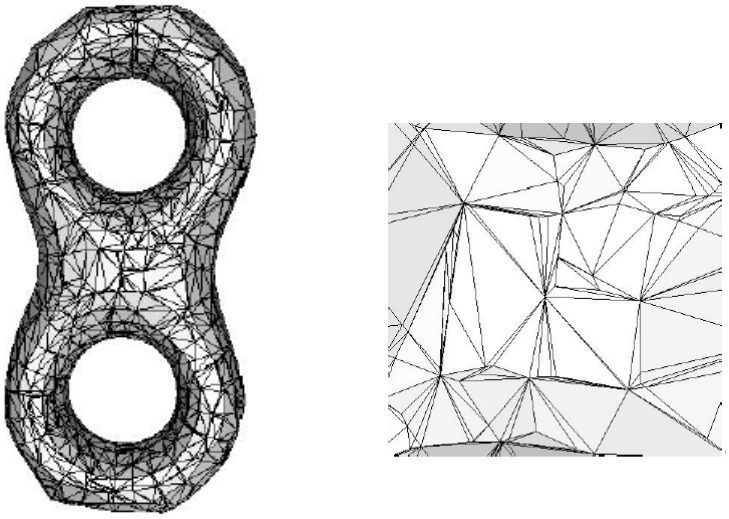
Usually, automatic mesh generators can produce surface meshes with a specified quality degree for simple predefined primitives, like spheres, cylinders or prisms. However, to generate models with high complexity two options frequently can be chosen: the application of the Boolean (union, intersection or subtraction) and assembly operations over predefined primitives; or the reconstruction of surfaces by an acquisition process, such as medical imagery, laser range scanners, contact probe digitizers, radar and seismic surveys. Unfortunately, both methods produce surface meshes with a large number of badly shaped elements. For the Boolean and assembly operations, the elements in the intersection areas are usually split into degenerate ones, decreasing drastically the elements quality, as Figure 1.1 illustrates. Each triangle from one object can be intersected by more than one triangle in the other object, and this generates small and badly shaped triangles which compose the resultant mesh. The surface meshes with this level of quality cannot be used as input for finite element volumetric mesh generators.

Badly shaped triangles also raises in meshes generated by an acquisition process. During the reconstruction process, the algorithm guarantees good approximation for the geometry and topology, but it does not guarantee the triangle shape quality. In many cases, the generated mesh possess a large amount of triangles and many of them are badly shaped, which sometimes makes impracticable the volumetric mesh generation. Figure 1.2 shows an example of reconstructed mesh.



(a) Gear Model (b) Three-phase transformer model

Figure 1.1: Examples of surface meshes generated by the application of Boolean and assembly operations over predefined primitives



(a) (b)

Figure 1.2: a) Reconstructed surface from a points cloud, b) zoom of a small part of the surface, [Goi04]

As consequence, the surface meshes of models generated by: the application of the Boolean and assembly applications over simple models; or from an acquisition process, should be improved, before being used as input to electromagnetic simulation by finite element method.

1.3 Approach

The final goal of our project is the generation of high quality surface meshes for models obtained by acquisition process or models resultants from the application of the Boolean and assembly operations over predefined primitives. A high quality mesh will result in a

well conditioned finite element system that minimizes numerical errors and singularities that might otherwise arise.

To improve the surface mesh quality, different post-processing methods can be used, such as surface smoothing, cleaning-up, refining, adaptive methods, among others. This work presents some advantages of using the mesh adaptation process to improve the mesh quality. It works directly on the surface mesh and applies series of local modifications operators on the mesh. These operators can enrich, simplify or improve locally the mesh. To avoid losing model geometric characteristics, it is necessary to know the model surface geometry.

Unfortunately, most of the time, only the mesh configuration is available. To overcome this, it is important to have an approximation of the model surface geometry. We introduce a smooth surface approximation evaluated by pieces from the mesh nodes to approximate the model geometry. Each mesh face is approximated by a B-spline surface patch using least squares formulation. The approximation is used to drive the nodes movements and assure that they stay on top of the original model surface during the application of the local mesh modifications.

The optimization procedure will then consist of analyzing the current mesh edges in order to collapse the short edges, to split the large ones, to improve mesh connections and distribution. Edge swapping and vertices relocations improve the elements shape quality. The deviation between the mesh elements and the geometric approximation of the model is controlled during the whole process.

1.4 Contributions

The main contribution of our work is the development of a remeshing method suitable to improve indistinctly the surface mesh of models generated by application of Boolean and assembly operation over 3D primitives and models reconstructed from a set of points. It performs series of local mesh modifications driven by a smooth approximation of the model surface.

We also introduce a new approach to evaluate the model surface approximation considering the mesh nodes. This method is able to give good approximation for models

generated by application of Boolean and assembly operation over 3D primitives and models reconstructed from a set of points.

1.5 Outline of this work

This document is organized in 6 chapters (including this one). Chapter 2 surveys the general concepts of finite element method, geometric modeling, mesh generation and mesh quality metrics. It also brings a brief study on surfaces and a technique to evaluate a smooth surface approximation of a scattered set of points based on the least square formulation.

In chapter 3, we discuss the state-of-the-art in surface mesh post-processing methods. The basic methods like smoothing, cleaning-up and refinement are presented first. Following two remeshing techniques are detailed.

Chapter 4 presents our method to improve surface mesh quality, the remeshing driven by smooth approximation of model surface. It preserves the geometric characteristics of models that can be obtained after the application of the Boolean and assembly operations to predefined primitives or reconstructed from a set of unorganized points.

In chapter 5, some results are shown to illustrate the accomplished work. The initial mesh elements quality and the initial model geometry will be compared to the respective results after the application of our remeshing approach.

Finally, chapter 6 presents our conclusions and perspectives of future work.

Chapter

2

General Concepts

This project goal is to produce high quality meshes, to guarantee the evaluation of accurate results in electromagnetic simulation throughout the finite element method.

In order to have a better understanding of the area where this work is inserted, this chapter surveys the general concepts of finite element method, geometric modeling, mesh generation, mesh quality metrics, surfaces and smooth approximation. The problems that may arise during the mesh generation, surface reconstruction or after the Boolean and assembly operations application are highlighted.

2.1 Finite Element Method

"The limitations of the human mind are such that it cannot grasp the behavior of its complex surroundings and creations in one operation. Thus the process of subdividing all systems into their individual components or 'elements', whose behaviors is readily understood, and then rebuilding the original system for such components to study its behavior is a natural way which the engineer, the scientist, or even economist proceeds" - O. C. Zienkiewicz and R. L. Taylor (1989).

"Everybody nowadays knows what finite elements are: they are the methods for solving

boundary value problems in which one divides the domains of the problem into little pieces over which the solution is approximated using polynomials. The little pieces are finite elements, and the polynomials are called shape functions" - J. T. Oden (1990).

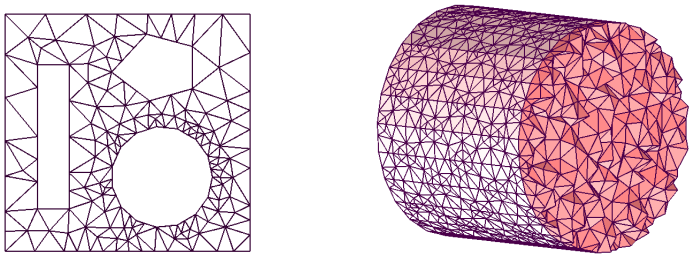
Many physical phenomena in science and engineering can be modeled by partial differential equations (PDEs). When these equations have complicated boundary conditions, are posed on irregularly shaped objects, or the domains include non-linear materials, the PDEs usually do not admit closed-form solutions. A numerical approximation of the solution is thus necessary.

The *finite element method* (FEM) is one of the numerical methods for solving PDEs. FEM evolved in the early 1950's with the aim of solving mechanical engineering problems, as heat diffusion, fluid flow, and stress analysis. In 1970, PP. Silvester and M. V. K Chari proposed the use of this method for electromagnetic problems, including in its formulation the solution of non-linear problems. The methods used before 1970 were not completely satisfactory, in particular when the structure had a complex geometry or when nonlinearity in ferromagnetic materials was considered [Ida97].

FEM numerically approximate the solution of a linear or nonlinear PDE by replacing the continuous system with finite number of coupled linear or nonlinear algebraic equations. This process of *discretization* associates a variable to each node in the problem domain.

It is not enough to choose a set of points to act as nodes; the problem domain must be partitioned into small pieces of simple shape. In the FEM, these pieces are called *elements* and they are usually triangles or quadrilaterals (in two dimensions) or tetrahedra or hexahedral bricks (in three dimensions). The FEM employs a node at every element vertex; each node is typically shared among several elements. The collection of nodes and elements is called a *finite element mesh*. Two and three-dimensional finite element meshes are illustrated in Figure 2.1.

Since the elements have simple shapes, it is easy to approximate the behavior of a PDE on each element. The change of the dependent variable with regard to position is approximated within each element by an interpolation function. The interpolation function is defined relative to the values of the variable at the nodes associated with each element. The original boundary value problem is then replaced with an equivalent integral



(a) Two dimensional mesh

(b) Three dimensional mesh

Figure 2.1: Finite element meshes

formulation. The interpolation functions are then substituted into the integral equation, integrated, and combined with the results from all other elements in the solution domain. The results of this procedure can be reformulated into a matrix equation of the form $A\Phi = b$, which is subsequently solved for the unknown variable.

The accuracy of the numerical solution depends on the size and also on the shape of finite elements. The smaller are the elements, the closer the numerical solution will be to the exact solution. But, as the number of elements increase, the processing time also increases.

Considering the elements shape, various metrics of quality for first order triangular and tetrahedral mesh elements have appeared in the mathematical and technical literature. Some of these metrics are discussed in section 2.4.

Before the discretization of the problem domain into the finite element mesh, the domain need to be modeled. Next section presents techniques to accomplish this.

2.2 Geometric Modeling

The term geometric modeling first came into use in the early 1970s with the rapidly developing computer graphics, computer-aided design (CAD) and manufacturing (CAM) technologies. It refers to a collection of methods used to define the shape and other geometric characteristics of an object [Mor85].

When a model of something is constructed, it means to create a substitute of that thing - a representation. If the model is a good one, it will respond to questions in the same way that the original would. Commonly, a model has only the essential information

for an objective, while ignoring other information. For many applications, the geometric model of a physical object may require the complete description of surface reflectance properties, texture and color; or it may include only information on elastic properties of the object material. The required detail can be determined by the uses and operations to which the model is intended to. If the model is rich enough in descriptive detail, one can perform operations on it that yield the same results as operations performed on the subject itself.

Geometric modeling methods are used to construct precise mathematical description of the shape of a real object or to simulate some process. The geometric modeling is divided in many subareas, among them three are the most common:

- **Wireframe modeling**: It describes the object in terms of its vertices and edges, the faces and other characteristics are referenced. It is simple to construct, but the representation is usually ambiguous with lack of visual coherence and information to determine the object interpretation;

- **Surface modeling**: It provides mathematical description of the shape of the objects surfaces, it also offers few integrity checking features. But surfaces are not necessarily properly connected and there is no explicit connectivity information stored. This subarea is often used where only the visual display is required, e.g. flight simulators.

- **Solid Modeling**: It contains information about the closure and connectivity of the volumes of solid shapes, it also guarantees consistent description of closed and bounded objects and provides a complete description of an object modeled as a rigid body in 3D space. Solid models enable a modeling system to distinguish between the inside and outside of a volume;

Solid modelers permit rapid construction of finite element models. They associate the geometric entities (abstract solids) to symbolic representations, throughout representation schemes. The major representation schemes are the constructive solid geometry, the sweep representation and the boundary representation. Following, these schemes are described and their description is based on [Req80, Mag00a, Nun02] works. The schemes can be used by themselves or combined forming hybrid representation schemes. The Boolean and assembly operations are also discussed in this section. They are an important class

of solid manipulation and, unfortunately, most of the time, the meshes quality degrades after their application.

2.2.1 Constructive Solid Modeler

In constructive solid geometry (CSG) a solid is defined as a combination of simple regular solids. This simple solids are called primitives, typically spheres, blocks, cylinders, pyramids, torus, etc. The primitives are combined among themselves by a set of Boolean and assembly operations or modified by geometric transformations to produce more complex solids. The Boolean operations are the intersection, union and difference. These operations are discussed in section 2.2.4.

Modeling a solid with this scheme is very intuitive for most users. The CSG models are usually represented by CSG trees. Figure 2.2 shows the construction steps of a CSG model.

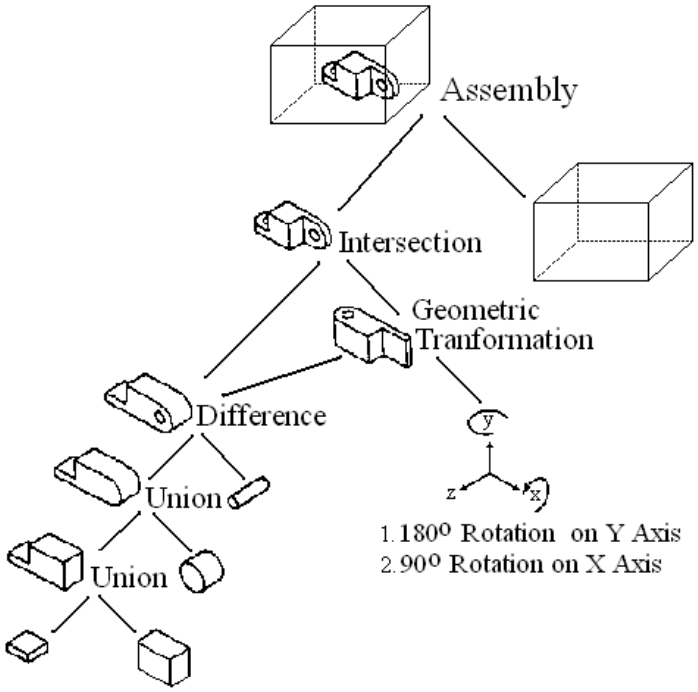
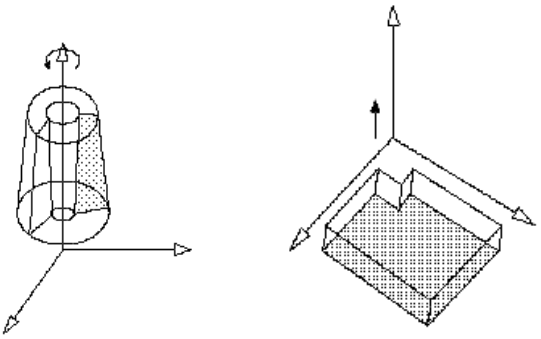


Figure 2.2: Steps of a CSG model construction [Hui92]

2.2.2 Sweep Representation

The basic notion embodied in sweeping schemes is very simple: A solid is represented by a *moving object* plus the *trajectory*. The most common are the rotational and translational sweep of 2D primitives (circles, squares, rectangles, etc.), Figure 2.3 presents both of them.



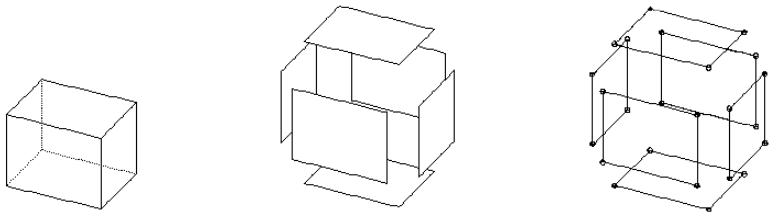
(a) Rotational (b) Translational

Figure 2.3: Sweep Solids [Mag00a]

Usually, this scheme is used to represent the primitives of the CSG models.

2.2.3 Boundary Representation

A boundary representation (B-rep) represents a solid by segmenting its boundary into a finite number of bounded subsets. Each B-rep element of the model is formed by a set of other elements, following a hierarchy: the model is formed by a set of regions, that are formed by a set of shells, that are formed by a set of faces that posses edges delimited by vertices, which represents the minimum structure, as shown in Figure 2.4.



(a) Region (b) Faces (c) Edges and Vertices

Figure 2.4: B-rep elements [Mag00a]

The B-rep data structure discussed here was proposed by Magalhães in [Mag00a,

Mag00b]. It implements the *use* concept, which enables the multiple existence of the basic topological elements (vertex, edge or face) to share the same geometry and form. For example, any internal face of the model make reference to exactly two *FaceUses* and all model boundary faces make reference to only one *FaceUse* (the one that takes part in the model). The *use* concept can also be applied to vertices and edges and these elements can have two or more *uses*. This concept is illustrated in Figure 2.5.

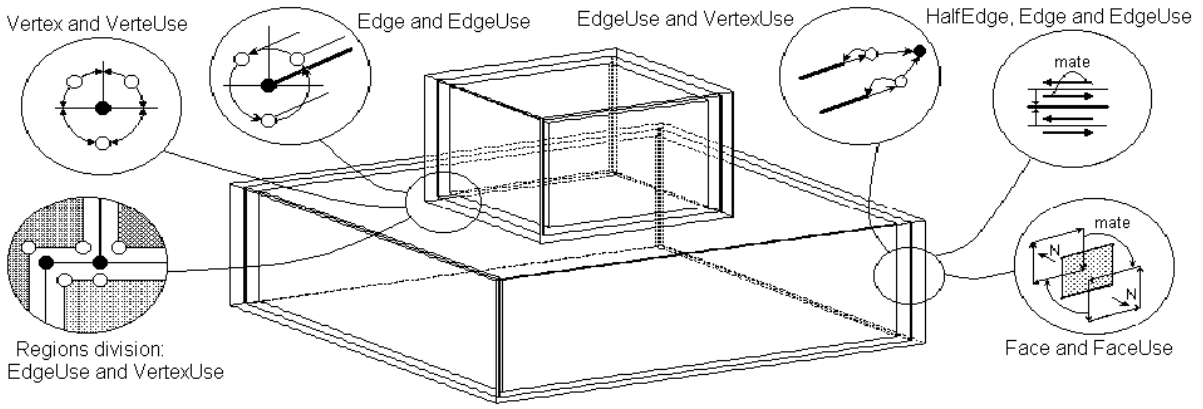


Figure 2.5: Main features of B-rep structure [Mag00a, Nun02]

Each bounded region has its own orientation (direction), compatible with the orientation of the other components. Each *FaceUses* is oriented by the outer *LoopUse*. The *Holes* on the face are defined by counter-oriented loops. Each *LoopUse* is composed of a sequence of *HalfEdges* related to *EdgeUses*, which are finally bounded by two *VertexUses*. The *HalfEdges* are responsible for the model orientation and they are always treated in pairs (called *mates*) belonging to the same region. Figure 2.6 shows how the B-rep elements interact. The structure is represented in UML[i] notation.

The B-rep data structure (Figure 2.6), involves the primitive elements *Model, Region, Shell, Face, Edge, HalfEdge* and *Vertex*. *Face*, *Edge* and *Vertex* have derivative elements representing their occurrence in each region (*FaceUse*, *EdgeUse* and *VertexUse*) and their geometry (*FaceGeom*, *EdgeGeom* and *VertexGeom*). *Faces* do not have any direction, but *FaceUses* do. So, the loop concept is related to the *FaceUse* direction, and is, then, called *LoopUse*. Auxiliary structures link all the uses of one edge to the uses of edges in one shell. The same situation occurs between the uses of one vertex and the uses of vertexes in one shell. All the topological structure are based on double linked lists. In these lists, only the head node makes reference to the element that is hierarchically one level higher.

[i]The Unified Modeling Language (UML) is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system.

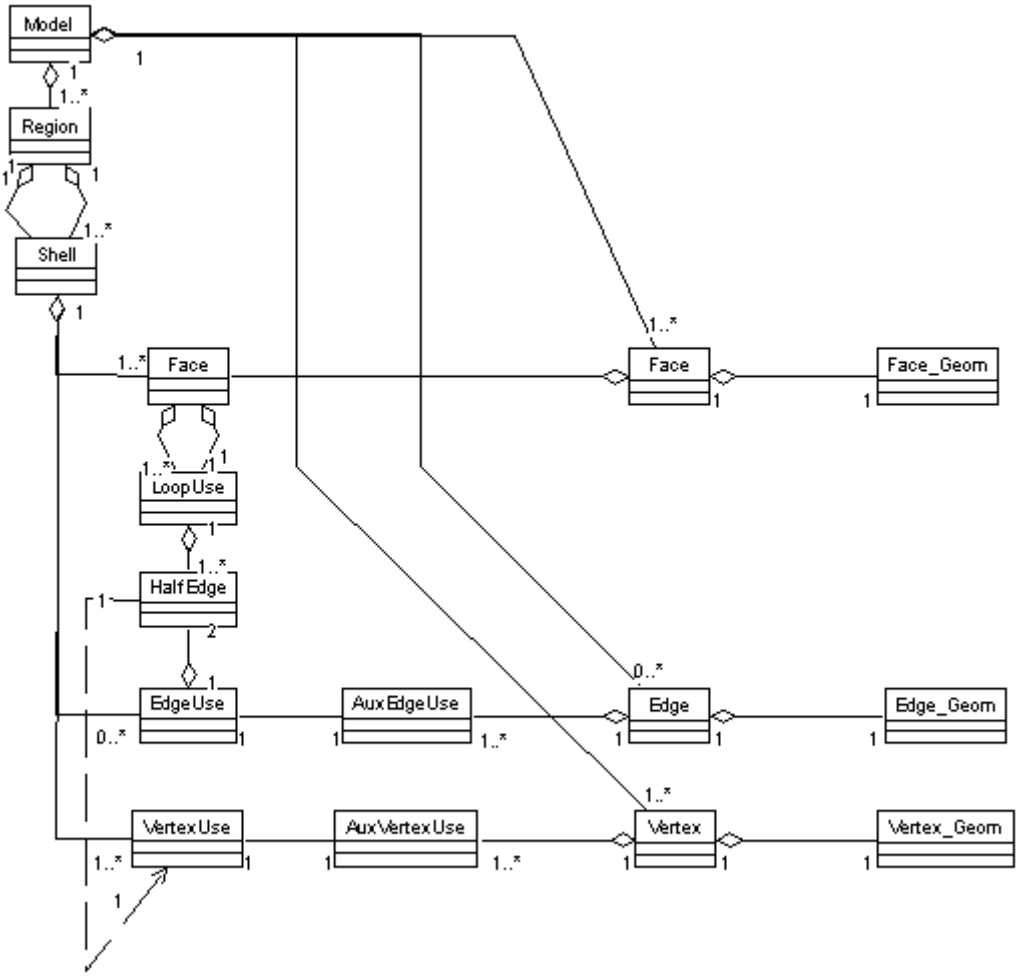


Figure 2.6: B-rep data structure [Nun02]

Usually, the B-rep models are built by a sequence of Euler operators to guarantee valid solids constructions. Next section presents a brief explanation about these operators.

2.2.3.1 Euler Operators

A B-rep solid is considered topologically correct if its elements are well connected (ex: faces are bounded by edges which, then, connect two vertexes). The assurance of topological validity is obtained by Euler operators [Mag00a, Mag00b, Nun02]. To guarantee the solid topological validity, a variation of Euler formula was defined:

$$V - E + F - L = 2 * (R + S - H) \tag{2.1}$$

being V, E and F, respectively, the number of vertex, edges and faces; L is the number of inner loops in faces; R is the number of regions (outer shells); S is the number of inner

shells (holes in regions) and it is the component genus (which corresponds to H, of *Hole*).

The Euler operators defined to the presented B-rep data structure are presented on Table 2.1. Even though the nomenclature and description use the words Face, Loop, Edge and Vertex, the operators will be applied to the use of these elements. The word *use* was omitted to facilitate the association between the elements and letters used in the name of the operators. For example, the operator MVFR introduces a new region to the B-rep model. The new region starts with one vertex and one face. The operator MEV introduces a new vertex and a new edge to the data structure. The new edge is connected from an existent vertex to the new one.

Table 2.1: Basic Euler operators defined to the B-rep data structure [Mag00b]

Operator	Description
MVFR	Make Vertex, Face, Region
KVFR	Kill Vertex, Face, Region
MEV	Make Edge, Vertex
KEV	Kill Edge, Vertex
MEF	Make Edge, Face
KEF	Kill Edge, Face
MEKL	Make Edge, Kill Loop
KEML	Kill Edge, Make Loop
MFKLH	Make Face, Kill Loop Hole
KFMLH	Kill Face, Make Loop Hole
MSKR	Make Shell, Kill Region
KSMR	Kill Shell, Make Region
MFFR	Make double Face, Region
KFFR	Kill double Face, Region

2.2.4 Boolean and Assembly Operations

The Boolean operations allow users to perform operations like union, subtraction and intersection over simple models to generate new models with higher complexity.

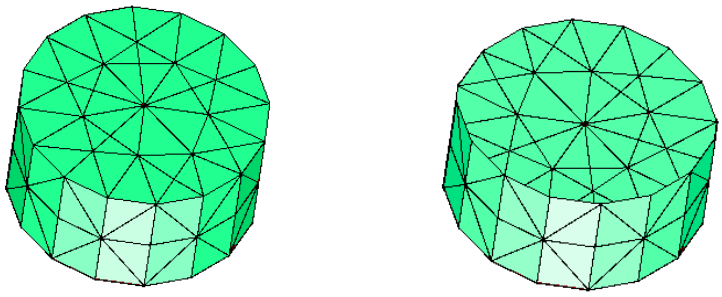
Assemblies are very important in electromagnetic problems where objects are composed of parts with different materials in direct contact. When the FEM is used to solve these problems, it is necessary to interpret the common boundary between the parts as a single piece, even when the parts are used separately to perform other operations. Doing that, the compatibility of the generated finite element mesh can be guaranteed. Consider,

for instance, the air that involves an electrical device. This air region can be easily shaped using the assembly operation performed in a box assembled to an electrical device located in its interior. A model for an assembly should reflect its important geometric properties: the shape, the arrangement and the individual characteristic of its components [Arb90].

The algorithm to generate the polygonal representation is called *boundary evaluator*. The process of boundary evaluation presented here is based on works [Nun03, Mag00a]. This process follows the *divide and conquer* algorithm presented in [Req85, Til80]. It has four steps:

1. mesh generation over primitives;
2. intersecting process;
3. elements classification;
4. Boolean evaluation and elimination of all undesired elements.

Each step returns a consistent and compatible polygonal mesh to be used in the next step, until the result is achieved. Compatible polygonal mesh means that there is coincidence among vertex and edges endpoints. Figure 2.7 shows examples of a compatible mesh and a non compatible one.



(a) Compatible mesh example

(b) Non-compatible mesh example

Figure 2.7: Examples of compatible and non-compatible meshes

The three first steps of the boundary evaluation are common to all operations. Only the Boolean evaluation and the elimination of undesired elements are specifically done, according to the executed operation.

Before starting this work, a triangular representation of a model was a prerequisite to evaluate the Boolean and assembly operations. As it will be discussed, working with

them is easier than working with polygonal meshes. The following subsections explains the boundary evaluation steps for general polygonal meshes and how it affects the resultant mesh quality.

2.2.4.1 Primitive Mesh Generation

The first step of boundary evaluation is the generation of surface mesh over primitives. Many solids in engineering problems have curved surfaces. These solids can not be represented by a collection of polygons without precision losses: the polygonal representation is only an approximation of the real solid. However, the polygonal mesh is very flexible and can be automatically generated.

The 3D primitive can be chosen from a set of predefined ones (ellipsoids, torus, cylinders or prisms), swept primitives, which are generated by applying sweep operations (translational or rotational) to 2D primitives (arcs or polylines) or it can be a reconstructed surface. Methods to generate surface meshes for these 3D primitives are detailed in section 2.3.3.

2.2.4.2 Intersecting Process

Finding the intersection points is the first step in the intersecting process. In order to accomplish this, the convex hull of each object is tested. If intersections exist, all components of one object need to be compared in relation to the other object components, to identify the intersection points. After evaluating all the intersection points and edges, they are separated into two groups: i) points located on facet's boundary and ii) points and edges located inside the facets. The first group is directly inserted into the data structure. Then each facet and the second group are triangulated using a constrained 2D Delaunay mesh generator [She96]. The data structure modifications are realized by the application of the Euler Operators 2.2.3.1.

The evaluation of the intersection points between triangular surface meshes is easier than in polygonal surface meshes. There are a limited number of possible combinations between triangles, as Figure 2.8 illustrates.

After finishing the intersecting process, the mesh over the primitives will be compati-

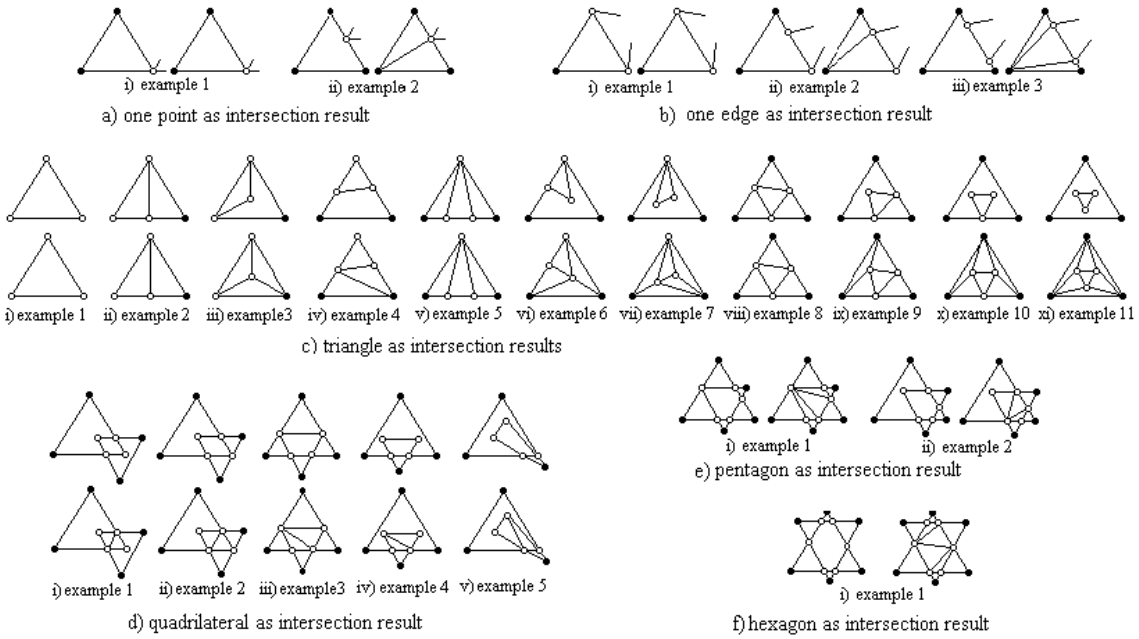
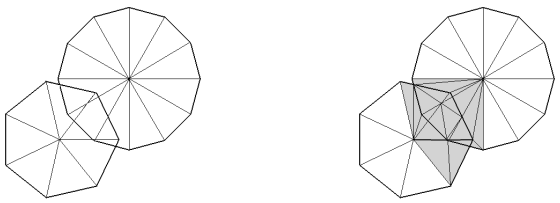


Figure 2.8: Possible combinations between triangles [Mag00a]

ble, correct and formed by triangular faces. It will be ready for the elements classification.

The elements quality deterioration that may appear in the resultant mesh occurs in this step. Each element from one object is commonly intersected by more than one element from other object, as in Figure 2.9(a). The small and badly shaped triangles appear when the triangulation is regenerated, as shown at Figure 2.9(b). The new triangles form the resulting mesh.



(a) Before (b) After

Figure 2.9: The shaded part represent the triangles generated after the intersections are evaluated

2.2.4.3 Elements Classification

When all intersections have been computed, each element of one shell needs to be classified in relation to the other shell. Each edge and vertex will be classified as *in*, *out* or *on* the other shell. The loop and face have extra classification, they can be *in*, *out*, *on*

shared or *on anti-shared*, depending on the orientation. This step is executed twice, once for the shell A elements and another for shell B elements.

Vertex is the first one to be classified. The *raycasting* technique is applied to classify a vertex from shell A, when it is not *on* a shell B. This technique is shown in Figure 2.10.

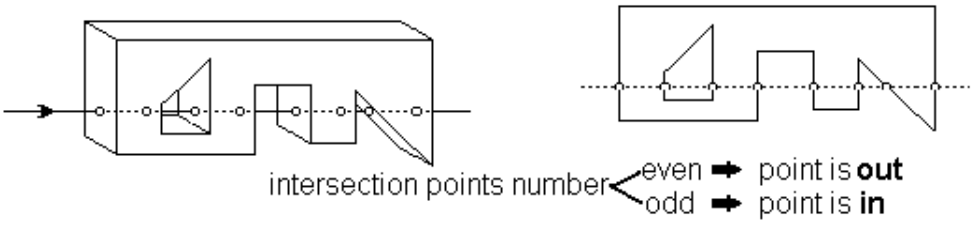


Figure 2.10: Raycasting technique

In *raycasting*, an arbitrary ray passing by a vertex of shell A is chosen and the number of intersections of this ray in shell B is evaluated. If it is *even*, the point is classified as *out*, and if it is *odd*, the point receives *in* as its classification.

When the vertex classification is over, the edges classification starts. This classification is done applying Table 2.2. Whether one vertex is not *on* the other shell, the edge receives this vertex classification. For *on/on* endpoints classification, the edge mid point needs to be classified by *raycasting* and the edge receives its classification. If the intersecting process was correctly executed the classification *in/out* can not exist. In Table 2.2, "RAY" means *raycasting* and "ER", error.

Table 2.2: Edges Classification [Arb90]

Endpoints	*OUT/OUT*	*OUT/ON*	*ON/ON*	*IN/ON*	*IN/IN*	*OUT/IN*
Edges	*OUT*	*OUT*	*RAY*	*IN*	*IN*	*ER*

The loops are classified according to their edges classification. A loop with one vertex has its classification. Whether one loop edge is not *on* the other shell the loop receives its classification. If a loop has edges with *in* and *out* classification that means error in the intersecting process. When all loop edges are *on*, the loop is *on* and an equal loop exists in the other shell. The orientation of these loops are compared and the loop receives an extra classification: *shared* for the same orientation and *anti-shared* for opposite orientation. To finish, the face will receive the extern loop classification. All possible combinations for triangular faces are presented in Table 2.3.

Table 2.3: Faces Classification [Arb90]

Edge	*OUT*	*OUT*	*OUT*	*OUT*	*OUT*	*OUT*	*OUT*	*OUT*	*OUT*
Edge	*OUT*	*OUT*	*OUT*	*IN*	*IN*	*IN*	*ON*	*ON*	*ON*
Edge	*OUT*	*ON*	*ON*	*OUT*	*IN*	*ON*	*OUT*	*IN*	*ON*
Face	*OUT*	*ER*	*OUT*	*ER*	*ER*	*ER*	*OUT*	*ER*	*OUT*
Edge	*IN*	*IN*	*IN*	*IN*	*IN*	*IN*	*IN*	*IN*	*IN*
Edge	*OUT*	*OUT*	*OUT*	*IN*	*IN*	*IN*	*ON*	*ON*	*ON*
Edge	*OUT*	*IN*	*ON*	*OUT*	*IN*	*ON*	*OUT*	*IN*	*ON*
Face	*ER*	*ER*	*ER*	*ER*	*IN*	*IN*	*ER*	*IN*	*IN*
Edge	*ON*	*ON*	*ON*	*ON*	*ON*	*ON*	*ON*	*ON*	*ON*
Edge	*OUT*	*OUT*	*OUT*	*IN*	*IN*	*IN*	*ON*	*ON*	*ON*
Edge	*OUT*	*IN*	*ON*	*OUT*	*IN*	*ON*	*OUT*	*IN*	*ON*
Face	*OUT*	*ER*	*OUT*	*ER*	*IN*	*IN*	*OUT*	*IN*	*ON*

2.2.4.4 Boolean Evaluation and Elimination of Undesired Elements

The previous steps are common for all Boolean and assembly operations. Now, each operation has a different treatment.

Boolean evaluation consists in deciding which elements remains and which will be eliminated. Each face received one of the eight possible classification. The A elements are: *onAinB*, *onAonBshared*, *onAonBanti-shared* or *onAoutB*, while the B elements are: *onBinA*, *onBonAshared*, *onBonAanti-shared* or *onBoutA*. For each classification and operation, an action is performed, Table 2.4 shows the necessary action for each face classification. In this table, $A + B$ corresponds to the assembly operation and "A-Shared" means *anti-shared*. The Euler operators are used to update the B-rep data.

Table 2.4: Decision Table for Boolean and Assembly Operations [Muu91]

A	B	$A-B$	$A \cup B$	$A \cap B$	$A+B$
ON	*IN*	*kill*	*kill*	*retain*	*retain*
ON	*ON SHARED*	*kill*	*retain*	*retain*	*retain*
ON	*ON A-SHARED*	*retain*	*kill*	*kill*	*retain*
ON	*OUT*	*retain*	*retain*	*kill*	*retain*
IN	*ON*	*retain+flip*	*kill*	*retain*	*retain*
ON SHARED	*ON*	*kill*	*kill*	*kill*	*kill*
ON A-SHARED	*ON*	*kill*	*kill*	*kill*	*kill*
OUT	*ON*	*kill*	*retain*	*kill*	*retain*

After implementing all Boolean evaluation steps, a correct triangular mesh is achieved. This mesh may be used as the input to another operation or to the volumetric mesh generator, if the mesh quality is sufficiently good.

2.3 Finite Element Mesh

As defined in the section 2.1, a finite element mesh is a discretization of a geometric domain into small simple shapes. Discretizing an object is a more difficult problem than it appears at first glance. A useful mesh should satisfy constraints that sometimes seem almost contradictory. A mesh must conform the object and ideally should meet constraints on size and shape of its elements.

Consider first the goal of correctly modeling the shape of a solid. Scientists and engineers often wish to model objects with complex shapes and possibly with curved surfaces. Boundaries may appear in the interior of a region as well as on its exterior surfaces. They are typically used to separate regions that have different physical properties. In practice, curved boundaries can often be approximated by piecewise planar boundaries, so theoretical mesh generation algorithms are often based upon the idealized assumption that the input geometry is piecewise linear. Given an arbitrary straight-line two dimensional region, it is not difficult to generate a triangulation that conforms to the shape of the region. But, finding a tetrahedralization that conforms to an arbitrary linear three-dimensional region is trickier [She97].

Another goal of mesh generation is that the elements should be relatively well shaped. Elements with small angles may degrade the quality of the numerical solution, because they can make the system of algebraic equations ill-conditioned. Whether a system of equations is ill-conditioned, roundoff error degrades the accuracy of the solution if the system is solved by direct methods, and the convergence is slow if the system is solved by iterative methods. By placing a lower bound on the smallest angle of a triangulation, one is also bounding the largest angle; for instance, in two dimensions, if no angle is smaller than θ, then no angle is larger than $180 - 2\theta$. Hence, many mesh generation algorithms take the approach of attempting to bound the smallest angle.

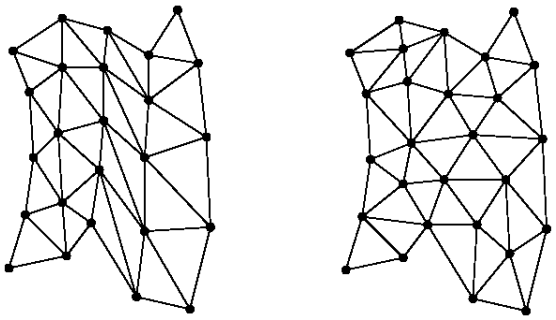
Satisfying both constraints of element size and shape are difficult, because the elements must meet only at their vertices. The edge of one triangular element can not be a portion of an edge of an adjoining element and also a face can not be a portion of other face as Figure 2.11 illustrates. There are variants of methods like the finite element method that permit such noncompatible elements. However, such elements are not preferred, as they may degrade or ruin the convergence of the method. Although elements that are not

compatible make it easier to create a mesh with seemingly nicely shaped elements, the problems of numerical error may still persist.

(a) 2D (b) 3D

Figure 2.11: Elements are not permitted to meet in the manner depicted here [She97]

The meshes can be categorized as *structured* and *unstructured* by the way the elements meet. A structured mesh is one in which the elements have the topology of a regular grid. Strictly speaking, it can be recognized by all interior nodes of the mesh having equal number of adjacent elements. Structured meshes are typically easier to compute with, but may require more elements or worse-shaped elements. Unstructured meshes, on the other hand, relaxes the node valence requirement, allowing any number of elements to meet at a single node. Figure 2.12 illustrates an example of each.



(a) Structured mesh (b) Unstructured mesh

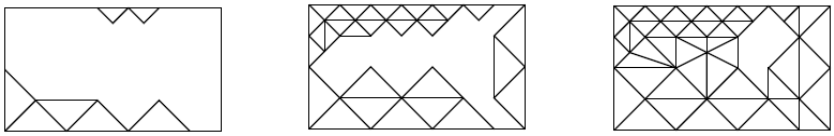
Figure 2.12: Type of meshes [She97]

Triangle and tetrahedral meshes are common when referring to unstructured meshing, although quadrilateral and hexahedral meshes can also be unstructured. While there is certainly some overlap between structured and unstructured mesh generation technologies, the main feature that distinguish the two fields are the unique iterative smoothing algorithms employed by structured grid generators. The generation of both structured and unstructured meshes can be surprisingly difficult, each posing challenges of their own.

This work focus on unstructured meshes, more precisely on automatic methods to improve the quality of unstructured surface meshes and consequently improve the volu-

metric mesh quality. Before studying methods to improve the mesh, it is important to understand how meshes can be generated and what to expect from their quality.

The most popular approaches to triangular and tetrahedral mesh generation can be classified in three classes: advancing front methods, methods based on grids, quadtrees and octrees and Delaunay methods. The advancing front methods begin by dividing the boundaries of the mesh into edges (in 2D) or triangles (in 3D). These discretized boundaries form the initial front. Triangles or tetrahedra are generated one-by-one, starting from the boundary edges or faces and work toward the center of the region being meshed, as illustrated in Figure 2.13.



(a) (b) (c)

Figure 2.13: Three stages in the progression of an advancing front algorithm [She97]

A quadtree is a recursive data structure used to efficiently manipulate multiscale geometric objects in the plane. Quadtree recursively partition a region in axis-aligned squares. A top-level square called root encloses the entire object. Each quadtree square can be divided into four child squares, which can be divided in turn, as Figure 2.14 shows. Octrees are the generalization of quadtrees to three dimensions; each cube in an octree can be subdivided into eight cubes.

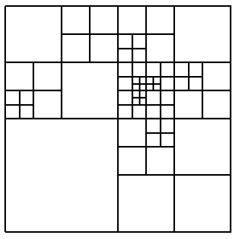


Figure 2.14: A quadtree [She97]

The Delaunay triangulation of a vertex set maximizes the minimum angle among all possible triangulations of that vertex set. The Delaunay tetrahedralization is not quite so effective as the Delaunay triangulation at producing elements of good quality, but it has nearly as much popularity as the 2D method. Both methods are detailed in the next two subsections.

Straddled between 2D and 3D meshes, there is the surface mesh, which has it's own set of issues. Subsection 2.3.3 presents some methods for surface mesh generation and explains some problems that may arise during the process.

2.3.1 Delaunay Triangulation

Let a triangulation of a set V of vertices be a set T of triangles, whose vertices collectively form V, the interiors do not intersect each other, and the union completely fills the convex hull of V.

The Delaunay triangulation D of V, is the graph defined as follows. Any circle in the plane is said to be empty if it contains no vertex of V in its interior, but they are permitted on the circle. Let u and v be any two vertices of V . The edge uv is in D if and only if there is an empty circle that passes through u and v. An edge satisfying this property is said to be Delaunay. Figure 2.15 illustrates a Delaunay triangulation.

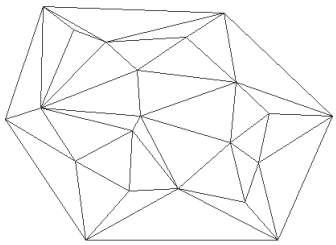


Figure 2.15: A Delaunay triangulation

The Delaunay triangulation of a vertex set is clearly unique, because the definition given above specifies an unambiguous test for the presence or absence of an edge in the triangulation. Every edge of the convex hull of a vertex set is Delaunay. Figure 2.16 illustrates the reason why. For any convex hull edge e, it is always possible to find an empty circle that contains e by starting with the smallest containing circle of e and *growing* it away from the triangulation.

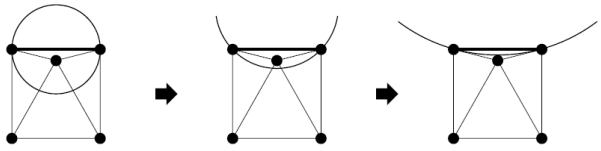


Figure 2.16: Each edge on the convex hull is Delaunay [She97]

Every edge connecting a vertex to its nearest neighbor is Delaunay. If w is the v

nearest vertex, the smallest circle passing through v and w does not contain any vertices.

It is not at all obvious that the set of Delaunay edges of a vertex set collectively forms a triangulation. For the definition, the Delaunay triangulation is guaranteed to be a triangulation only if the vertices of V are in general position, which means that no four vertices of V lie on a common circle. As a first step to proving this guarantee, it is important to know the notion of a Delaunay triangle. The circumcircle of a triangle is the unique circle that passes through all three of its vertices. A triangle is said to be Delaunay if and only if its circumcircle is empty. This defining characteristic of Delaunay triangles, illustrated in Figure 2.17, is called the empty circumcircle property.

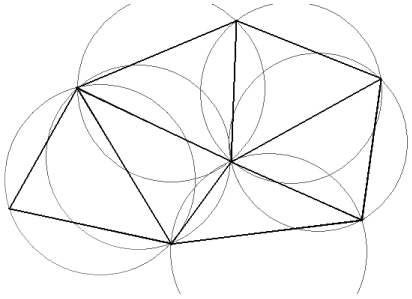
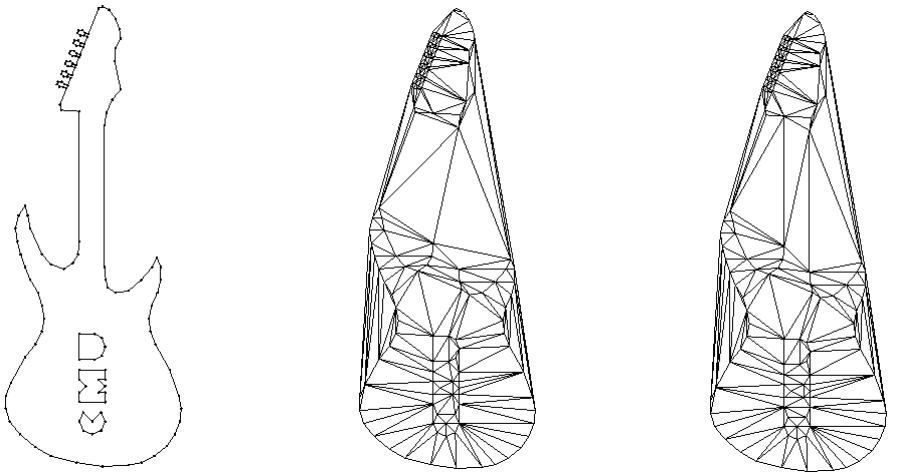


Figure 2.17: Every triangle of a Delaunay triangulation has an empty circumcircle

Commonly, 2D mesh generators uses a planar straight line graph (PSLG) as input. PLSG is defined to be a collection of vertices and segments, where the endpoints of every segment are included in the list of vertices. Figure 2.18(a) illustrates a PSLG defining an electric guitar. Figure 2.18(b) shows the Delaunay triangulation of the electric guitar and Figure 2.18(c) illustrates its constrained Delaunay triangulation. The constrained Delaunay triangulation is similar to the regular Delaunay triangulation, but all the input PSLG segments are in the final Delaunay triangulation. Each segment of the PLSG is inserted by deleting the triangles that it overlaps, and re-triangulating the regions on each side of the segment. No new vertices are inserted.

2.3.2 Delaunay Tetrahedralization

The Delaunay tetrahedralization is one of the most important methods for volumetric mesh generation. Before defining it, the input upon which the algorithm will operate must be defined. Many programs use a generalization of PSLG called a piecewise linear complex (PLC) [She97].



(a) Electric guitar PSLG

(b) Delaunay triangulation of vertices of PSLG

(c) Constrained Delaunay triangulation of PSLG

Figure 2.18: PSLG example and its Delaunay triangulation with and without constraints [She97]

In three dimensions, a PLC (Figure 2.19) is a set of vertices, segments, and facets. The facets can be quite complicated in shape, may have any number of sides, may be non-convex and may have holes, slits, or vertices in its interior. However, a fixed requirement is that a facet must be planar. When a PLC is tetrahedralized, each facet of the PLC is partitioned into triangular subfacets, respecting the holes, slits and vertices.

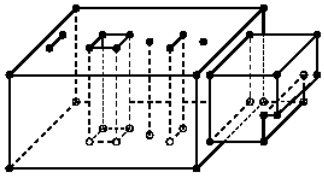


Figure 2.19: A piecewise linear complex (PLC) [She97]

A PLC X is required to have the following properties:

- For any facet in X, every edge and vertex of the facet must appear as a segment or vertex of X. Hence, all facets are segment-bounded;
- X contains both endpoints of each segment of X;
- X is closed under intersection. Hence, if two facets of X intersect at a line segment, that line segment must be represented by a segment of X. If a segment or facet of X intersects another segment or facet of X at a single point, that point must be represented by a vertex in X;

- If a segment of X intersects a facet of X at more than a finite number of points, then the segment must be entirely contained in the facet. A facet cannot be bounded by a segment that extends beyond the boundary of the facet.

The Delaunay tetrahedralization of a PLC is a straightforward generalization of the Delaunay triangulation to three dimensions. A tetrahedron whose vertices are members of V is said to be Delaunay if there is an empty sphere that passes through all its vertices. If no five vertices are cospherical, the Delaunay tetrahedralization is a tetrahedralization and is unique. If cospherical vertices are present, it is customary to define the Delaunay tetrahedralization to be a true tetrahedralization. As with degenerate Delaunay triangulations, a subset of the Delaunay edges, faces, and tetrahedra may have to be omitted to achieve this, thus sacrificing uniqueness. The definition of Delaunay triangulation generalizes to dimensions higher than three as well.

In general, volumetric mesh generators that uses constrained Delaunay algorithm can add new points and edges to the surface mesh, when it is necessary, to generate and improve the volumetric mesh. However, vertices, edges, faces slits and holes from the input geometric description are part of the resulting mesh. When there is more freedom to triangulate the facets, the overall quality of the volumetric mesh can be improved and the number of new points and edges is reduced. This characteristic will be exploited in section 4.1 to improve the quality degree of volumetric meshes.

2.3.3 Surface Mesh Generation

Commonly, the surface meshes are input for volumetric mesh generators. Many of mesh generation problems involve the formation of elements on arbitrary three-dimensional surfaces or the modification of an existent mesh, like the application of Boolean and Assembly operations discussed at section 2.2.4.

The algorithms used for two dimensional mesh generation require some modification in order to be generalized for the use on three-dimensional surfaces or new algorithms need to be implemented. Surface mesh generation algorithms can be classified as either parametric space or direct 3D.

Parametric surfaces have an underlying $u - v$ representation (see section 2.5.1 for

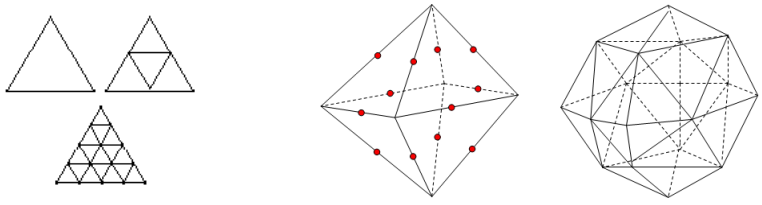
parametric surfaces). Parametric space algorithms will form elements in the two dimensional parametric space of the surface and as a final step, map the $u-v$ coordinates back to the world space, $x-y-z$ coordinates, [Owe98]. The main drawback to this method is that the elements formed in parametric space may not always form well-shaped elements in three dimensions once mapped back to the surface. To solve this, parametric surface mesh generators can do one of two things: 1) modify or re-parameterize the underlying parametric representation so there is a reasonable mapping from parametric space to world space; or 2) modify the mesh generation algorithm so that stretched or anisotropic elements meshed in 2D will map back to well-shaped, isotropic elements in 3D.

Direct 3D surface mesh generators form elements directly on the geometry without regard to the parametric representation of the underlying geometry. The next three subsections give a brief description of three techniques for direct 3D surface mesh generation.

2.3.3.1 Sphere Discretization

Sphere discretization is a well known method of recursive subdivision [Mag99]. It begins with a coarse approximation to the sphere, with only triangular faces, like an octahedron. Then, the process is done by subdividing the octahedron until the desired discretization level is achieved. New vertices are inserted in the middle of the triangles edges to create four new triangles. The new vertices are projected onto the sphere surface, improving the surface approximation. This discretization process is recursive and it is done as illustrated in Figure 2.20, until the desired level of approximation is reached. Looking carefully at how the process is executed from the top point of the triangle in Figure 2.20(a), one can see that each horizontal line or level curves has one vertex more than the previous level. Since each triangle at the initial step corresponds to one side of the octahedron, there are four vertex inclusions on each level curve. When the hemisphere is finished, the number of vertices starts to decrease by four per level, until one vertex is reached. The level curves can be seen as the covered trajectory of the arc vertices to generate the sphere.

This process generates meshes with good quality and good elements distribution.



(a) (b)

Figure 2.20: Recursive process of sphere discretization

2.3.3.2 Swept Primitives Discretization

The sphere discretization property of increment and decrement of vertices in the level curve permits to extend it to any surface generated by translational or rotational sweep. The sphere is a special case of the rotational sweep, when an arc is swept through a circular path. For a general rotational sweep, the vertices of any 2D primitive is swept through a circular path.

A cylinder is built by moving a circle by a linear path. In this case, the number of vertices per level remains the same during the trajectory. The cylinder top and bottom circles are meshed by the Delaunay triangulation method and connected to the mesh generated for the trajectory of the circle, as well as any top and bottom parts of any translational swept primitive.

The vertices of the generator profile should be well distributed in order to guarantee the generation of high quality elements.

2.3.3.3 Surface Reconstruction

Models with complex geometries can be generated by an acquisition process, such as medical imagery, laser range scanners, contact probe digitizers, radar and seismic surveys. These devices collect a cloud of unorganized sample points on the model surface. Surface reconstruction consists of turning the 3D point cloud into a surface, either triangulated, defined implicitly, or defined parametrically.

The problem of surface reconstruction can be stated as follows: Let S be a surface of object O, and assume that S is a smooth twice-differentiable two-dimensional manifold, embedded in Euclidean three-dimensional space $\mathbb{R}^3$. Given a discrete set of points P , $p_i \in P \subset \mathbb{R}^3, i = 1, ..., N$, that samples surface S, find a surface S' that approximates S,

using the data set P. The reconstructed mesh S' must be topologically equivalent to the surface S of the original object. For an overview of the problem see Figure 2.21.

Figure 2.21: Overview of the problem of the surface reconstruction [Hrá03]

Several algorithms have been proposed to solve this problem, Hrádek in [Hrá03] and Kuo and Yau in [Kuo05] give a good survey on them. The surface reconstruction algorithms can be classified into three main categories: Delaunay-based, implicit-surface and region-growing approaches. The Delaunay-based approach consists of two main steps: (i) first, it builds a Delaunay triangulation D of the set P; (ii) then it extracts a collection of facets from D that approximate S. Its main advantage is that the structural characteristics of the Delaunay triangulation complement the absence of geometric information in P. As a consequence, it is more robust and systematic than the other approaches.

The region-growing methods reconstruct the surface incrementally. They begin by initiating a triangle as an initial region and iterate to attach new triangles only on the regions boundaries. Although these methods are extremely fast due to keeping the Delaunay computation off, there exists a common drawback that the reconstruction quality heavily depends on the user-defined parameters, which vary with the sampling density and cannot be assigned easily. In addition, the region-growing method may still leave behind small holes after the growing procedure when poor data (for example, noise) exists, therefore it cannot guarantee the creation of a closed manifold model if no appropriate hole-filling post-processing procedure is used.

For the implicit surface approach, a signed distance function defined from sample points is first defined and computed, and then uses a zero-set of the signed distance function to construct an approximate triangulated surface with topology as the actual surface. Compared with the Delaunay-based and the region-growing methods, the implicit surface approach approximates rather than interpolates sample points and therefore limits its applications only to computer graphics and virtual reality. In CAD/CAM, however, accuracy of model representation is sometimes more important.

The Delaunay-based algorithms have been quite effective both in theory and practice. The crust and power crust algorithms of Amenta et al. in [Ame99, Ame01] fall into its category and are used int this work. First, Amenta et al. proposed a Voronoi-based surface reconstruction algorithm that performs well in two and three dimensions, the crust algorithm. It first calculates the extreme vertices or poles from the sample points, and then evaluate the Delaunay triangulations of the sample points and all the poles. Redundant triangular facets are then removed from the resulting tetrahedra.

Later on, Amenta et al. devised another new approach called power crust that takes sample points from the surface of three-dimensional object and produces a surface mesh and an approximate medial axis. The approach is to first approximate the medial axis transform (MAT) of the object, and then use an inverse transform to produce the surface representation from the MAT.

The reconstruction methods must guarantee the correct geometry and topology. But, commonly, they do not care about the shape quality of the mesh element, [Fre05]. So, most of the time, the resultant mesh is full of badly shaped elements, as the example in Figure 1.2 shows.

2.4 Mesh Quality Measures

The notion of a nicely shaped element varies depending on the numerical method, the type of problem being solved, and the polynomial degree of the piecewise functions used to interpolate the solution over the mesh.

Shewchuk [She02] and Tsukerman [Tsu97, Tsu98] present good surveys on how the shape of the finite elements influences numerical solutions accuracy. The main points are: for isotropic PDEs, small angles (but not large angles in the absence of small ones) can cause poor conditioning, that can be quantified in a way useful for comparing differently shaped elements; and that PDEs with anisotropic coefficients are best served by anisotropic elements. This work focus only on quality metrics related to isotropic PDEs.

The system of linear equations constructed by finite element discretization is solved using either iterative methods or direct methods. The speed of iterative methods, such as the Jacobi and conjugate gradient methods, depends on the conditioning of the global

stiffness matrix: a larger condition number implies slower performance. Direct solvers rarely vary as much in their running time, but the solutions they produce can be inaccurate due to roundoff error in floating-point computations, and the size of the roundoff error is roughly proportional to the condition number of the stiffness matrix. As a rule of thumb, Gaussian elimination loses one decimal digit of accuracy for every digit in the integer part of the condition number. These errors can be countered by using higher-precision floating-point numbers.

The global stiffness matrix K is termed ill-conditioned when $10^{-s}\kappa = 1$, where s denotes the number of decimals in the computer and κ the condition number of K, [Fri72].

For some applications that use direct solvers, the degree of accuracy required might be small enough, or the floating-point precision great enough, that a poorly conditioned stiffness matrix is not an impediment. Usually, though, conditioning should be kept under control.

The finite element method is different for every partial differential equation and so is the relationship between element shape and matrix conditioning. Although the approach discussed here is valid for a wide class of electromagnetic problems, as a reference, the Poisson equation will be studied

$$-\nabla^2 f(p) = \eta(p), \tag{2.2}$$

where $\eta(p)$ is a known function of p, and the goal is to find an approximation $h(p)$ of the unknown function $f(p)$ for some boundary conditions.

In the Galerkin formulation of FEM, the piecewise approximation h is composed from local piecewise basis functions, which are in turn composed from shape functions. Each shape function is defined on just one element. If h is piecewise linear, the shape functions are the barycentric coordinates $\omega_i(p)$.

For each element t, the finite element method constructs a $(d+1) \times (d+1)$ element stiffness matrix K_t, where d is the dimension. The element stiffness matrices are assembled into an $n \times n$ global stiffness matrix K, where n is the number of mesh vertices - for Poisson's equation on linear elements. The assembly process sums the entries of each element stiffness matrix into the entries of the global stiffness matrix.

The difficulty of solving the linear system of equations associated with K typically grows with K's condition number $\kappa = \lambda^K_{max}/\lambda^K_{min}$, where λ^K_{max} and λ^K_{min} are the largest and smallest eigenvalues of the stiffness matrix K.

λ^K_{min} is closely tied to properties of the physical system being modeled, and to the sizes of the elements. Fried [Fri72] offers a lower bound for λ^K_{min} that is proportional to the area or volume of the smallest element, and an upper bound proportional to the largest element. Fortunately, λ^K_{min} is not strongly influenced by element shape, but badly shaped elements often do have tiny areas or volumes.

In contrast, λ^K_{max} can be made arbitrarily large by a single badly shaped element. For each element t, let λ^t_{max} be the largest eigenvalue of its element stiffness matrix. Let m be the maximum number of elements meeting at a single vertex. λ^K_{max} is related to the largest eigenvalues of the element stiffness matrices as follows

$$\max_t \lambda^t_{max} \leq \lambda^K_{max} \leq m \max_t \lambda^t_{max} \tag{2.3}$$

then, the condition number κ is proportional to the largest eigenvalue among the element stiffness matrices.

The element stiffness matrix for a linear triangle is

$$K_t = A \begin{bmatrix} \nabla\omega_1 \cdot \nabla\omega_1 & \nabla\omega_1 \cdot \nabla\omega_2 & \nabla\omega_1 \cdot \nabla\omega_3 \\ \nabla\omega_2 \cdot \nabla\omega_1 & \nabla\omega_2 \cdot \nabla\omega_2 & \nabla\omega_2 \cdot \nabla\omega_3 \\ \nabla\omega_3 \cdot \nabla\omega_1 & \nabla\omega_3 \cdot \nabla\omega_2 & \nabla\omega_3 \cdot \nabla\omega_3 \end{bmatrix} \tag{2.4}$$

Geometrically, each $\nabla\omega_1$ is equal to $1/a_i$, where a_i is the altitude of vertex i of the triangular element. Then, Equation 2.4 becomes

$$K_t = \frac{1}{8A} \begin{bmatrix} 2l_1^2 & l_3^2 - l_1^2 - l_2^2 & l_2^2 - l_1^2 - l_3^2 \\ l_3^2 - l_1^2 - l_2^2 & 2l_2^2 & l_1^2 - l_2^2 - l_3^2 \\ l_2^2 - l_1^2 - l_3^2 & l_1^2 - l_1^2 - l_3^2 & 2l_3^2 \end{bmatrix} \tag{2.5}$$

The triangle side lengths can be written in terms of the triangles internal angles and

it can be shown that

$$K_t = \frac{1}{2}\begin{bmatrix} \cot\theta_2 + \cot\theta_3 & -\cot\theta_3 & -\cot\theta_2 \\ -\cot\theta_3 & \cot\theta_1 + \cot\theta_3 & -\cot\theta_1 \\ -\cot\theta_2 & -\cot\theta_1 & \cot\theta_1 + \cot\theta_2 \end{bmatrix} \tag{2.6}$$

The sum of the eigenvalues of K_t equals the sum of the diagonal entries. Both the eigenvalues and diagonal entries are nonnegative, so λ^t_{max} is large if and only if at least one of the diagonal entries is large. If one of the angles approaches 0°, its cotangent approaches infinity, and so does λ^t_{max}. Therefore, small angles can ruin matrix conditioning. Of course, if one of the angles approaches 180°, the other two angles approach 0°.

Similar manipulation can be done for linear tetrahedron, the element stiffness matrix is

$$\begin{aligned} K_t &= V\begin{bmatrix} \nabla\omega_1\cdot\nabla\omega_1 & \nabla\omega_1\cdot\nabla\omega_2 & \nabla\omega_1\cdot\nabla\omega_3 & \nabla\omega_1\cdot\nabla\omega_4 \\ \nabla\omega_2\cdot\nabla\omega_1 & \nabla\omega_2\cdot\nabla\omega_2 & \nabla\omega_2\cdot\nabla\omega_3 & \nabla\omega_2\cdot\nabla\omega_4 \\ \nabla\omega_3\cdot\nabla\omega_1 & \nabla\omega_3\cdot\nabla\omega_2 & \nabla\omega_3\cdot\nabla\omega_3 & \nabla\omega_3\cdot\nabla\omega_4 \\ \nabla\omega_4\cdot\nabla\omega_1 & \nabla\omega_4\cdot\nabla\omega_2 & \nabla\omega_4\cdot\nabla\omega_3 & \nabla\omega_4\cdot\nabla\omega_4 \end{bmatrix} \\ &= \frac{1}{6}\begin{bmatrix} \sum_{1\neq i<j} l_{ij}\cot\theta_{ij} & -l_{34}\cot\theta_{34} & -l_{24}\cot\theta_{24} & -l_{23}\cot\theta_{23} \\ -l_{34}\cot\theta_{34} & \sum_{2\neq i<j\neq 2} l_{ij}\cot\theta_{ij} & -l_{14}\cot\theta_{14} & -l_{13}\cot\theta_{13} \\ -l_{24}\cot\theta_{24} & -l_{14}\cot\theta_{14} & \sum_{3\neq i<j\neq 3} l_{ij}\cot\theta_{ij} & -l_{12}\cot\theta_{12} \\ -l_{23}\cot\theta_{23} & -l_{13}\cot\theta_{13} & -l_{12}\cot\theta_{12} & \sum_{i<j\neq 4} l_{ij}\cot\theta_{ij} \end{bmatrix} \end{aligned} \tag{2.7}$$

If one of the dihedral angles (angle between the adjacent faces of an element) approaches 0°, its cotangent approaches infinity, and so does λ^t_{max}. Unlike with triangles, it is possible for one dihedral angle of a tetrahedron to be arbitrarily close to 180° without any dihedral angle of the tetrahedron being small. Surprisingly, such a tetrahedron does not induce a large eigenvalue in K_t. An angle approaching 0° has a cotangent approaching infinity, but an angle approaching 180° has a cotangent approaching negative infinity. Each entry on the diagonal of K_t is nonnegative and has the form $\sum_{i,j} l_{ij}\cot\theta_{ij}$. Therefore, if t has no dihedral angle near 0°, the diagonal entries of K_t are bounded and thus so is λ^t_{max}. It does not matter that t has planar angles near 0°.

For FEM, the condition number of the stiffness matrix associated with the method

should be kept as small as possible. The element shape has a strong influence on the matrix conditioning; small angles are deleterious, while large ones (alone) are not. The equilateral elements are preferred, since the angles are maximized.

A mesh with only one or two bad elements will typically impose only a few large eigenvalues; the rest of the spectrum of K may lie within limited range. Some iterative solvers for systems with linear equations can take advantage of this, and some cannot. Primitive methods like Jacobi method or steepest descent behave poorly whenever the condition number of K is large, even if only one bad eigenvalue is responsible. By contrast, some Krylov subspace methods perform well in these circumstances.

Bern and Plassmann [Ber99] discusses another property of the linear system that element shape affects, besides condition number. A triangular mesh with well-shaped elements gives a symmetric *M-matrix* - positive definite with negative off-diagonal entries - for a finite element formulation of an equation with a Laplacian operator. M-matrices are exactly those matrices that satisfy a discrete maximum principle; this desirable property rules out oscillation of the numerical method. In this case, "well-shaped" has a precise meaning: the two angles opposite each interior edge of the mesh should sum to at most 180°. This requirement implies that the triangulation must be Delaunay or constrained Delaunay triangulation. Depending on the boundary conditions associated with the differential equation, an M-matrix may also require that the single angle opposite a boundary edge should measure at most 90°.

In three dimensions, an unstructured tetrahedral mesh gives an M-matrix if and only if, for each edge e' in the mesh, the sum $\sum_e |e| \cot \theta_e$ is nonnegative, where the sum is over all edges e that are opposite to e' in the tetrahedra of the mesh, $|e|$ denotes the length of e and θ_e the dihedral angle at e. All such sums will be nonnegative if all dihedral angles in the mesh are non obtuse. This condition is sufficient but not a necessary condition.

Following, the most commonly used quality measures for first order triangular and tetrahedral elements are summarized. Tsukerman [Tsu97, Tsu98] proved that some of these metrics are particular cases of a general maximum eigenvalue criterion. These metric were defined for isotropic PDEs. But, they may be used as an *a priori* error estimate for anisotropic problems and for isoparametric or high order elements, where no other shape conditions are immediately available.

2.4.1 Triangular Elements

2.4.1.1 The Minimum/Maximum Angle

The minimum/maximum angle is a straightforward index for measuring mesh quality. An element with angles near 0° or 180° will create difficulties in the process of finite element analysis. Either the minimum angle is maximized or the maximum angle is minimized in order to eliminate severely distorted elements.

2.4.1.2 The Aspect Ratio

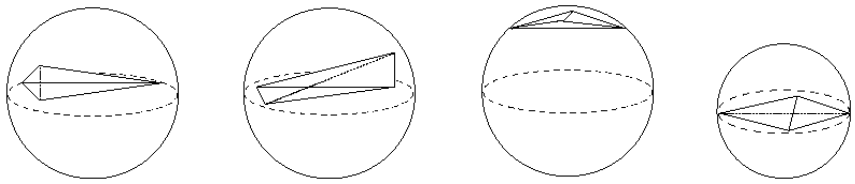
The aspect ratio is the radius ratio of the circumscribed circle to the inscribed circle of a mesh element. An equilateral triangle has the optimal aspect ratio of 2.0. When the element becomes more distorted, the aspect ratio increases.

2.4.1.3 The Distortion Metrics

The metrics are related to the area, internal angle or the edge length of elements. The shape quality of an element can be evaluated quantitatively by using this type of metrics. An equilateral triangle has the optimal value of 1, and a severely distorted element has a value near zero. These metrics can also be used for quadrilateral elements.

2.4.2 Tetrahedral Elements

A tetrahedron can have very small angles either because its vertices are close to a line, or, if that is not the case, its vertices are close to a plane. In the former case, the tetrahedron is said to be *skinny*. In the latter case it is said to be *flat*. Skinny tetrahedra include the *needle* and the *wedge* tetrahedra illustrated in Figures 2.22(a) and 2.22(b), respectively. While flat tetrahedra include the *cap* and the *sliver* tetrahedra presented in Figures 2.22(c) and 2.22(d). A flat tetrahedron has small angles but is not skinny. The needle, the wedge, and the cap have circumradius much larger than their shortest edge, but the sliver does not. However, the ratio of the circumradius of a sliver to the radius of its inscribed sphere could be as large as possible.



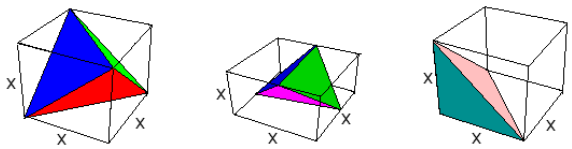
(a) Needle (b) Wedge (c) Cap (d) Sliver

Figure 2.22: Tetrahedra with very small angles

Some quality metrics for tetrahedral elements are presented ahead. They can be used separately or together to identify bad shaped elements.

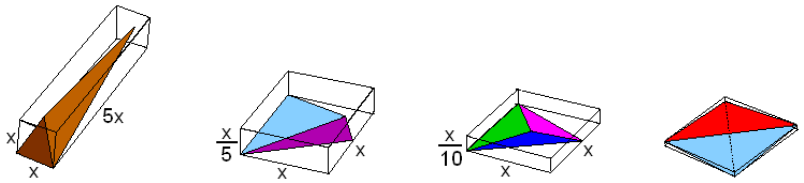
2.4.2.1 Radius-edge ratio

A tetrahedron has a unique circumsphere. Let R be its radius and L the length of the shortest edge. Then the radius-edge ratio is defined as $\rho = R/L$ and it is effective for measuring the quality of a tetrahedron. This value is small for all well-shaped tetrahedra (Figure 2.23), while for most of badly-shaped tetrahedra (Figure 2.24), it is large. The smallest ratio is $\rho = \sqrt{6}/4 \approx 0.612$, which occurs when the tetrahedron is equilateral, Figure 2.23(a). Unfortunately, the *sliver* (Figure 2.24(d)) can have a radius-edge ratio as small as $\sqrt{2}/2 \approx 0.707$ [Si04] and it can not be identified by this metric. The following quality metrics can be used to identify this element.



(a) $\rho \approx 0.612$ (b) $\rho \approx 0.645$ (c) $\rho \approx 0.866$

Figure 2.23: The radius-edge ratios (ρ) of some well-shaped tetrahedra [Si04]



(a) $\rho \approx 2.539$ (b) $\rho \approx 3.165$ (c) $\rho \approx 3.783$ (d) $\rho \approx 0.707$ (*Sliver*)

Figure 2.24: The radius-edge ratios of some badly-shaped tetrahedra [Si04]

2.4.2.2 Aspect ratio

The aspect ratio of an element is usually defined as the ratio of the radius of its circumsphere, R, to the radius of its inscribed sphere, r, $\vartheta = R/r$. The smaller the aspect ratio, the better the tetrahedron is. Normally, tetrahedra with small radius-edge ratio also have small aspect ratio. *Slivers* are the only tetrahedra that have small radius-edge ratio but have large aspect ratio.

2.4.2.3 Dihedral Angle

Another widely used quality metric is the minimum dihedral angle, ϕ. It is defined as the angle between the adjacent faces of an element.

Let $\vec{n_1}$ and $\vec{n_2}$ be the normal vector of two adjacent faces, the dihedral angle is:

$$\phi = \cos^{-1}\left(\frac{\vec{n_1} \cdot \vec{n_2}}{|\vec{n_1}|\,|\vec{n_2}|}\right). \tag{2.8}$$

The regular tetrahedron maximizes the minimum dihedral angle at $\phi = \cos^{-1}(1/3) = 70.528...^{\circ}$, while the *slivers* has its almost zero.

2.5 Surfaces

During the study of methods to improve the quality of the surface meshes, the necessity of evaluating an approximation of the treated model surface arose. This approximation should be able to supply the necessary information of the model surface in order to maintain the model geometric properties during the mesh improvement process. The idea of using smooth approximations came from their ability of representing a large variety of shapes and their extensive use in CAD systems.

This section begins with a brief review on general surface theory an its two special cases, the Bezier and the B-splines surfaces. In the sequence, the technique to evaluate the smooth surface approximation of a model surface based on least squares approximation is introduced.

2.5.1 Surface Representation

The simplest mathematical element used to model a surface is a *patch*. A patch is a curve-bounded collection of points whose coordinates are given by continuous, two parameter, single valued mathematical functions of the form

$$x = x(u, v) \quad y = y(u, v) \quad z = z(u, v) \tag{2.9}$$

The parametric variables u and v are constrained to the intervals $u, v \in [0, 1]$. Figure 2.25 illustrates a patch.

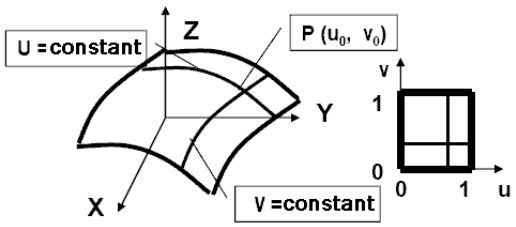


Figure 2.25: Patch

Associated with every patch is a set of boundary conditions. For an ordinary patch, there are always four and only four corner points and edge curves. This follows from the possible combinations of the two parametric variables limits.

There are many examples of nonparametric surfaces; for example, an equation of the form

$$F(x, y, z) = 0 \tag{2.10}$$

is the implicit equation of a surface. If this equation is linear in all variables, then the surface is an unbounded plane. If it is a second-degree equation, then the surface is a quadric, of which the sphere is a special case. Finally, if one of the variables is missing from the equation, the surface must be a cylinder whose generators are parallel to the axis of the missing variable.

When we solve the implicit equation for one of the variable as a function of the other two, say, for z as a function of x an y, it is

$$z = f(x, y). \tag{2.11}$$

which is the explicit equation of the same surface in Equation 2.10.

Although the implicit and explicit forms are useful in representing surfaces, they suffer from inherent weakness: their inability to represent an easily transformable and bounded surface. However, one can either directly adapt or closely approximate most, if not all, of these classical surfaces by a parametric formulation.

Another advantage of the parametric representation of a surface is the complete control of the domain of a surface modeling operation by an appropriate choice of the parametrization scheme. By carefully specifying subsets of a particular domain $[u_{min}, u_{max}] \times [v_{min}, v_{max}]$, one can readily define certain sections of a surface. This feature is useful whenever a surface is composed of several patches.

The algebraic form of a bicubic patch is given by

$$\mathbf{p}(u,v) = \sum_{i=0}^{3}\sum_{j=0}^{3} \mathbf{a}_{ij} u^i v^j \quad u,v \in [0,1] \tag{2.12}$$

The $\mathbf{a}_{ij}$ vectors are called the algebraic coefficients of the surface. The parametric variables u and v are restricted by definition to values in the interval $[0,1]$. Both parametric variables can be cubic terms.

The expansion of Equation 2.12 gives:

$$\begin{aligned} \mathbf{p}(u,v) &= \mathbf{a}_{33}u^3v^3 + \mathbf{a}_{32}u^3v^2 + \mathbf{a}_{31}u^3v + \mathbf{a}_{30}u^3 \\ &+ \mathbf{a}_{23}u^2v^3 + \mathbf{a}_{22}u^2v^2 + \mathbf{a}_{21}u^2v + \mathbf{a}_{20}u^2 \\ &+ \mathbf{a}_{13}u^1v^3 + \mathbf{a}_{12}u^1v^2 + \mathbf{a}_{11}uv + \mathbf{a}_{10}u \\ &+ \mathbf{a}_{03}v^3 + \mathbf{a}_{02}v^2 + \mathbf{a}_{01}v + \mathbf{a}_{00} \end{aligned} \tag{2.13}$$

The 16-term polynomial in u and v defines the set of all points lying on the surface. Since each $\mathbf{a}$ coefficient has three independent components, there is a total of 48 algebraic coefficients or 48 degrees of freedom. Thus, each vector component is simply

$$\mathbf{x}(u,v) = \mathbf{a}_{33_x}u^3v^3 + \mathbf{a}_{32_x}u^3v^2 + \mathbf{a}_{31_x}u^3v + ... + \mathbf{a}_{00_x} \tag{2.14}$$

There are similar expressions for $y(u,v)$ and $z(u,v)$.

The algebraic form in matrix notation is

$$\mathbf{p} = \mathbf{U}\mathbf{A}\mathbf{V}^T \tag{2.15}$$

where

$$\mathbf{U} = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \tag{2.16}$$

and

$$\mathbf{V} = \begin{bmatrix} v^3 & v^2 & v & 1 \end{bmatrix} \tag{2.17}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_{33} & \mathbf{a}_{32} & \mathbf{a}_{31} & \mathbf{a}_{30} \\ \mathbf{a}_{23} & \mathbf{a}_{22} & \mathbf{a}_{21} & \mathbf{a}_{20} \\ \mathbf{a}_{13} & \mathbf{a}_{12} & \mathbf{a}_{11} & \mathbf{a}_{10} \\ \mathbf{a}_{03} & \mathbf{a}_{02} & \mathbf{a}_{01} & \mathbf{a}_{00} \end{bmatrix} \tag{2.18}$$

Note that the subscripts of the vector elements in the **A** matrix correspond to those in Equation 2.13.

The algebraic coefficients of a patch determine its shape and position in space. Patches of the same size and shape have a different set of coefficients if they occupy different positions in space. A change on one of the 48 coefficients results in a completely different patch.

A patch consists of an infinite number of points given by their x,y,z coordinates. There is also an infinite number of pairs of u,v values in the corresponding parametric space. But, there is a unique pair u,v values associated with each point in object space.

The parametric surfaces can be associated to two kinds of continuity:

- Geometric Continuity

  G^0 : the end positions of two surfaces coincide;

  G^1 : the first derivatives are proportional at a joint.

- Parametric Continuity

  C^0 : the same as G^0;

C^1 : the first derivatives are equal at a joint;

C^n : both the direction and the magnitude of the n^{th} derivate of the surface $(d^n S(u)/du^n)$ are the same at a joint.

The next two subsections present two special cases of parametric surfaces, the Bezier and the B-Splines surfaces. They were extensively studied in the last decades and are largely used in CAD systems.

2.5.2 Bezier Surface

The Bezier surface has a characteristic polyhedron. Points on a Bezier surface are given by

$$\mathbf{s}(u,v) = \sum_{i=0}^{m}\sum_{j=0}^{n} \mathbf{c}_{ij} B_{i,m}(u) B_{j,n}(v) \quad u,v \in [0,1] \tag{2.19}$$

where the $\mathbf{c}_{ij}$ are vertices of the characteristic polyhedron that form an $(m+1)\times(n+1)$ rectangular array of points and $B_{i,m}$ e $B_{j,n}$ are the blending functions, defined as

$$B_{i,m}(u) = \frac{m!}{i!(m-i)!} u^i (1-u)^{m-i} \tag{2.20}$$

$$B_{j,n}(v) = \frac{n!}{j!(n-j)!} v^j (1-v)^{n-j} \tag{2.21}$$

Figure 2.26 shows blending-function curves for $m = 3$.

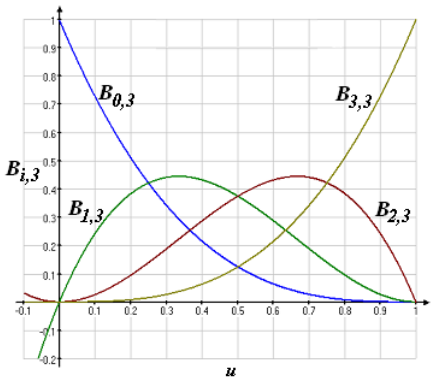


Figure 2.26: Bezier blending functions for $m = 3$

The matrix equation for a patch defined by a 4×4 array of points is

$$\mathbf{s}(u,v) = \begin{bmatrix} (1-u)^3 & 3u(1-u)^2 & 3u^2(1-u) & u^3 \end{bmatrix} \mathbf{C} \begin{bmatrix} (1-v)^3 \\ 3v(1-v)^2 \\ 3v^2(1-v) \\ v^3 \end{bmatrix} \tag{2.22}$$

where

$$\mathbf{C} = \begin{bmatrix} \mathbf{c}_{11} & \mathbf{c}_{12} & \mathbf{c}_{13} & \mathbf{c}_{14} \\ \mathbf{c}_{21} & \mathbf{c}_{22} & \mathbf{c}_{23} & \mathbf{c}_{24} \\ \mathbf{c}_{31} & \mathbf{c}_{32} & \mathbf{c}_{33} & \mathbf{c}_{34} \\ \mathbf{c}_{41} & \mathbf{c}_{42} & \mathbf{c}_{43} & \mathbf{c}_{44} \end{bmatrix} \tag{2.23}$$

The matrix $\mathbf{C}$ contains the position vectors for points that define the characteristic polyhedron and, thereby, the Bezier surface patch.

In the Bezier formulation, only the four corner points $\mathbf{c}_{11}$, $\mathbf{c}_{41}$, $\mathbf{c}_{14}$ and $\mathbf{c}_{44}$ lie on the patch. The points $\mathbf{c}_{21}$, $\mathbf{c}_{31}$, $\mathbf{c}_{12}$, $\mathbf{c}_{13}$, $\mathbf{c}_{42}$, $\mathbf{c}_{43}$, $\mathbf{c}_{24}$ and $\mathbf{c}_{34}$ control the slope of the boundary curves. The remaining four points $\mathbf{c}_{22}$, $\mathbf{c}_{32}$, $\mathbf{c}_{23}$ and $\mathbf{c}_{33}$ control the cross slopes along the boundary curves.

As Figure 2.27 shows, the Bezier surface is completely defined by a net of design points describing two families of Bezier curves on surface. Each curve is defined by a polygon of four points or vertices.

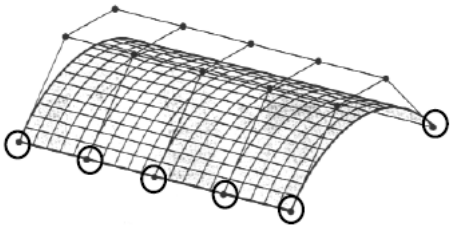


Figure 2.27: Bezier Surface

The Bezier surface has good advantages to be used in an iterative modeling environment. It is continuous and the partial derivatives of all orders exist and are continuous. The surface always lies entirely within the polyhedron convex hull and it would never oscillate wildly away from its defining control points. But, there are also drawbacks like a small change in the position of a vertex of a characteristic polyhedron tend to be strongly

propagated throughout the entire curve and the degree of the blending functions depend on the number of vertices used to specify a particular patch.

Bezier surfaces are special cases of more general B-splines surfaces, which are considered next.

2.5.3 B-Spline Surface

The B-spline surface, like the Bezier surface, is defined in terms of a characteristic polyhedron. Its formulation is

$$\mathbf{s}(u,v) = \sum_{i=0}^{m}\sum_{j=0}^{n} \mathbf{c}_{ij} N_{i,k}(u) N_{j,l}(v) \quad u,v \in [0,1] \tag{2.24}$$

The $\mathbf{c}_{ij}$ are the vertices of the defining polyhedron and they are also called control points. The $N_{i,k}(u)$ and $N_{j,l}(v)$ are the blending functions. The most important difference between Bezier and B-spline surfaces is the way the blending functions are formulated. For Bezier surfaces, the number of control points determines the degree of the blending function polynomials. For B-splines, the degree of these polynomials is specially controlled by the parameters k and l and usually it is independent of the number of control points, except as limited by Equation 2.29. The $N_{i,k}(u)$ and $N_{j,l}(v)$ functions are defined recursively, the expressions for $N_{i,k}(u)$ are introduced, $N_{j,l}(v)$ can be deduced in similar way:

$$N_{i,1}(u) = \begin{cases} 1 & if\ t_i \le u < t_{i+1} \\ 0 & otherwise \end{cases} \tag{2.25}$$

and

$$N_{i,k}(u) = \frac{(u - t_i) N_{i,k-1}(u)}{t_{i+k-1} - t_i} + \frac{(t_{i+k} - u) N_{i+1,k-1}(u)}{t_{i+k} - t_{i+1}} \tag{2.26}$$

where k controls the degree $(k-1)$ of the resulting polynomial in u and the continuity. The t_i are called knot values. They relate the parametric variable u to the $\mathbf{c}_i$ control points. For an open curve, the t_i, are

$$\begin{aligned} t_i &= 0, && if\ i < k \\ t_i &= i - k + 1, && if\ k \le i \le m \\ t_i &= n - k + 2, && if\ i > m \end{aligned} \tag{2.27}$$

with

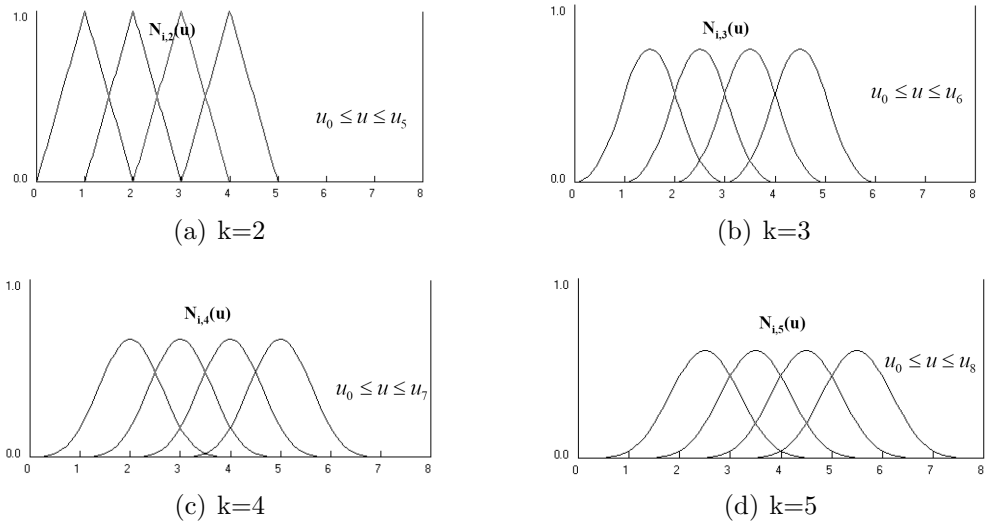
$$0 \leq i \leq m + k \tag{2.28}$$

The range of the parametric variable u is

$$0 \leq u \leq m - k + 2 \tag{2.29}$$

Since the denominators in Equation 2.26 can be zero, 0/0 is defined as 1.

Figure 2.28 shows blending-function curves for $m = 3$ and different values of k.



(a) k=2 (b) k=3

(c) k=4 (d) k=5

Figure 2.28: B-spline blending-functions for $m = 3$

A unit square on the parametric variables u and v is used to compute patches on the B-spline surface. The general matrix form of an open, periodic B-spline surface that approximates an $(m + 1) \times (n + 1)$ rectangular array of points is:

$$\begin{aligned} \mathbf{s}_{rw}(u, v) = \mathbf{U}_k \mathbf{M}_k \mathbf{C}_{kl} \mathbf{M}_l^T \mathbf{V}_l^T \quad & r \in [1 : m + 2 - k] \\ & w \in [1 : n + 2 - l] \\ & u, v \in [0, 1] \end{aligned} \tag{2.30}$$

where k and l denote the parameters that control the continuity of the surface and the degree of the blending-function polynomial; r and w identify a particular patch in the surface. The range on r and w is a function of the parameters k and l and the dimensions

of the rectangular array of control points. The $\mathbf{U}_k$ and $\mathbf{V}_l$ are

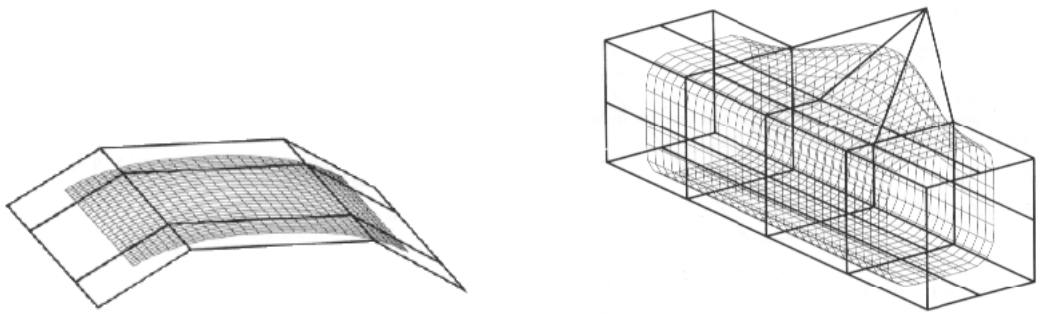
$$\mathbf{U}_k = \begin{bmatrix} u^{k-1} & u^{k-2} & ... & u & 1 \end{bmatrix} \tag{2.31}$$

$$\mathbf{V}_l = \begin{bmatrix} v^{l-1} & v^{l-2} & ... & v & 1 \end{bmatrix} \tag{2.32}$$

Elements of the $k \times l$ matrix of control points depend on the particular patch to be evaluated. Let c_{ij} denote these matrix elements, then

$$\begin{aligned} \mathbf{C}_{kl} = \mathbf{c}_{ij} \quad & i \in [r-1 : r+k-2] \\ & j \in [w-1 : w+l-2] \end{aligned} \tag{2.33}$$

Some advantages of B-spline surfaces are: their ability to preserve arbitrarily high degrees of continuity over complex surfaces; changes in the local shape of a B-spline surface are not propagated throughout the entire surface; they lie entirely within the polyhedron convex hull; and the degree of the polynomials is separated from the number of control points. These characteristics make the B-spline surfaces very attractive for use in an interactive modeling environment, as CAD systems. Figure 2.29 shows two examples of B-spline surfaces.



(a) (b)

Figure 2.29: B-spline surfaces

2.6 Smooth Approximation of a Points Cloud

Approximating a given surface (in any representation) or an unstructured cloud of points by a B-spline surface is a widely investigated problem. The main approach uses a least squares formulation with a regularization term that expresses the fairness of the

final result [Pot03, yH05].

The problem consists of looking for a B-Spline patch, which approximates an unstructured cloud of points, with a representation presented in Equation 2.24. To make the problem formulation easier, the standard B-spline notation will be rewritten as follows, [Sid01]. Given the set of $\nu = (m+1) \times (n+1)$ control points,

$$s(u,v) = \sum_{\omega=0}^{\nu} \mathbf{M}_{\omega}(u,v) c_{\omega}, \tag{2.34}$$

where $c_{\omega} = c_{ij}$ and $\mathbf{M}_{\omega}(u,v) = N_{i,k}(u) N_{j,l}(v)$, such that $i = 1 + \lfloor \omega/(n+1) \rfloor$ and $j = 1 + \omega \ \mathbf{mod}\ (n+1)$.

Let p_{τ}, $\tau = 1, ..., \mu$, be the input data points that will be approximated. The s_{τ} points should be close to the points p_{τ}. Then, it is necessary to find a set of c_{ω} that minimizes

$$F = \sum_{\tau=0}^{\mu} \| s(u_{\tau}, v_{\tau}) - p_{\tau} \|^2 = \sum_{\tau=0}^{\mu} \left[\sum_{\omega=0}^{\nu} \mathbf{M}_{\omega}(u_{\tau}, v_{\tau}) c_{\omega} - p_{\tau} \right]^2. \tag{2.35}$$

Since, $\mathbf{M}_{\omega}$ are pre-computed, the function is quadratic in the unknown control points, c_{ω}. This classical least squares fitting always has a solution, although it is not necessarily unique. Also, the resulting surface, $s(u,v)$, may not be sufficiently smooth. One may augment 2.35 with a regularization term, also called the smoothing term or penalty term, to guarantee uniqueness and control the smoothness of the solution. Commonly, this term is obtained from an approximation of the *thin-plate energy*, a quadratic function in the second partial derivatives

$$F_s = \int \int (s_{uu}^2 + 2 s_{uv}^2 + s_{vv}^2) du dv. \tag{2.36}$$

or from the *membrane energy* that is quadratic in the first partial derivatives

$$F_s = \int \int (s_u^2 + s_v^2) du dv. \tag{2.37}$$

Considering the smoothing term, the functional we want to minimize is, now

$$F = \sum_{\tau=0}^{\mu} \left[\sum_{\omega=0}^{\nu} \mathbf{M}_{\omega}(u_{\tau}, v_{\tau}) c_{\omega} - p_{\tau} \right]^2 + \lambda F_s, \tag{2.38}$$

with the smoothing parameter, $\lambda \geq 0$. Since F and any of the smoothing terms (F_s) are quadratic in the unknowns c_ω, Equation 2.38 can be written in the matrix form

$$\mathbf{F} = \|\mathbf{Bc} - \mathbf{p}\|_2^2 + \lambda \mathbf{c}^T \mathbf{E} \mathbf{c}, \tag{2.39}$$

where, $\mathbf{c} = (c_1, ..., c_\nu)^T$, $\mathbf{p} = (p_1, ..., p_\mu)^T$, $\mathbf{E}$ is a symmetric and positive definite $\nu \times \nu$ matrix and $\mathbf{B}$ is the $\mu \times \nu$ matrix:

$$\mathbf{B} = \begin{pmatrix} \mathbf{M}_1(u_1, v_1) & \dots & \mathbf{M}_\nu(u_1, v_1) \\ \mathbf{M}_1(u_2, v_2) & \dots & \mathbf{M}_\nu(u_2, v_2) \\ \vdots & \dots & \vdots \\ \mathbf{M}_1(u_\mu, v_\mu) & \dots & \mathbf{M}_\nu(u_\mu, v_\mu) \end{pmatrix} \tag{2.40}$$

Setting the gradient of F equal to zero leads to the normal equations:

$$\left(\mathbf{B}^T \mathbf{B} + \lambda \mathbf{E}\right) \mathbf{c} = \mathbf{B}^T \mathbf{p}, \tag{2.41}$$

The $n \times n$ matrix $B^T B$ is symmetric and positive semi-definite, so the solution to 2.41 with $\lambda = 0$ is not necessarily unique. As $\mathbf{E}$ is positive definite, then the system matrix $(\mathbf{B}^T \mathbf{B} + \lambda \mathbf{E})$ with $\lambda > 0$ is also positive definite, and thereby nonsingular, which implies that 2.41 has unique solution.

B-Splines surface combine efficiency with geometric flexibility, which makes them a powerful tool for describing scattered points in CAD systems. Figure 2.30 shows an example of a given noisy data points approximation.

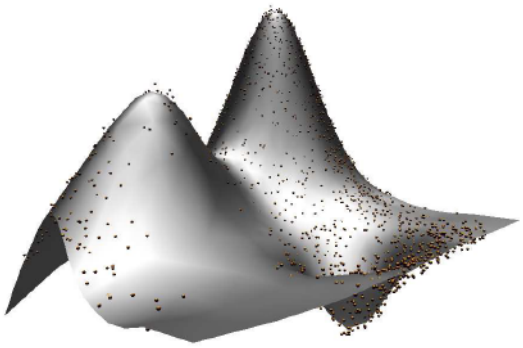


Figure 2.30: Approximated surface of a points cloud [yH05]

2.7 Discussion

The surface meshes, when generated by the sphere and swept primitives discretization methods, respect all the quality requirements, such as regularity, connectivity and have its elements internal angles relatively close to 60°. Unfortunately, the elements quality of the meshes generated by a 3D model acquisition process often can be very poor. The reconstruction algorithms do not care about the element shape quality or their connections when building the mesh. Their responsibilities are to guarantee correct geometry, topology and features.

The badly shaped elements also arises after applying the Boolean and assembly operations over 3D primitives: the resultant surface mesh frequently contains a large number of thin elements. As it was discussed in the section 2.2.4.2, the badly shaped triangles appear during the operations application, because many triangles from the intersection region are split into degenerate ones. Each triangle from one object can be intersected by more than one triangle in the other object. The small and badly shaped triangles appear when the intersection points and edges are included in the meshes of both regions and the triangulation is regenerated. The new triangles form the resulting mesh.

On the other hand, surface meshes that maximize internal angles are a necessary condition for the generation of high quality volumetric meshes. A high quality mesh results in a well conditioned finite element system that minimizes numerical errors and singularities that might otherwise arise during the electromagnetic simulation.

The vertices, edges, faces, slits and holes from the input surface mesh are part of the resulting volumetric mesh. The volumetric mesh generators can try to improve their results by inserting new points on the facets, but they must not remove any of the input geometric entities on the received surface mesh. Then, the quality of the surface mesh affects the quality of the volumetric mesh and by consequence the accuracy of the finite element simulation is also affected. In this context, improving the surface mesh quality is very important and it is the main contribution of this work to the CAD systems area.

In section 2.3.2, the PLC was presented as one kind of input for volumetric mesh generators. It is a polygonal description of the geometry, which leads to a non triangulated input for the volumetric mesh generators. The higher is the degree of freedom of the

volumetric mesh generator, the better is the quality of the resulting mesh. An easy way to increase this degree of freedom is to avoid the planar facet triangulation during the boundary evaluation process. Evaluating the intersection points and edges between polygons is more complicated, as it needs to deal with both non-convex polygons and polygons with holes. Although, working just with triangles is easier, because all the possible kinds of intersections between them can be easily identified, it, normally, generates a large quantity of sharp angles. Then, a first movement toward a high quality surface mesh is to modify the steps of primitive mesh generation and intersecting process of the boundary evaluation process (section 2.2.4). These modifications are detailed in section 4.1. However, this idea guarantees improvements only for the volumetric meshes with planar facets.

Since, in general, the models are composed by planar facets and curved facets, which are still approximated by triangular faces, other methods to improve the mesh should be investigated. These are called mesh post-processing methods and they are discussed in next chapter. Some of these methods need an approximation of the model surface during their application process. The approximation should limit the maximum deviation between the original surface mesh and the improved surface mesh. Section 2.6 introduced a technique to evaluate a smooth approximation of a points cloud, which will be used in a new approach to generate model surface approximation in chapter 4.

Chapter

3

Surface Mesh Post-Processing Methods

It is rare to obtain good meshes without some form of post-processing to improve the overall quality of the elements. There are three basic techniques of mesh improvement: smoothing, clean-up and refinement methods. Smoothing includes any method that adjust node locations while maintaining the element connectivity. Clean-up generally refers to any process that changes the element connectivity, while the refinement reduces the local element size. These techniques can be considered the basis for mesh post-processing. Most of the time, applying just one of these techniques is not enough to achieve the desirable mesh quality level. So, in order to guarantee a better resultant mesh, two or all previous techniques are combined to build hybrid methods and take advantage of each one best part.

Surazhsky and Gotsman [Sur03] classifies the surface mesh post-processing methods in another way. One group of algorithms is based on partitioning 3D meshes into patches, and treating each patch separately. While these techniques yield reasonable results, they are very sensitive to the patch structure and the vertex sample is difficult to control. Other group of algorithms is based on global parameterization of the original mesh, and then re-sampling the parameter domain. Following this, the new triangulation is projected back into 3D space, resulting in an improved version of the original model. The main drawback of global parameterization methods is the sensitivity of the result to the used

parameterization. Embedding a non trivial 3D structure in the parameter plane severely distorts the structure and important information, which is not specified explicitly, may be lost on the way. Even if the parameterization minimizes the metric distortion of the 3D original model in some reasonable sense, it is impossible to eliminate it completely. Moreover, methods finding a global parameterization are slow, because they involve the solution of a large set of equations that are sometimes nonlinear.

The main alternative to global parameterization is to work directly on the surface mesh and perform series of local mesh modifications to improve, enrich and simplify the mesh. This approach is known as the mesh adaptation process or simply *remeshing* and it is used in this work. Resmeshing algorithms can be seen as hybrid methods, since their local mesh modification encapsulates the basic techniques of smoothing, clean-up and refinement. These algorithms usually involve computationally expensive optimizations in 3D or more efficient but less accurate optimization in the tangent plane. Another difficulty of this approach is that the mesh vertices must remain on the original model surface or on an approximation of it during the adaptation process.

This chapter begins with a review of the basic mesh post-processing techniques. In the sequence, the local mesh modifications operators and two hybrid methods for remeshing are introduced.

3.1 Smoothing Methods

Most smoothing procedures involve some form of iterative process that reposition individual vertices to improve the local quality of the elements. A wide variety of smoothing techniques have been proposed. These methods can generally be classified as follows, [Owe98]:

1. Averaging methods;
2. Optimization-based methods;
3. Physically-based methods;

3.1.1 Averaging Methods

Among a variety of smoothing algorithms, the simplest and most straightforward is Laplacian smoothing [Owe98]. With this method, an internal node in the mesh is placed at the average location of any node connected to it by an edge. This is an iterative method in the sense that all the vertices in the mesh are adjusted one by one. With little modification, this technique can be applicable for any element shape. Most smoothing procedures will iterate through all the internal vertices in the mesh several times until any individual node has not moved more than a specified tolerance. Although it has its problems, it is simple to implement and it is in wide use.

Similar to Laplacian, there are a variety of other smoothing techniques, which iteratively reposition vertices based on a weighted average of the geometric properties of the surrounding vertices and elements. Averaging methods quite often also employ some form of additional constraint on the movement of a node. For example, because Laplacian smoothing alone sometimes has the tendency to invert elements or degrade the local element quality, a comparison of local element quality is made before and after the proposed move and the node is moved only if element quality is improved. This is often referred to as constrained Laplacian smoothing. This method is effective in avoiding inverted elements. The computational cost, however, is much higher; for example, if the minimum angle is used as the constraint, three angles need to be calculated for each triangular element before and after smoothing.

3.1.2 Optimization-Based Methods

Rather than relying on heuristic averaging methods, some codes use optimization techniques to improve element quality. Optimization-based smoothing methods use some mesh quality measures, as the ones presented in section 2.4.1, to define cost functions. Mesh vertices are moved so that the cost function is minimized or maximized.

One advantage of optimization-based smoothing is that it can guarantee the improvement of mesh quality. By optimizing the quality measures, severely distorted elements are effectively eliminated. The computational cost, however, is much higher than Laplacian smoothing. For a two dimensional triangular mesh, for example, optimization-based

smoothing method can be 30 to 40 times slower than Laplacian smoothing.

Combining Laplacian/optimization-based approach is recommended. What is generally advocated is that Laplacian smoothing is done for the majority of the time, reverting to optimization based smoothing only when local element shape metrics drop below a certain threshold.

3.1.3 Physically-Based Methods

Another important area of mesh improvement includes methods that reposition the vertices based on a simulated physically based attraction or repulsion force. The force between neighboring vertices as a system of springs interacting with each other can be simulated. Or, the vertices can be viewed as the center of bubbles that are repositioned to attain equilibrium. With changes in the magnitude and direction of inter-particle forces, different element sizes can be achieved.

3.2 Clean-up Methods

Like smoothing, there are a wide variety of methods currently employed to improve the quality of the mesh by making local changes to the element connectivities. Clean-up methods generally apply some criteria that must be met in order to perform a local operation. The criteria in general can be classified as a shape improvement or a topological improvement. In addition, clean-up operations are generally not done alone, but are used in conjunction with smoothing.

3.2.1 Shape Improvement

For triangle meshes, simple diagonal swaps are often performed. For each interior edge in the triangulation a check can be made to determine at what position the edge would effectively improve the overall or minimum shape metric of its two adjacent triangles.

In some cases, particularly with curved surfaces, the elements resulting from the mesh generator may deviate significantly from the underlying geometry. Edge swaps can be performed based on which local position of the edge will deviate least from the surface.

3.2.2 Topological Improvement

A common method for improving meshes is to attempt to optimize the number of edges sharing a single node. This is sometimes referred to as node valence or degree. In doing so, it is assumed that the local element shapes will improve. For a triangle mesh there should optimally be 6 edges at a node, which maximizes the chance of having angles close to 60°. Whenever there is a node that does not have an ideal valence, the quality of the elements surrounding it will also be less than optimal. Performing local transformations to the elements can improve topology and hence element quality.

3.3 Refinement Methods

Element refinement procedures are numerous. For our purposes, refinement is defined as any operation performed on the mesh that effectively reduces the local element size. The reduction in size may be required in order to capture a local physical phenomenon, or it may be done simply to improve the local element quality. Starting with a coarse mesh, a refinement procedure can be applied until the desired nodal density has been achieved. Quite frequently, refinement algorithms are used as part of an adaptive solution process, where the results from a previous solution provide criteria for mesh refinement.

Although there are certainly more methods defined, two of the principal methods for triangle refinement are the edge bisection and the point insertion.

3.3.1 Edge Bisection

Edge bisection involves splitting individual edges in the triangulation. As a result, the two triangles adjacent the edge are split into two. An example of edge bisection use is the backward longest-edge refinement method [Riv97], shown in Figure 3.1. It exploits the knowledge of the reference triangulation to work locally in the refinement area and some neighboring triangles. The new points introduced in the mesh are mid-points of the longest edge of, at least, one triangle of the reference mesh. In order to maintain a conforming triangulation, the local refinement of a given triangle involves the refinement of the triangle itself and the refinement of some of its neighbors.

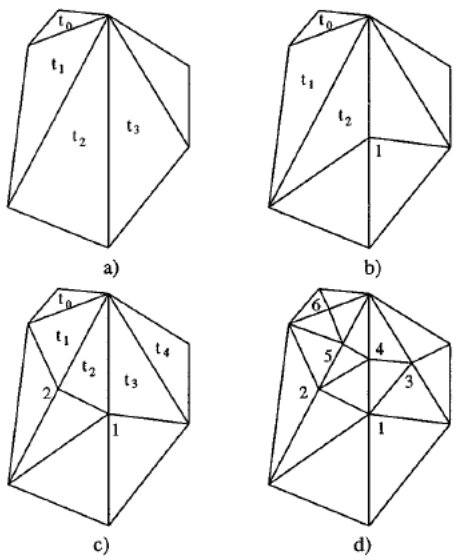


Figure 3.1: Backward longest-edge bisection of triangle t_0: (a) initial triangulation; (b) First step of the process; (c) second step in the process; (d) final triangulation [Riv97]

3.3.2 Point Insertion

A simple approach to refinement is to insert a single node at the centroid of an existing element, dividing the triangle into three. This method does not generally provide good quality elements, particularly after several iterations of the scheme. To improve the scheme, a Delaunay approach can be applied to connect the new node to the existent triangulation, while the Delaunay criterion is maintained, as Figure 3.2 shows.

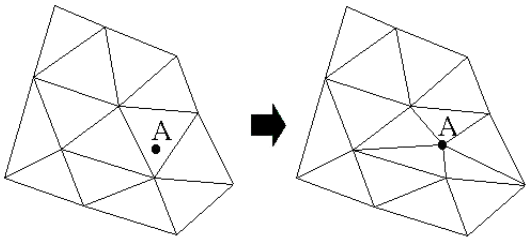


Figure 3.2: Example of Delaunay refinement, where point A is inserted [Owe98]

3.4 Local Mesh Modification Operators

The basic techniques of refining, smoothing and cleaning-up are usually encapsulated by local mesh modification operators. The commonly used operators are edge-swapping, edge-collapsing, edge-splitting and vertex relocation [Fre00, Sur03].

More precisely, the edge-collapsing operation consists of identifying the two endpoints of an edge AB into a single vertex, as illustrated in Figure 3.3. This operation is executed if the new faces do not deviate from the model surface approximation by more than the permitted deviation.

(a) Initial configuration (b) New configuration

Figure 3.3: Edge-collapsing operator application

For a mesh edge AB (Figure 3.4(a)), the edge-splitting operation consists of introducing the edge midpoint M and in snapping M onto the model surface approximation (Figure 3.4(b)). This operation is applied if the edge deviation is higher than the permitted one or to improve the elements shape. The elements that share the edge are replaced by four new ones.

(a) Initial configuration (b) New configuration

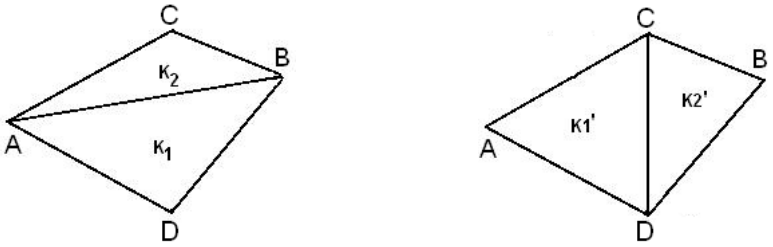
Figure 3.4: Edge-splitting operator application

The vertex relocation procedure consists of redefining all the elements sharing a given vertex A, as Figure 3.5 presents. At first, for each element that shares A, an optimal location for A is evaluated. Then, the average of the optimal locations is projected onto the model surface approximation to find A'.

The edge-swapping operator changes the connections of the vertices A, B, C and D by removing the edge AB and inserting a new edge CD, like the Figure 3.6 shows. This operation is performed only if the two triangles sharing the edge are coplanar or almost coplanar and the new configuration is better than the previous one. It introduces local changes in the surface curvature along the edge.

(a) Initial configuration (b) New configuration

Figure 3.5: Vertex relocation operator application



(a) Initial configuration (b) New configuration

Figure 3.6: Edge-swapping operator application

The deviation between the mesh elements and the geometric approximation of the model surface is controlled during any operator application. A local mesh modification is performed only if the model surface approximation is preserved and the mesh quality is increased.

The four local mesh modification operators presented here are applied during the optimization process of the next two remeshing methods and also in our proposed approach to improve surface meshes quality.

3.5 Hybrid Methods

The post-processing techniques previously presented in this chapter can either refine, smooth or clean-up the surface mesh. When they are used by themselves, they might not produce surface meshes good enough for electromagnetic simulation through finite element method. However, the combination of them can get the best of each one and improve the mesh quality to the necessary quality level discussed in section 2.4.

The mesh adaptation process applies series of local mesh modifications to optimize a surface mesh. Local adaptation is necessary to achieve accurate solutions with an acceptable effort in terms of simulation time and memory consumption. The local refine-

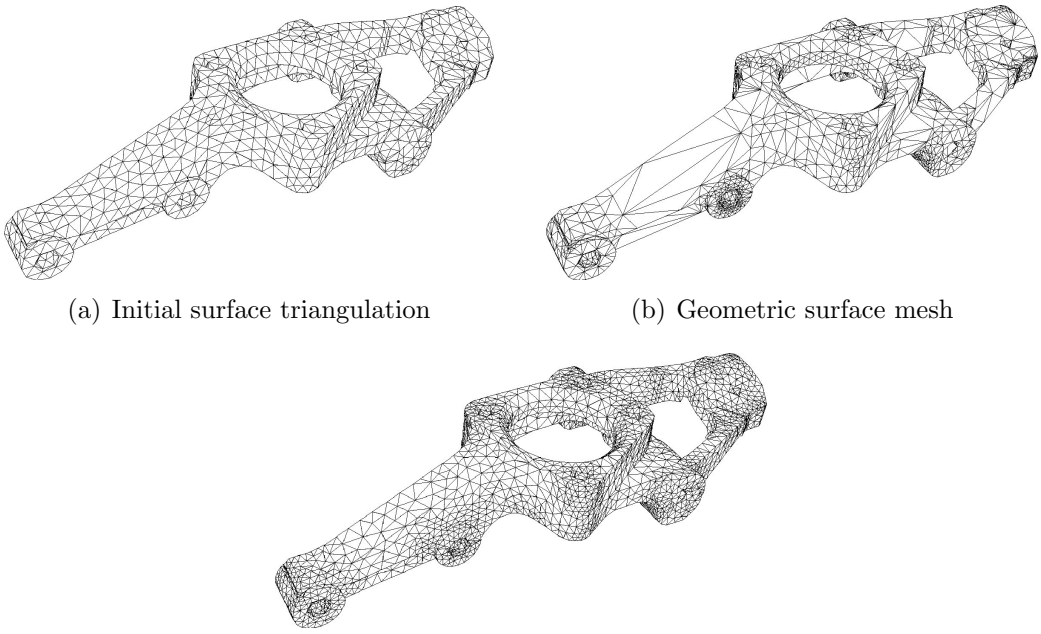
ment, coarsening, or smoothing steps are performed to enhance purely geometrical quality aspects or can be guided by a control function. In the first case, the refined regions concentrate around areas with small local feature size[i], i.e, areas with high curvature. Large elements which resolve small geometrical features are usually badly shaped and require refinement. In the latter case the mesh density is adapted to a stationary solution or dynamically for each time step of a transient simulation. The regions of refinement have to migrate as the characteristics of the transient solution change over the domain. Essentially, local refining in some regions as well as local coarsening in other regions becomes necessary to avoid meshing the entire domain repeatedly.

The adaptive process is used as an alternative to global parametrization methods and it is proving to produce very attractive results. The remeshing methods we will discuss in sections 3.5.1 and 3.5.2 are based on it.

3.5.1 Frey's Method

Frey [Fre98, Fre00] introduced a scheme suitable to obtain geometric as well as finite element mesh given an initial surface triangulation $\mathcal{T}$. The whole process involves three main stages. At first, the triangulation $\mathcal{T}$ is analyzed in order to construct a geometric mesh $\mathcal{M}_{\mathcal{G}}$ in which the distance from the initial approximation is bounded. This mesh can be considered as an accurate piecewise linear approximation of the underlying surface, although it may not be suitable for numerical computations. Then, depending on the context of application, the mesh $\mathcal{M}_{\mathcal{G}}$ is optimized so as to improve the element shape quality and/or to enforce prescribed elements sizes. To this end, a G^1 continuous geometric support is defined on $\mathcal{M}_{\mathcal{G}}$, that will be used to construct a geometric metric $\mathcal{G}$ (defined in the tangent planes) as well as to govern the mesh modification operations. Finally, the mesh $\mathcal{M}_{\mathcal{G}}$ is locally optimized to obtain a unit surface mesh $\mathcal{M}$ conforming the metric $\mathcal{G}$. Figure 3.7 illustrates the stages involved in the creation of a finite element mesh of a mechanical device, following these stages are detailed.

[i]The local feature size (lfs) at a point $p \in S$, where S is a surface, is the distance from p to the nearest point of the medial axis of S. The medial axis of a closed subset $C \in \mathbb{R}^3$ is the subset of $\mathbb{R}^3 - \{C\}$ that consists of all points in $\mathbb{R}^3 - \{C\}$ having two or more nearest points in C.



(a) Initial surface triangulation

(b) Geometric surface mesh

(c) Resulting computational mesh

Figure 3.7: Three stages in the construction of a finite element mesh. [Fre00]

3.5.1.1 Geometric Surface Mesh

The aim of this stage is to transform the initial triangulation into a geometric surface mesh containing less elements, in accordance to the geometric requirements. A preprocessing stage consists in identifying the C^1 discontinuities of the model (corners, sharp edges, etc.) and in computing the normals at the mesh vertices. This is necessary because realistic meshes involves ridges, corners and curves traced onto the surface.

The suggested approach guarantees that the elements of the simplified mesh stay within a given (user-specified) tolerance of the original surface triangulation. The tolerance region can be seen as an *envelope* that surrounds the initial triangulation, in which the simplification is performed. Usually, a tolerance value is considered to bound the maximum allowable deviation of the simplified mesh from the initial one. The goal is to compute a piecewise linear approximation $\mathcal{M}_\mathcal{G}$ of the initial triangulation $\mathcal{T}$ such that every point of $\mathcal{T}$ is within a distance δ of a point of $\mathcal{M}_\mathcal{G}$ and conversely, that every point of $\mathcal{M}_\mathcal{G}$ is within a distance δ of some point of $\mathcal{T}$.

The extraction of a geometric surface mesh is based, mainly, on edge collapsing and edge swapping operations. In addition, a vertex relocation operation can also be applied after the edge collapsing stage in order to improve the element shape quality. The geom-

etry of the surface around P is locally approached by a quadric surface and the optimal point is projected onto this surface.

In order to control the quality degradation during the collapse operation, a triangle shape quality measure is used:

$$Q(K) = \alpha \frac{h_{max}}{\rho}, \tag{3.1}$$

where ρ denotes the inradius of a triangle K, h_{max} is the largest edge length of K and α is a normalization coefficient so that $Q(K) = 1$ for an equilateral triangle and $Q(K) = \infty$ for triangles with null area.

The surface roughness is controlled checking if a triangle K, conforms the following measure:

$$\min_j(\vec{n}(K(P_j)),\ \vec{n}(K)) \leq \cos\theta, \tag{3.2}$$

where $\vec{n}(K(P_j))$ denotes the surface normal at a vertex P_j of K; $\vec{n}(K)$ is the triangle K normal; and $\cos\theta$ is a given tolerance value (in practice, $\theta \approx 45°$).

Schematically, the geometric simplification algorithm can be written as follows:

Initialization:
  $\mathcal{M}_\mathcal{G} = \mathcal{T}$;
  $\delta_i = 0$;
while $\delta_i < \delta$
  collapse edges of $\mathcal{M}_\mathcal{G}$ if geometry and mesh quality are preserved;
  optimize mesh quality (edge swapping);
  optimize the element shape qualities using vertex relocation;
  evaluate δ_i;
end while

3.5.1.2 The Geometric Support

The geometric mesh $\mathcal{M}_\mathcal{G}$, evaluated in the previous subsection, is used to construct the geometric support, which represents the analytical definition of a surface and can be used to emulate the features of a geometric modeling system. If the initial triangulation is an accurate approximation of the surface, the geometric support simulates reasonably well a geometric modeling system.

A geometric size map, denoted as $\mathcal{G}$, is constructed and associated with the surface points, such that, for a mesh satisfying this map, the gap value between the triangles and the surface is controlled. Practically, this map $\mathcal{G}$ prescribes, at any vertex, a size proportional to the minimum of the principal radius of curvature. The principal curvatures at a point P of a C^2 surface are computed numerically based on the given surface triangulation. To this end, the underlying surface geometry is locally approached by a quadric surface.

The construction of the geometric support involves the definition of a piecewise planar surface of order G^1 based on the geometric mesh previously extracted. Each triangle represents a patch, two adjacent patches must have the same tangent plane along their common boundary, if it is not a ridge. Schematically, the support consists in a network of patch boundary curves and their transverse planes, where the normals to the surface at the vertices are simply interpolated. Each curve and the related tangent planes are completely defined from the normals at its endpoints.

Given a mesh vertex, the geometric support is used to supply the location of the closest point onto the surface from the point. Moreover, at a G^1-continuous point, the surface normal and the principal radius of curvature can be returned by the support. For a point located along a ridge, the tangent to the curve at the point is returned.

3.5.1.3 Unit Surface Mesh Construction

The final stage involves the construction of a *unit mesh* via local geometric and topological mesh modifications. The geometric mesh, $\mathcal{M}_{\mathcal{G}}$, is modified to create a unit mesh that is normalized conforming to the evaluated geometric support, $\mathcal{G}$. Usually, the unit mesh contain well-shaped element and it will be the input for numerical computations.

The optimization of the mesh is based on local modification operators, such as edge-collapsing, edge-splitting, vertex relocation and edge-swapping. A local mesh modification is performed only when the geometric approximation is preserved and the mesh quality is not degraded. Practically, the optimization procedure consists in analyzing the current mesh edges in order to collapse the short edges and to split the large ones.

The optimization procedure modifies iteratively the current mesh, in order to adapt the element sizes to the prescribed size map. l_{AB} is the normalized length of the edge

AB. It is evaluated with respect to $\mathcal{G}$[Fre00]. The main steps of this approach can be summarized as follows:

Preliminary definitions:
construction of the geometric support, $\mathcal{G}$,
$\mathcal{M} = \mathcal{M}_{\mathcal{G}}$;
while $\mathcal{M}$ is modified
compute l_{AB} considering $\mathcal{G}$
if $l_{AB} < \frac{1}{\sqrt{2}}$
collapse AB
else if $l_{AB} > \frac{1}{\sqrt{2}}$
split AB
end while
optimize mesh quality (edge swapping and node relocation)

A mesh satisfying the rectified geometric size map is a geometric mesh in which the element are all well-shaped.

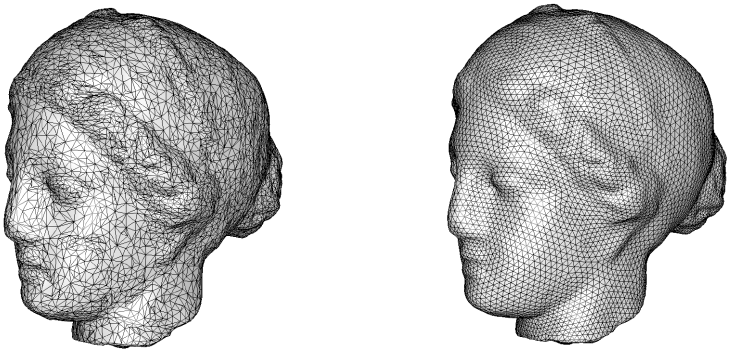
Frey's method is an important reference in remeshing area. It is a general scheme that works directly on the mesh to construct high quality geometric surface meshes from a given surface triangulation. The geometric information is guaranteed by locally approximating the input data by quadric surface patches. This approach proved to provide good results when the input is a large data set either produced as piecewhise linear approximation of curved parametric surfaces or obtained from surface reconstruction algorithms.

3.5.2 Surazhsky and Gotsman's Method

Surazhsky and Gostman [Sur03] proposed a remeshing scheme based on idea of improving mesh quality by a series of local modifications of the mesh geometry and connectivity. Their contribution was the area based smoothing technique, which allows the control of both triangle quality and vertex sampling over the mesh, as a function of some criteria, e.g. the mesh curvature. To perform local modifications on meshes of arbitrary *genus*[ii], a dynamic patch-wise parametrization is used. The parametrization is constructed on-the-fly as the algorithm progress with local updates. As a post-processing stage, a new

[ii]Genus is a topologically invariant property of a surface defined as the largest number of non-intersecting simple closed curves that can be drawn on the surface without separating it. Roughly speaking, it is the number of "holes" in a surface.

algorithm to improve the regularity of the mesh connectivity is introduced. The algorithm is able to create an unstructured mesh with a very small number of irregular vertices, as Figure 3.8 shows.



(a) Original (8.268 vertices)

(b) Remeshed (9.240 vertices)

Figure 3.8: A remeshing example for the Venus model [Sur03]

The input of this scheme is a 2-manifold[iii] (except at the boundaries) 3D mesh $\mathcal{M}_0$ with arbitrary genus and possible holes. It is considered to be a piecewise linear approximation of a smooth surface, which is C^1-continuous except at the boundaries and a set of curves specified by feature edges. These feature edges can be provided by the user or computed automatically as edges whose dihedral angle is less than some threshold angle.

3.5.2.1 Geometric Background

An estimative of the model surface in the vicinity of a mesh triangle is obtained by an approximation using triangular cubic Bézier patches. Vlachos et al. [Vla01] presented a simple and efficient yet robust and accurate method to construct such curved patches called *PN triangles*. The triangle vertex normals together with vertex coordinates are used to construct a PN triangle. PN triangles usually maintain a G^1-continuous surface along adjacent triangles when their common vertices have identical normals. The normal of any point within a PN triangle is defined as an efficient quadratic interpolation of the normals at the triangle vertices.

Two error measures are used to evaluate the distance between the two meshes and to ensure fidelity of the new mesh to the original mesh geometry. Let $f = (v_1, v_2, v_3)$ be a face whose error is to be estimated. The first measure E_{smth} captures the degree of

[iii] Closed surface

smoothness and should not exceed some threshold angle θ_{smth}:

$$E_{smth}(f) = \max_{i \in \{1,2,3\}} \langle N_f, Nv_i \rangle < \cos \theta_{smth}. \tag{3.3}$$

N_f and N_v are unit normals of f and its vertex v, respectively; $\langle ., . \rangle$ denotes the dot product. N_v is taken from the original surface. Intuitively, E_{smth} describes how well f coincides with tangent planes of the surface at the vertices of f. The second measure E_{dist} captures the gap between f and the surface:

$$E_{dist}(f) = \max_{i \in \{1,2,3\}} \langle N_{v_i}, Nv_{i+1} \rangle < \cos \theta_{dist}. \tag{3.4}$$

Vertex index are modulo 3; θ_{dist} is a threshold angle. A greater value of the maximum angle between the normals of two face vertices corresponds to a more curved surface above face f, and thus, to a bigger distance.

3.5.2.2 Remeshing

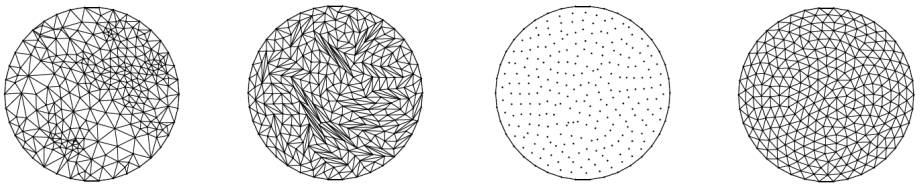
The focus of this remeshing scheme is on maximizing the angles of all triangles of the mesh. Remeshing of the given mesh $\mathcal{M}_0$ is performed by applying series of local modifications (edge-flip, edge-collapse, edge-split or vertex relocation). The modifications are applied sequentially in order to achieve desirable mesh characteristics. These local modifications are applied on the new mesh $\mathcal{M}$, while the original mesh $\mathcal{M}_0$ provides a reference to the original mesh geometry. Before starting $\mathcal{M}$ is initialized to $\mathcal{M}_0$. To ensure fidelity, a modification is applied only if all faces created or affected by the modification satisfy the error conditions.

The main stages of the remeshing scheme are as follows:

1. Adjust the number of vertices of $\mathcal{M}$;
2. Apply the area-based remeshing procedure on $\mathcal{M}$;
3. Regularize $\mathcal{M}$ connectivity;
4. Apply the angle-based smoothing procedure on $\mathcal{M}$.

Edge-collapse and edge-split are used to change the number of mesh vertices. Edge-flip and vertex relocation improve the quality of the mesh triangles. The area-based remeshing procedure is the heart of the scheme. Another two stages improves the regularity of the mesh connectivity leaving only a small number of irregular vertices. The angle-based smoothing then polishes the mesh to obtain the optimal mesh geometry without changing its connectivity.

The area equalization is done iteratively by relocating every vertex such that the areas of the triangles incident on the vertex are as equal as possible. Triangle area is usually used to assist, analyze or control meshing and it has not been used as central factor in mesh generation. The reason for this is that by using triangle areas alone, meshes of reasonable quality can not be obtained. A mesh optimization that equalizes the areas of the mesh triangles or brings triangle areas to specified (absolute or relative) values will, in most cases, result in many long skinny triangles. Nevertheless, Surazhsky and Gotsman discovered that a 2D triangulation having triangles with equal (or close to equal) areas has globally uniform spatial vertex sampling; see Figure 3.9. Then, they presented the following remeshing scheme that exploits this: alternate between area equalization and a series of angle-improving edge-flips. Applying this simple scheme results in a 2D mesh with a very uniform and well-shaped triangles, Figure 3.9(c). Unfortunately, this process does not usually converge. After a uniform sampling rate is obtained, the process oscillates, producing different but similar uniform vertex distributions. However, the subsequent steps of this remeshing scheme improve the mesh quality further by regularizing and smoothing it.



(a) Original Mesh (b) The triangles area are equalized (c) Discarding the edges (d) After edge-flips, the mesh results in a close regular mesh

Figure 3.9: Area-based remeshing [Sur03]

Another component of the remeshing scheme is an effective, simple and efficient algorithm to improve the mesh quality by regularizing its connectivity. The algorithm also

performs series of local operations that modify the connectivity (edge-flips, edge-collapses and edge splits). Improving the regularity means minimizing the following function:

$$\mathcal{R}(\mathcal{M}) = \sum_{v \in \mathcal{M}} (d(v) - d_{opt}(v))^2, \tag{3.5}$$

where $d(v)$ is the degree of vertex v and $d_{opt}(v)$ its optimal degree. Vertices on the boundary, but not in the corners, have $d_{opt} = 4$, and for inner vertices $d_{opt} = 6$. These values increases the possiblity of having 60°internal angles.

To regularize $\mathcal{M}$ connectivity, all edges that need modification are stored on a priority queue. The edges are classified and sorted following this classification. The algorithm processes the first edge in the queue until the queue is empty. After each modification the queue is updated.

Surazhsky and Gotsman introduced an efficient and robust remeshing scheme. Their scheme is able to improve the quality of large triangular meshes, which represents a piecewise approximation of smooth surfaces. During the entire mesh adaptation process, the mesh is compared to the approximation of the model surface to assure fidelity to the original mesh. The model approximation is a set of triangular cubic Bezier patches, where each face of $\mathcal{M}_0$ is approximated by one patch.

3.6 Discussion

The basic methods do not achieve good results in mesh improvements when applied isolated. The smoothing methods are good when the distribution are not too degraded, because they do not change the elements connectivity. None vertex is inserted or removed by this method, the mesh vertices are just relocated to improve elements quality. The clean-up methods are able to improve element connectivities, but usually are used in conjunction with smoothing. The refinement methods, frequently, are used as part of an adaptive solution process.

The hybrid methods like Frey´s and Surazhsky and Gotsman´s combine the basic methods to guarantee a better resultant mesh. Frey's method applies series of local mesh modification to improve the mesh in an adaptive process. It approximates the model surface by a quadric surface to guarantee the model geometric characteristics. Firstly, the

geometric surface mesh is extracted from an initial triangulation with bounded distance. Then, a geometric support is built to govern the mesh modification operations during the construction of the normalized unit mesh. This method input is considered to be a large triangular mesh and the method to evaluate the approximation is the same for the entire model.

Surarzhsky and Gotsman's method is based on Frey's work with some strategies alterations. It also use local mesh modification operators to improve connectivity and element shape. A large triangular mesh that represents a C^1-continuous surface is the input and it is approximated by a set of triangular cubic Bézier patches. The mesh adaptation process alternates the application of operators for area equalization and angle improvements. The regularization of the connectivity is realized at the end.

Both methods are applied only for large triangular meshes representing smooth surfaces. They do not consider if their works are able to improve meshes with low element density or meshes of models composed by curved and planar areas. All the faces are always approximated in the same way, quadric surface approximation for Frey´s method and triangular cubic Bézier patches for Surazhsky and Gotsman´s method.

Solid models can be represented by a scarce data set, common in models generated by solid modelers, or a dense data set, as normally the algorithms for surface reconstruction produce. They also can have planar facets that need to remain as so when the mesh quality is being improved. In order to address these cases, next chapter introduces a new approach to generate the surface approximation of arbitrary models. In addition, a remeshing scheme driven by the model surface approximation and angle improvements was developed to improve the mesh elements quality of general CAD models.

Chapter

4

Remeshing Driven by Smooth Approximation of Model Surface

When a mesh of simplicial elements (triangles or tetrahedra) is used to form a piecewise linear approximation of a function, the accuracy of the approximation depends on the size and shape of its elements. This is also true for the stiffness matrices conditioning in finite element methods (section 2.4).

Automatic mesh generators can produce surface meshes with a specified quality degree at the beginning. But, after some Boolean and assembly operations application, the quality can decrease drastically (section 2.2.4). This problem also arises in models generated by a 3D model acquisition process (section 2.3.3.3). The reconstruction algorithms do not care about the element shape quality or their nodes interconnections. They only intend to guarantee the topological and geometric characteristics correctness.

Although the volumetric mesh generators are normally able to refine the surface mesh by inserting new vertices, they are not allowed to remove any pre-existent elements to achieve the necessary quality degree for the finite element analysis (section 2.3.2). Then, the surface mesh quality directly affects the finite element volumetric mesh quality generated from this surface mesh. If the surface mesh quality is poor the volumetric mesh can not be generated or a poor quality volumetric mesh is obtained, which compromises

the accuracy and cost of a finite element analysis.

To solve the quality lack problem in surface meshes, many algorithms were developed in the last decade. Last chapter presented an overview on surface post-processing methods discussing their concepts, characteristics and drawbacks. The basic techniques can not reach the desired surface mesh quality level when applied separately. The methods based on global parameterization of the original mesh are slow and very sensitive to the parameterization. Then, the main alternative are the methods that work directly on the mesh to reduce the number of sharp angles, improve the nodes distribution and their interconnections. These approaches apply series of local mesh modifications operators, which encapsulates the smoothing, cleaning-up and refinement techniques. They are called mesh adaptation processes.

The two adaptive approaches that were presented in chapter 3 are very good to improve the surface mesh quality when the model surface is smooth and represented by oversampled meshes, like in the models obtained by reconstruction algorithms. In this case, the approaches are able to improve the mesh without loosing geometric and topological properties. However, models generated by Boolean or assembly operations applications over predefined primitives have normally restricted number of elements and planar facets; simplifications in these models might generate big geometric distortions. Since this work main goal is to address both cases indistinctly, we need an approach that guarantees improvements on mesh quality independent of how its surface mesh was generated.

Then, we propose the combination of two methods to improve the surface mesh quality: i) as a pre-processing phase, we present modifications in the boundary evaluation process of the Boolean and assembly operations to avoid the planar facets triangulation, allowing the surface mesh to be a polygonal mesh [Nun05]; and ii) a remeshing method driven by smooth surface approximation of mesh nodes [Nun06a, Nun06b, Nun07]. The first technique increases the volumetric mesh generator degree of freedom to obtain a higher mesh quality. It suppress some edges that do not add any geometric information to the model, like the edges shared by planar faces. The second method improves the intrinsic properties of the mesh elements by applying series of local mesh modification operators. During the process, the nodes movements are driven by the model surface approximation, which guarantees the model geometric characteristics preservation.

Next section explains the first approach to improve the quality of models with planar facets. The following section presents our remeshing scheme and the new method to approximate the model surface. The implementation characteristics of both methods are discussed in section 4.3.

4.1 Avoiding the Planar Facets Triangulation

The triangulation of the surface mesh a priori makes the volumetric mesh generation process more difficult. When the volumetric mesh generator has more freedom to triangulate the facets, the overall quality of the volumetric mesh can be improved with fewer vertices addition.

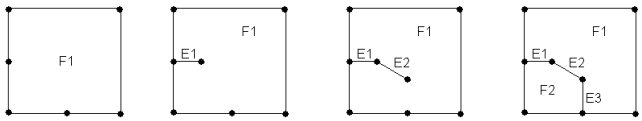
As it was presented in section 2.3.2, the volumetric mesh generator does not need a triangulated surface mesh as input. With a goal to increase the degrees of freedom available to the volumetric mesh generator, as well as improve the mesh quality, an approach that avoids the triangulation of the planar facets is introduced. However, because of the planar representation restriction, 3D curved solids, such as spheres and torus, are still represented by triangulated surface meshes. Such approach leads to an hybrid system in which there is a polygonal representation for the planar facets and triangulated representations for the curved ones.

Avoiding the triangulation of the planar facets [Nun05] means working with polygons instead of just triangles during the Boolean and assembly operations application. This is somewhat more challenging than the method explained in section 2.2.4.

The steps of the boundary evaluation: i) mesh generation over primitives; ii) intersecting process; iii) elements classification; and iv) Boolean evaluation and elimination of all undesired elements, remain conceptually the same. The modifications start in the mesh generation step by removing the triangulation of the polygonal facets. As a result, the following steps have to deal with any kind of polygons. Working just with triangles is easier, because all the possible kinds of intersections between them could be easily identified, as shown in Figure 2.8. Evaluating the intersection points and edges between polygons, on the other hand, is more complicated, as it needs to deal with both convex and non-convex polygons and also polygons with holes.

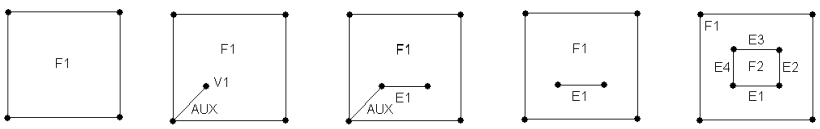
After the intersection points evaluation, the way that these points are inserted in the data structure was changed. The intersection points are still divided into two groups, like presented in section 2.2.4.2: i) points located on facet's boundary and ii) points and edges located inside the facets. The first set is treated the same way as before and a new function was developed to insert the second group of elements into the planar facets. The intersection points and edges should be included into the B-rep data structure in a way to guarantee the correct facet splits and the generation of holes and new facets, while maintaining the facets integrity and compatibility.

The process of including the intersection edges and points is incremental [Nun05]. To explain it, two examples were chosen and the steps to insert the edges are shown in Figs. 4.1 and 4.2. In Fig. 4.1, the edges to be inserted split the facet in two and in Fig. 4.2, they create a smaller facet inside a bigger one. The process starts inserting an edge that has one vertex on the facet's boundary, Fig. 4.1(b). The other edges are inserted following the connection sequence, as shown in Fig. 4.1(c). The insertion of an edge, which has both vertices already in the facet, causes the creation of a new facet, Fig. 4.1(d). In Fig. 4.2, an auxiliary edge is created first (Fig. 4.2(b)), because no edge has one vertex on the facet's boundary. After the first edge insertion (Fig. 4.2(c)), the auxiliary edge is removed (Fig. 4.2(d)) and the other steps occur in the same manner as in the previous example. The data structure modification is realized by the application of the Euler Operators [Mag00a, Nun02]. It is important to remember that one face from one shell can be split by more than one face in the other shell. Then, the whole set of intersection edges to be include in one face is processed as explained before.



(a) Start (b) Step 1 (c) Step 2 (d) End

Figure 4.1: Steps to split a facet



(a) Start (b) Step 1 (c) Step 2 (d) Step 3 (e) End

Figure 4.2: Steps to generate a smaller facet inside a bigger one

The step of elements classification also was modified. The vertices and edges classification remains the same, as explained in section 2.2.4.3, but the face classification rules were updated. Table 2.3 cannot be applied to polygons, instead of it the following rules are applied to the face edges:

- if all edges are *in*, the face is classified as *in*;
- if all edges are *out*, the face is classified as *out*;
- if most of the edges are *on* and at least one is *in*, the face is classified as *in*;
- if most of the edges are *on* and at least one is *out*, the face is classified as *out*;
- if all edges are *on*, the middle point needs to be investigated to conclude if the face classification is: *in*, *on shared* or *on anti-shared*. After the intersecting process, each face from shell A that is *on* shell B has an equal face, in terms of its coordinates, in shell B; this two faces are compared to evaluate if the faces share the same loop orientation or not;
- an edge set of a face cannot have edges classified as *in* and *out*. This means error in the intersecting process or in the edges classification step.

The Boolean evaluation and elimination of all undesired elements step did not suffer any modification.

The improvements of avoiding the planar facets triangulation are very important, but they are not suitable to eliminate all ill-shaped elements generated by the Boolean and assembly operation application and it does not affect the models generated by acquisition process. A more general method must be applied in order to guarantee better quality to any surface mesh. This method is detailed in next section.

4.2 Remeshing

The mesh adaptation process turns out to be a very interesting approach. This method is very used in the literature and it presents good results for the overall mesh quality.

This work proposes an approach based on the Frey [Fre00, Fre98] and Surazhsky and Gostman [Sur03] to improve the surface meshes. Our approach should optimize the surface meshes generated by the application of the Boolean and assembly operations over predefined primitives, as well as the meshes of reconstructed models. It should guarantee high quality surface meshes with few or none sharp angles; good vertex sampling; and good underlying surface approximation.

The optimization of the mesh is based on local modification operators that are able to simplify, enrich or locally improve the mesh. To avoid losing model geometric characteristics, it is necessary to know the model surface geometry.

Unfortunately, for the models that we want to optimize, only the mesh configuration is available. To overcome the lack of geometric information, an approximation of the model surface geometry is necessary. The surface approximation would drive the nodes movements and assure that they stay on top of the original model surface during the application of the local mesh modifications. This work introduces a smooth surface approximation evaluated by pieces from the mesh nodes to approximate the model surface geometry. Each mesh face will be approximated by a surface patch.

Our local mesh modification operators are *edge-swapping*, *edge-collapsing*, *edge-splitting* and *vertex relocation*, already explained in section 3.4. They are applied sequentially in order to achieve the desirable mesh characteristics. Edge-collapsing ans edge-splitting operations are used to improve the element shapes and also to control the sampling rate, which typically varies according to the curvature. More curved regions will contain small elements and dense vertex sampling, while almost flat regions will have large elements with more sparse vertices. The model surface approximation provides the curvature information. The edge swapping and vertices relocations will improve only the elements shape quality. The quality of the geometry approximation and the elements shape quality of the new configurations are measured *a priori*. Operations that degrade the minimum angle quality or the geometric approximation higher than a specified limit are not applied. The application of the local mesh modification results in new elements configurations.

The new approach for the model surface approximation is presented in the following section. Section 4.2.2 presents our remeshing algorithm, its detailed explanation and its computational cost.

4.2.1 Approximation of the Model Surface

The model surface approximation is very important during the remeshing process. It is through the approximation that the geometric characteristics of the model will be known and consequently preserved. A local mesh modification operator is applied only if the resultant mesh remains withing a certain tolerance of the model surface approximation and the element shape quality is not degraded.

Our model surface approximation should be able to represent models generated by the Boolean and assembly application over predefined primitives, as well as models obtained from an acquisition process. By definition, the reconstructed models are obtained from a set of unorganized sample points P, usually dense, drawn from a smooth surface S (section 2.3.3.3). On the other hand, the resultant meshes of the application of the Boolean and assembly operations can be formed by curved and planar facets. Its set of mesh vertices is normally scarce. In order to address both cases, we introduce an approach that approximates the model surface by a set of surface patches. Each mesh face has a patch associated to it. Approximating the model geometry by pieces decreases the processing time and approximation errors, because the global parameterization is avoided.

To evaluate a mesh face approximation, the points to be used are the vertices of the face we want to approximate and the vertices that surround it. For example, if the shaded face in Figure 4.4 will be approximated by a patch, the vertices: P_{32}, P_{33}, P_{22} and also the vertices: P_{11}, P_{12}, P_{13}, P_{14}, P_{21}, P_{23}, P_{24}, P_{31}, P_{34}, P_{41}, P_{42}, P_{43}, P_{44} will compose the set P used to evaluate the approximation.

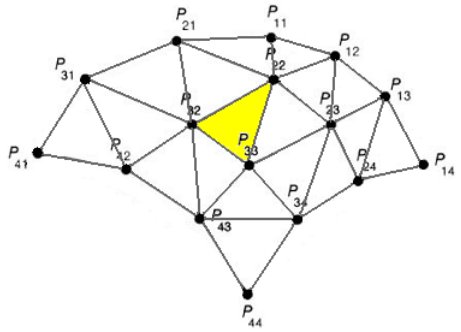


Figure 4.3: A face vertices and its neighbor vertices

The use of the face neighbor vertices is important to improve the quality of the approximation by getting its local curvature behavior. If the mesh face vertices and its

neighbor vertices are coplanar, the face approximation is a plane patch, if not, the face is approximated by a smooth curved surface patch. After the curved patch is generated, the error between the patch and the set P is evaluated. If this error is higher then a user specified limit, the approximation is discarded and the face is approximated by a plane. The model surface approximation is then a collection of smooth curved and plane patches. For the curved patches, we use the B-Spline patches representation. The B-Splines can give good approximation for a large variety of solids and still provide high continuity degree and local control, as discussed in section 2.5.3.

Section 2.6 presented the use of B-Spline surfaces to approximate a cloud of unorganized points using B-Spline surface. In the context of approximating a model surface, the set P is the input data points for the B-Spline patch, e. g. the P points are the p_τ in the Equation 2.35. Then, the face curved approximation follows the steps of the B-Spline patch evaluation through the least square formulation. Figure 4.4 illustrates the approximated curved patch for the vertices set of Figure 4.3.

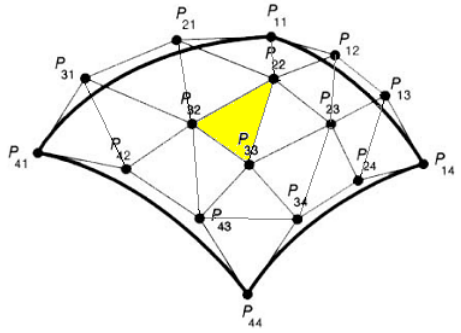


Figure 4.4: Approximated patch

The approximated patch should be able to evaluate: the closest point on the patch of a point in the 3D space; the curvature of a point on the patch; and the distance between a patch and a point in the 3D space.

A good approximation of the mesh nodes is very important to preserve the model geometric characteristics during the process of improving the mesh. The smooth surface approximation technique presented here guarantees this. It reduces approximation errors and can be used to approximate the model surface of a large variety of solids.

4.2.2 Algorithm

The algorithm of our proposed remeshing scheme is presented in Table 4.1; the surface mesh is modified iteratively in order to improve the mesh elements quality.

```
1:  Preliminary Definitions:
2:      Obtain the patch set representing the region that will be remeshed
3:      Evaluate the permitted deviation between the surface mesh and the surface model
4:      Fill the edges set E
5:  Remeshing process:
6:      while the mesh is modified
7:          for each edge e ∈ E
8:              if edge-collapse preserves geometry and increases mesh quality
9:                  collapse e
10:             else if edge-swap preserves geometry and increases mesh quality
11:                 swap e
12:             else if edge-split improves geometry or mesh quality
13:                 split e
14:             end if
15:             relocate the edge e vertices
16:         end for
17:     end while
```

Table 4.1: Our Remeshing Algorithm

Some lines of the algorithm in Table 4.1 are detailed as follows.

Line 2: For each face mesh, a surface patch that can be a plane or a B-spline patch is generated and associated to it. When the model is generated by the Boolean and assembly application, the patches of the faces in the intersection area are generated before the step of elimination of all undesired elements in the boundary evaluation (section 2.2.4.4) and passed to the remeshing process. The vertices that will be destroyed help to guarantee a better approximation of the surface model. This guarantees a better approximation in the intersection areas. If the model is generated by an acquisition process the patches for all the faces are generated. Only points of the same region as the face that is being approximated can be used in its patch generation. The maximum number of points used for the B-spline patch evaluation is set as 15. This number was chosen to give to a sphere of radius 100.0 and maximum discretization error equal to 5.0 an approximation error equal to 1.0%. The error that a B-Spline patch can have in relation to the vertices that

generated it is limited to a maximum equal to 2.0%. When the error is higher than 2.0%, the patch is discarded and the model region is approximated by a plane. Then, regions with high curvature need higher discretization to guarantee an acceptable smooth approximation.

Line 3: The deviation (AVG_TOL) between the initial surface mesh and the surface model is estimated as an average of the distances between the mesh faces barycenters and the face approximation. During the remeshing process, this value is used in conjunction to the local curvature behavior to test if the local mesh operators are degrading, improving or maintaining the model geometric characteristics. Estimating the deviation as an average of the initial mesh edges deviation, the resultant mesh will have the same magnitude order for the mesh elements number, but with improved quality. The AVG_TOL could be set by the user: for big deviation values, the remeshing algorithm would decimate the mesh, decreasing the number of mesh elements and degrading the model geometric characteristics; for small values, the algorithm would enrich the mesh, increasing the number of mesh elements and reducing the deviation between the mesh and the model geometric approximation.

Line 4: The remeshing approach works on an edge set during the adaptation process. For models generated by Boolean and assembly operations, this test set consists only of the edges from the faces modified by the the intersection process (section 2.2.4.2). For models evaluated through an acquisition process, all the model edges participate in the process.

Line 8: Before applying the edge-collapsing operator, all the mesh elements that share the vertex that will be removed are tested. To decide if the operator can be applied, two conditions need to be verified: first, the distance between the affected mesh edges and the geometric approximation of the model must be within the limited deviation; second, the minimum angle of the mesh elements must increase in the resultant mesh. If both conditions are true, the operator is applied. To limit the deviation between the new configuration and the model approximation, the distance of the edge midpoint to the surface approximation is evaluated. This distance should be smaller than the maximum permitted deviation at the edge midpoint, which is:

$$maximum\_deviation = \left(\frac{\alpha}{curvaturevalueatthemidpoint} \right) * AVG\_TOL, \qquad (4.1)$$

where α is a constant evaluated to give maximum distance equal to AVG_TOL for all edge midpoints of a sphere with radius of 100.0 and maximum discretization error equal to 5.0, and $curvature(midpoint)$ gives the mean curvature[i] of the approximated patch at the closest point in the patch of the edge midpoint.

Line 10: The edge-swapping tests its ability to increase the internal angles of the two faces that share the edge. To be swapped, the new edge must maintain the elements topology valid (do not invert any of the faces normal) and do not exceed the maximum deviation at the new edge midpoint (Equation 4.1).

Line 12: Edge-splitting is applied when the minimum angle of the two sharing faces can be increased, or to guarantee that the mesh elements do not go too far from the approximation. The maximum distance between the edge and the model approximation is limited, and its value is K times $themaximum\_deviation$, where K is also evaluated to result in an improved mesh for a sphere with radius equal to 100.0 and maximum discretization error equal to 5.0 with the same magnitude order as its initial mesh.

Line 15: The vertex relocation are always applied. The new vertex location is evaluated in order to improve the quality of all the elements that share the vertex. These evaluation is done mapping the faces that share a vertex into a plane and solving the problem there. Then, the resultant location is brought back to the approximated model surface. It is important to check if the new location do not invert any of the face normals of the elements that share the vertex.

The computational cost of the surface approximation step is proportional to the number of mesh faces and the cost of the remeshing part is proportional to the number of edges. Since the numbers of faces and edges are proportional to the number of mesh vertices, the computational cost of the entire scheme is proportional to the number of mesh vertices.

[i]The mean curvature is the average of the maximum and minimum normal curvature

4.3 Implementation Characteristics

This work is part of a bigger project at our research group, which involves the development of a multi-representational solid modeler (Gopac Solid Modeler - GSM) to be used for the definitions of 3D geometric models. The GSM architecture was already presented in other works [Mag98, Mag00a, Nun02]. Its main goal is the generation of high quality surface meshes to be used as input data for tetrahedral volumetric mesh generators, which can be used to solve electromagnetic problems by the finite element method.

Basically, the GSM is composed by four main subsystems. Figure 4.5 shows them and their interconnections.

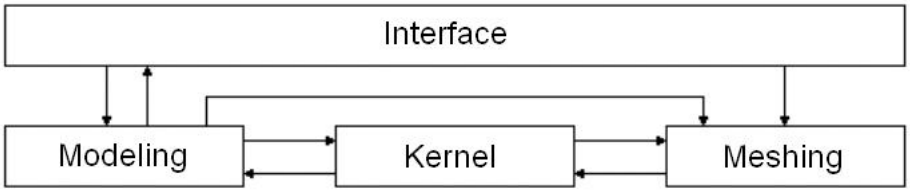


Figure 4.5: GSM structure [Nun02]

The four subsystems are implemented with object-oriented concepts which facilitates most of the modifications and the inclusion of new features. They operate in a independent form, and collaborate among themselves by functionally links:

- The *interface* subsystem handles the user access on model creation, edition and visualization. This subsystem captures the tasks to be realized and activates the procedures of the other subsystems. The results are shown by the *interface*.
- The *modeling* subsystem translates the *interface* requests into commands to generate or modify the internal representation. This subsystem interprets the actions for generating a new model description or a modification on a existing model, and call the routines of the *kernel* subsystem. The CSG scheme is handled by this subsystem.
- The *kernel* subsystem realizes the maintenance, the management and the access to the set of internal representation (B-rep data structures). This subsystem includes functions for storage of the objects description and compositions in permanent databases. The information of the internal representation can be read by the other subsystems procedures, but it only can be modified by this subsystem procedures.

- The *meshing* subsystem generates the finite elements mesh from the geometric description supplied by the B-Rep representation. A division of the problem domain (the model) in a set of sub-domain (the elements), respecting the boundary and the interfaces is obtained.

During this project execution, we worked mostly in the modeling and meshing subsystems. The step of avoiding the planar facet modifications is done during the boundary evaluating (section 2.2.4) that is part of the modeling subsystem. The remesh process and the smooth surface approximation were located in the meshing subsystem. We also introduced the functionalities to evaluate and present the surface and volumetric mesh quality metrics through tables and graphs.

The Euler operators [Mag00a, Nun02] responsible for the B-rep data structure updates needed some modifications. The operators were designed to work with any kind of facets, but in practice, the implemented functions did not work well with non-convex polygons and polygons with holes.

The evaluation of the polygon normals received special attention, since the polygons can be non-convex. The convex hull of the polygon or at least three vertices of it should be computed for the correct polygon normal evaluation. This function must have a low computational cost, since it is used many times during any construction or modification of a model.

Aiming to improve the model approximation of the models generated by Boolean and assembly operations, the mesh vertices are labeled during their creation to indicate their degree of freedom. These labels mean the vertex capacity of movement and it can be: *zero*, when a vertex cannot move; *one* when a vertex can move on a model edge; *two*, when a vertex can move on a surface; and *intersection*, when the vertex can move on a intersection line. The vertices of a model generated from an acquisition process are able to move on the surface and receive degree of freedom equal to *two*. The degree of freedom is taken in consideration by the local mesh operators during the optimization process. For example, a node with degree of freedom equal to *zero* cannot be moved by the vertex relocation operator or removed by the edge collapsing operator; or if an edge has its two endpoints with degree of freedom equal to *zero*, *one* or *intersection*, it cannot be swapped.

The SINTEF LSMG library (version 1.0) was used to evaluate and handle the B-spline

patches [SIN06]. The LSMG is a multigrid approach for solving least squares approximation to scattered data with B-splines. The library was developed at the SINTEF (Foundation for Scientific and Industrial Research at the Norwegian Institute of Technology) Applied Mathematics. The methods have been developed for making high quality smooth surface approximations to scattered data in applications such as geological modeling, GIS and CAD/CAM. Special attention has been payed to i) creating large surfaces fast and ii) "natural" extrapolation of surfaces outside the domain of the scattered data or to areas of the domain with no scattered data. The approximation scheme is a least square fit with a thin plate spline smoothing term. A set of multigrid schemes are implemented to solve the equation system that arises.

For planar facets triangulation, the Triangle program (version 1.6) is used. It is a C program developed by Carnegie Mellon University [She96] for two-dimensional mesh generation and construction of Delaunay triangulations, constrained Delaunay triangulations, and Voronoi diagrams. Triangle is fast, memory-efficient, and robust; it computes Delaunay triangulations and constrained Delaunay triangulations exactly. Guaranteed quality meshes are generated using Ruppert's Delaunay refinement algorithm. Features include user-specified constraints on angles and triangle areas, user-specified holes and concavities, and the economical use of exact arithmetic to improve robustness.

The TetGen program (version 1.4.1), developed by Hang Si[Si04], is coupled to GSM to generate the finite element volumetric meshes. It is based on the algorithm described by Shewchuk [She97], which is a smooth generalization of Ruppert's algorithm [Rup93] to three dimensions. Given a complex set of vertices, constraining segments and facets in three dimensions, this algorithm can generate a mesh of Delaunay tetrahedra, which bounds on a maximum quality factor Q. Theoretically, the input can not have angles less than 90°. Nevertheless, the implementation of TetGen shows that the algorithm surpasses the theoretical bounds.

The Power Crust Software (version 1.2) is used for surface reconstruction from a points cloud. It is an implementation of the power crust algorithm and medial axis transform approximation developed by Nina Amenta, Sunghee Choi and Ravi Kolluri, [Ame01]. The power crust algorithm works well (in fact, it's proved correct) on a dense point sample from the surface of a smooth object without boundary. The input might not be sufficiently dense, it might have noise in it, it might be from a surface with boundary, it might fail to

cover the whole surface, or it might be from a surface with sharp corners. The software can often get a good output in these situations as well, and it includes some command-line parameters which can be twiddled to try to improve the quality of the output. No twiddling is necessary on good inputs.

4.4 Discussion

The remeshing approach proposed here is composed by two methods. The first can be considered as a preprocessing step, since it modifies the boundary evaluation procedure of the Boolean and assembly operations to permit polygonal facets. This step aim is to increase the volumetric mesh generator degree of freedom, giving to it a surface mesh with less unnecessary information. The edges of the planar facet triangulation do not add any important geometric information to the model. However, they difficult the volumetric mesh work, because once they are part of the input surface mesh, they cannot be removed. The improvements obtained by this modification are restricted only to models formed by planar facets. The 3D curved solids, such as spheres, torus and the reconstructed models are still represented by triangulated surface meshes. Usually, the simple primitives are generated with the desired degree of quality at the beginning, but after the Boolean and assembly operations, not only elements with thin angles are generated. The intersecting process also generates small edges that, as well as the sharp angles, have a bad influence on the volumetric mesh generation. Then, to increase the small edges resulting from the Boolean and assembly operations applications and to improve the overall mesh of the reconstructed models, a more general improvement step is necessary. Our adaptive improvement method is responsible to improve these model meshes.

The adaptive method is based on the works presented in chapter 3 to improve indistinctly the surface mesh of models generated by application of Boolean and assembly operation over 3D primitives and models reconstructed from a set of unorganized points. It performs series of local mesh modifications driven by the smooth approximation of the model surface. Section 4.2.1 introduced a new approach to evaluate the model surface approximation considering the mesh nodes. The model approximation is a collection of patches that can be planar or curved patches generated for each face. The face curved approximation are the B-Splines surface patches evaluated through the least square formu-

lation using the vertices of the face that is being approximates and its neighbors vertices. Generating approximations for each face and also using the vertices around it avoid the global parametrization and decrease the approximation errors.

The implemented local mesh modification operators only performs modifications if the resultant configuration preserves the model geometry and the mesh quality is increased.

Our method is able to give good approximation for models generated by application of Boolean and assembly operation over 3D primitives and models reconstructed from a set of points.

Next chapter presents several results to illustrate the improvements achieved with the approaches here presented.

Chapter

5

Results

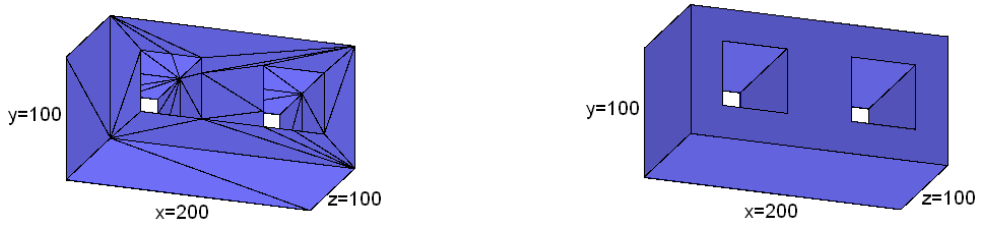
In this chapter, we present and discuss the achieved improvements on surface mesh quality for simple and more complex models obtained from an acquisition process or from the Boolean and assembly operation application over predefined primitives. We also show the improvements in finite element volumetric mesh quality when its surface mesh is improved.

5.1 Models Generated by Boolean and Assembly Operations

5.1.1 Results when the planar facet triangulation is avoided

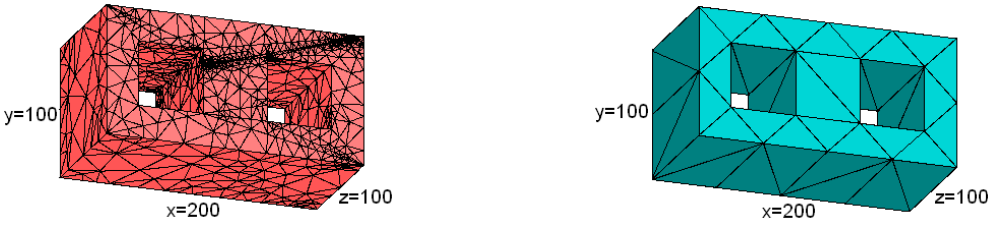
To compare the improvements achieved avoiding the planar facets triangulations, a simple model is chosen to start. The model is formed by a prism from which two smaller prisms were subtracted to produce holes. Figure 5.1(a) shows the meshed model with planar facets triangulation, while the surface mesh in Figure 5.1(b) has polygonal facets.

Figure 5.2 presents the generated volumetric meshes using the surface meshes in Figure 5.1 as inputs. Using the triangular surface mesh (Figure 5.1(a)) as input resulted in Mesh I (Figure 5.2(a)). Mesh II (Figure 5.2(b)) was generated using the polygonal surface mesh (Figure 5.1(b)) as input.



(a) Triangular faces (b) Polygonal faces

Figure 5.1: Surface meshes of the same prism with holes



(a) Triangular surface mesh as input (Mesh I)

(b) Polygonal surface mesh as input (Mesh II)

Figure 5.2: Volumetric meshes of the same prism with holes

A maximum radius-edge ratio equal to 2.0 was specified to generate the volumetric meshes in Figure 5.2. The number of tetrahedra generated to try to achieve this bound decreases drastically, from 7288 tetrahedra, in Mesh I, to 66 tetrahedra in Mesh II. This number is an important measure, since the processing time for an electromagnetic simulation by the finite element method is bigger for a mesh with a larger number of elements. Attempting to create tetrahedra with better quality, the volumetric mesh generator created very small elements in Mesh I, the smallest one has volume equal to $2.6548e-05$, when in Mesh II it is 20833. The small tetrahedra in Mesh I result in another drawback for the problem simulation. The total time to create the volumetric mesh dropped from 1.662s (Mesh I) to 0.02s (Mesh II). Table 5.1 shows the radius-edge ratio quality factor distribution for the meshes. The number of tetrahedra that are out of the quality bound is 2% (150 tetrahedra) for Mesh I, while all tetrahedra from Mesh II do not exceed the bound.

For a maximum tetrahedron volume constraint equal to 50 and a maximum radius-edge ratio equal to 1.41, new meshes were generated using the different approaches. The new approach produced 58056 tetrahedra within the bounds. As expected, the worst results were for the mesh that respected the facets triangulation, 65269 tetrahedra were produced without respecting any of the constraints.

Table 5.1: Radius-edge ratio quality factor (Q) Distribution

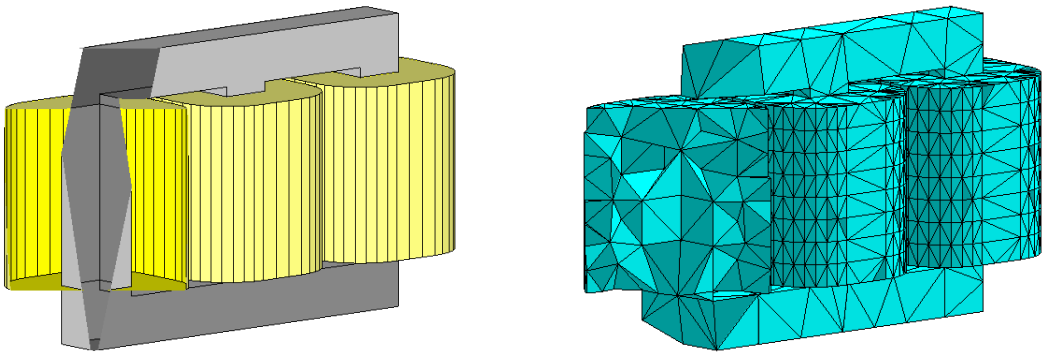
Q bounds	Mesh I elements (%)	Mesh II elements (%)
< 0.707	0.6449	0
0.707 - 1.0	23.2711	0
1.0 - 1.1	11.1690	0
1.1 - 1.2	10.8809	0
1.2 - 1.4	21.1992	9.0909
1.4 - 1.6	16.1773	90.9091
1.6 - 1.8	10.7162	0
1.8 - 2.0	3.8831	0
2.0 - 2.5	0.6586	0
2.5 - 3.0	0.3705	0
3.0 - 10.0	0.8233	0
$10.0 >$	0.2058	0

When just a maximum radius-edge ratio equal to 1.0 is set and no maximum tetrahedron volume constraint is imposed, only the new approach was able to generate the volumetric mesh and all its 406 tetrahedra respected the bound limit.

The results showed that when the surface mesh has elements with small internal angles, the volumetric mesh generator can not do much to improve the mesh. The small angles from the surface mesh have to remain in the final volumetric mesh, and they produce badly shaped tetrahedra. When the volumetric mesh generator tries to remove them and improve the mesh quality, it generates many small elements, which results in a large increase in the total number of elements. Avoiding the planar facet triangulation makes the work of the volumetric mesh generator easier and, consequently, faster. It also guarantees the specified mesh quality. Another good side effect is that the required data structure size for the surface mesh and the number of intersection evaluations are reduced, since fewer edges and faces are generated.

Figure 5.3(a) shows the surface mesh for the same model presented before in Figure 1.1(b), but here, the planar facet triangulation was avoided. Figure 5.3(b) shows its volumetric mesh, which could not be generated before. The view point is the same for both figures and it was chosen to allow the visualization of the inside and outside of the meshes. To generate the three-phase transformer model, the difference operation was applied to obtain the core and each winding; and the assembly operation was also applied to automatically guarantee the mesh compatibility between each winding and the core, as they share a common interface that must be respected by the volumetric mesh generator.

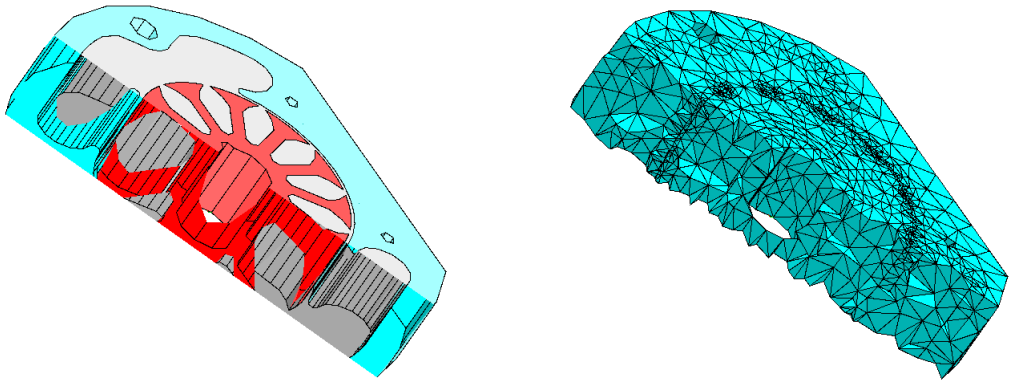
The three-phase transformer volumetric mesh was generated with 4622 tetrahedra and the maximum radius-edge ratio limited to 2.0.



(a) Polygonal surface mesh (b) Volumetric mesh

Figure 5.3: Three-phase transformer model constructed by applying difference and assembly operations over 3D primitives

The surface and volumetric meshes of an electrical machine model were also built and they are presented in Figure 5.4. Before the modifications, even the surface mesh could not be obtained due to precision problems, because the resulting triangles were too small. The number of tetrahedra generated was 30794 with a maximum radius-edge ratio restriction equal to 2.5.



(a) Polygonal surface mesh (b) Volumetric mesh

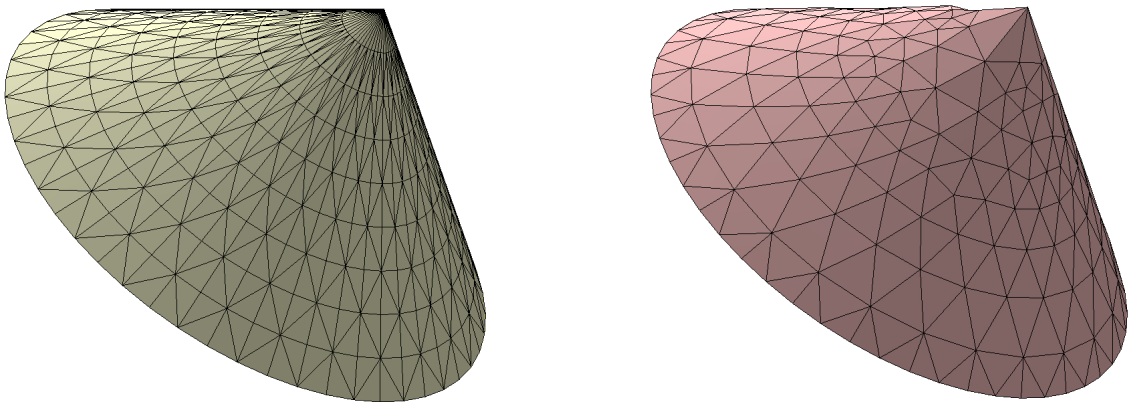
Figure 5.4: An electrical machine model and its volumetric mesh

Before the modifications described in section 4.1, many surface meshes generated by the application of the Boolean and assembly operations could not be used as input to the volumetric mesh generators, because of their badly shaped elements. Avoiding the triangulation of the planar facets the volumetric meshes of many 3D models can be generated. Even some surface meshes, which could not be evaluated before due to accuracy problems, as the example in Figure 5.4, can be generated now.

5.1.2 Results Applying the New Remeshing Scheme

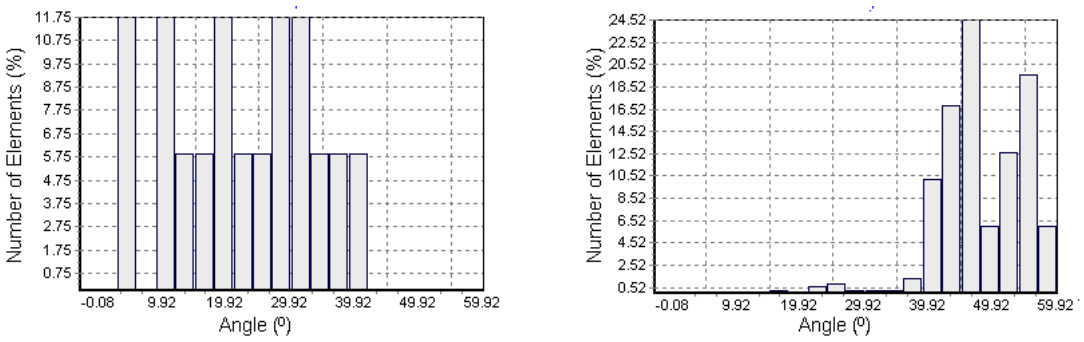
The improvements achieved when our remeshing method is applied to the surface meshes generated by the application of the Boolean and assembly operations to predefined primitives are very significant.

To start, a simple model was chosen. Figure 5.5(a) presents a cone with its original mesh and Figure 5.5(b) shows the same cone with a new mesh after the remeshing process is applied. One can see that there are modifications in the elements shapes and their interconnections. Quantitatively, the number of elements decreases from 851 in Figure 5.5(a) to 363 in Figure 5.5(b). Figure 5.6 shows the distribution of the minimum angles for both cone meshes. The number of elements with low minimum angle less than 30° was reduced from 70% to 1%. 95% of the elements in the remeshed model have their minimum angle higher than 40°, instead of only 6% in the original mesh.



(a) initial mesh (b) remeshed model

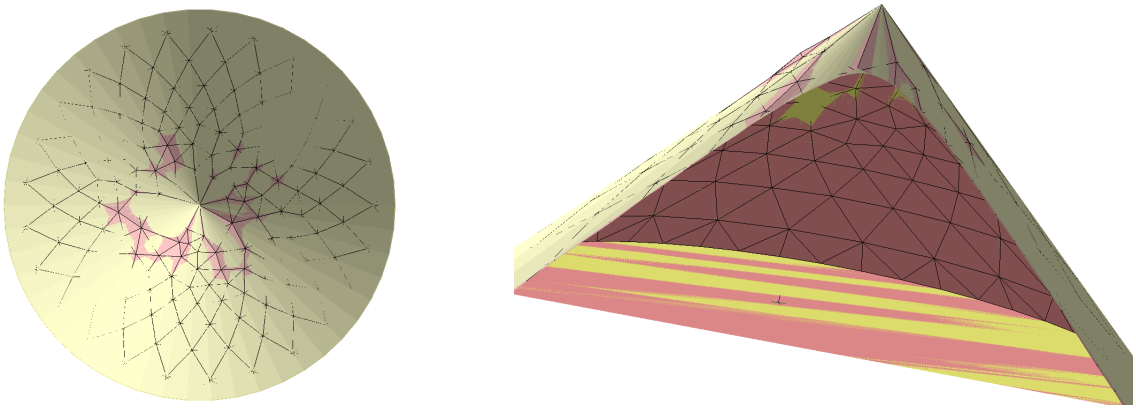
Figure 5.5: Surface meshes for a cone



(a) initial model (b) remeshed model

Figure 5.6: Surface mesh quality graphs considering the minimum angle measure for the mesh in Figure 5.5

The control of the deviation during the remeshing process can be seen in Figure 5.7 that shows the cone initial mesh and the improved mesh at the same location. The initial mesh (Figure 5.7(a)) is in yellow and its edges were not drawn. The remeshed model (Figure 5.7(b)) is represented in red and its edges are in black.



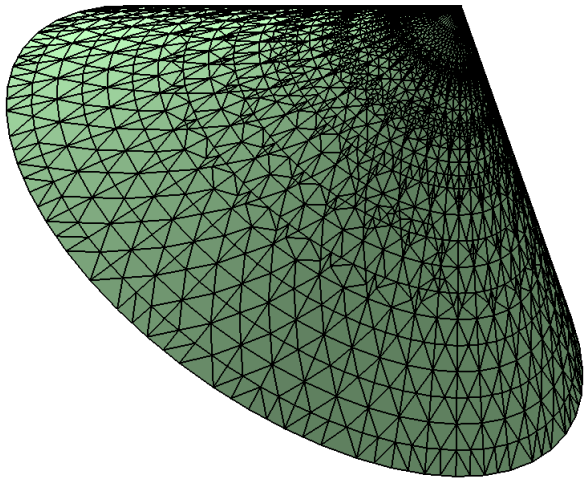
(a) Remeshed model is red and the initial mesh is in yellow

(b) Zoom into the meshes

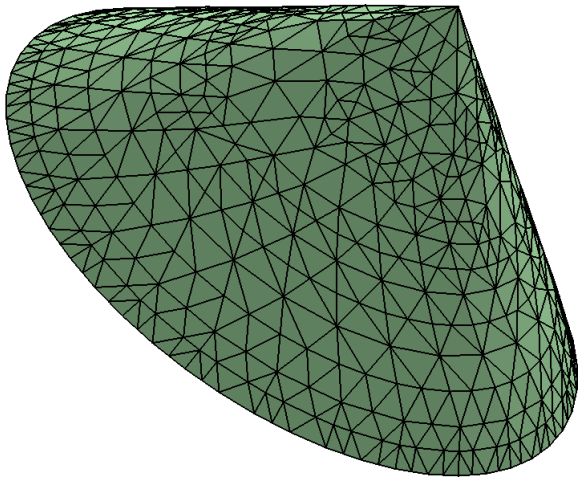
Figure 5.7: Initial mesh of the cone model on top of the remeshed cone model

The lines that appear in Figure 5.7(a) are the edges of the remeshed model. It is possible to see most of them, showing that the remeshed model do not deviate much from the initial mesh. The elements that share the cone top in the remeshed model are the ones that deviate more from the original mesh. Unfortunately, this happens due to the fact that the model surface approximation use a limited number of points to construct the smooth patch and in that area the points are very close to the others. This makes the approximation evaluation less accurate. But, since the degree of freedom of the cone top vertex is *zero*, its position is preserved.

Figure 5.8 shows the volumetric meshes for the cone models in Figure 5.5, with a maximum radius-edge ratio restriction equal to 2.0. The number of elements in Figure 5.8(a) is 19570, while it is only 2417 in Figure 5.8(b). The radius-edge ratio range varied from 0.63 to 10.30 for the first mesh, and from 0.65 to 1.99 for the second. 976 elements had their radius-edge ratio higher than 2.0 in Figure 5.8(a), while none element exceeded this limit in Figure 5.8(b).



(a) initial model



(b) remeshed model

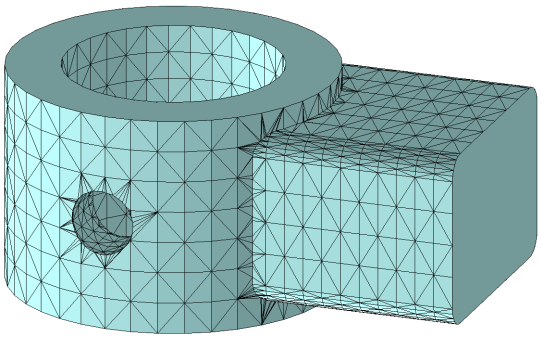
Figure 5.8: Volumetric meshes for the meshes in Figure 5.5

Figure 5.9(a) shows the surface mesh of a model generated after the application of one union and two difference operations; Figure 5.9(c) illustrates the mesh when planar facet triangulation is avoided during the Boolean and assembly operations applications; Figure 5.9(e) presents the mesh after the remeshing step at the intersection region; and Figure 5.9(g) is fully remeshed. Figures 5.9(b), 5.9(d), 5.9(f) and 5.9(h) show the zoom in a section of the intersection region for each model. Comparing the surface meshes, one can see that the overall geometric characteristics of the original model were preserved, while the vertices have their locations and interconnections modified. There is a difference in the models geometry and it can be better seen by comparing the Figures 5.9(b), 5.9(f) and 5.9(h) in the marked area, but it is very small.

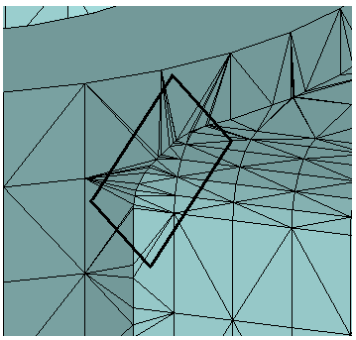
The lowest minimum angle (0.11°) appears in Figure 5.9(a) that illustrates the model obtained without any kind of remeshing step. Figure 5.9(c) shows the mesh with minimum angle equal to 15.0°, this mesh is the result of the Boolean operations application when the triangulation of the planar facets are avoided. The minimum angle equal to 14.66° appears in Figure 5.9(e), which shows the resulting surface mesh when the remeshing process is applied to the intersection areas. When the model is fully remeshed after the Boolean operations applications the minimum angle increases to 15.4°. The number of faces in the surface mesh varied a little, the first is 2517, the second is 2106,the third is 2325 and the last is 1933.

The distribution of the minimum angles for each model presented in Figure 5.9 can be seen in Figure 5.10. The percentage of elements with minimum angle smaller than 30° varied from 38% in Figure 5.9(a) to 25% in Figure 5.9(c), 25% in Figure 5.9(e) and 10% in Figure 5.9(g). When the number of elements with minimum angle higher than 40°are counted, one can also see the improvements in the mesh: it starts as 18% in the first model, it goes up to 24% for the second, it is 26% for the third and finally it is 49%.

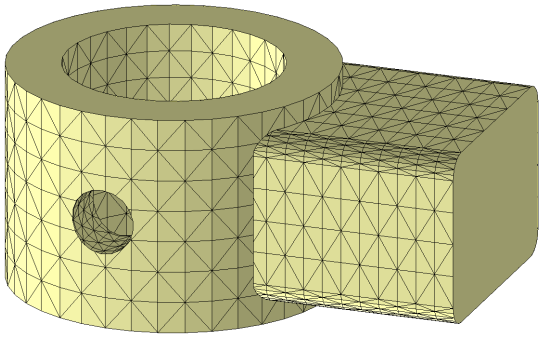
The distribution of the shortest edges for the mesh elements in Figure 5.9 are presented in Figure 5.11. The length of the mesh shortest edge increased from 0.04 in Figures 5.9(a) and 5.9(c) to 3.45 in Figure 5.9(e) and it slightly decreased to 3.33 in Figure 5.9(g).



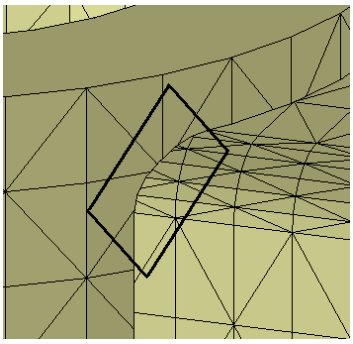
(a) without remeshing



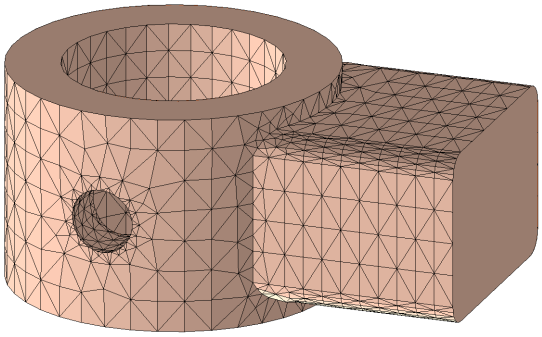
(b) zoom in Figure 5.9(a)



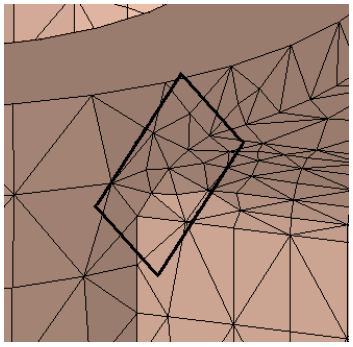
(c) avoiding planar facets triangulation



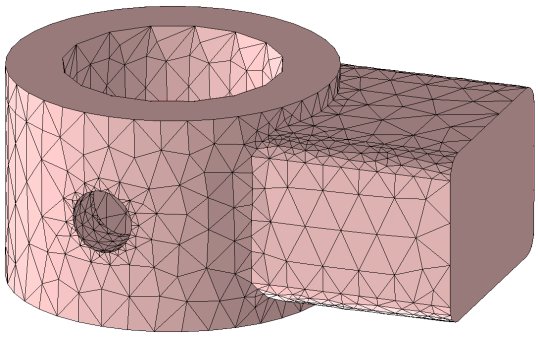
(d) zoom in Figure 5.9(c)



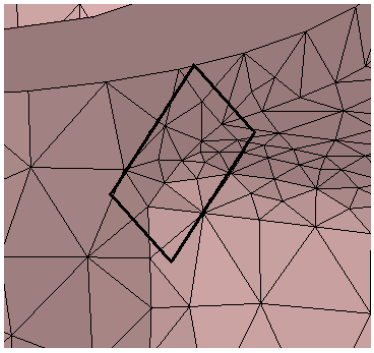
(e) remeshing the intersection regions



(f) zoom in Figure 5.9(e)

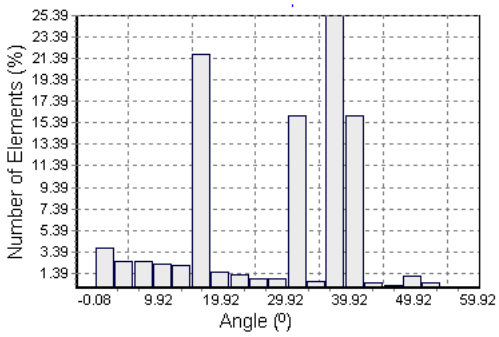


(g) fully remeshing

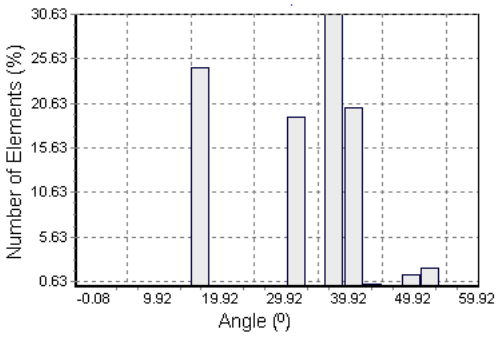


(h) zoom in Figure 5.9(g)

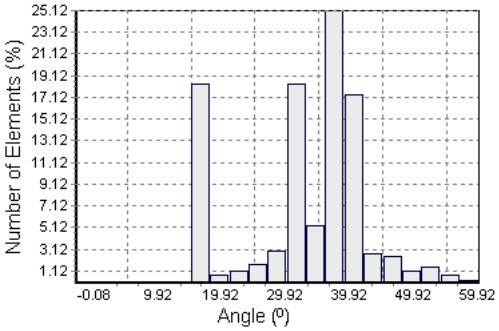
Figure 5.9: Surface mesh of a model generated by application of one union and two difference operations



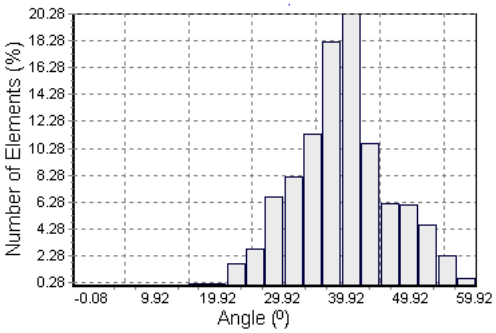
(a) without remeshing



(b) avoiding planar facets triangulation

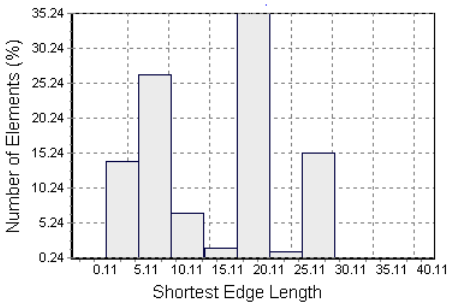


(c) remeshing the intersection regions

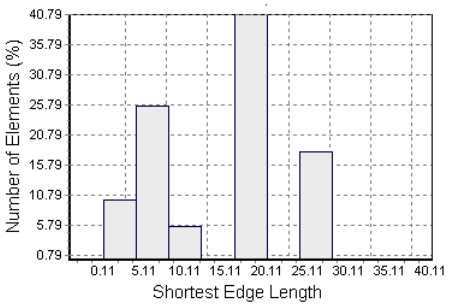


(d) fully remeshing

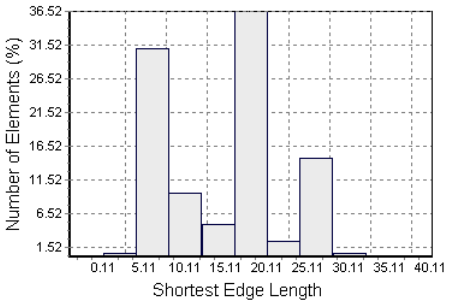
Figure 5.10: Surface mesh quality graphs considering the minimum angle measure for the models in Figure 5.9



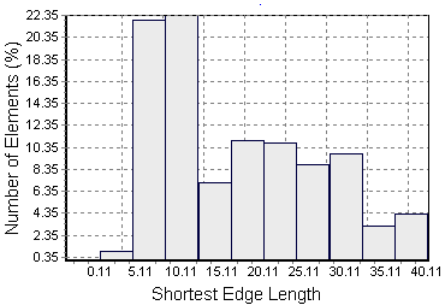
(a) without remeshing



(b) avoiding planar facets triangulation



(c) remeshing the intersection regions



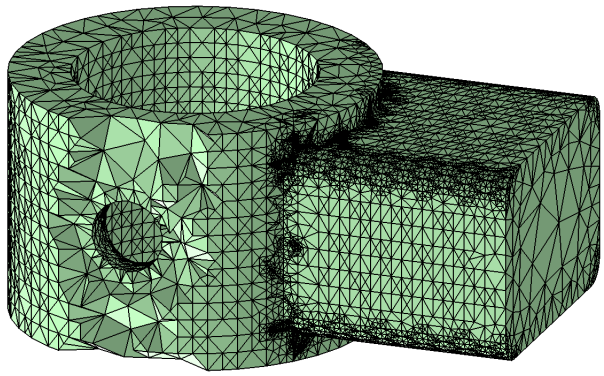
(d) fully remeshing

Figure 5.11: Surface mesh quality graphs considering the shorted edge length of the models in Figure 5.9

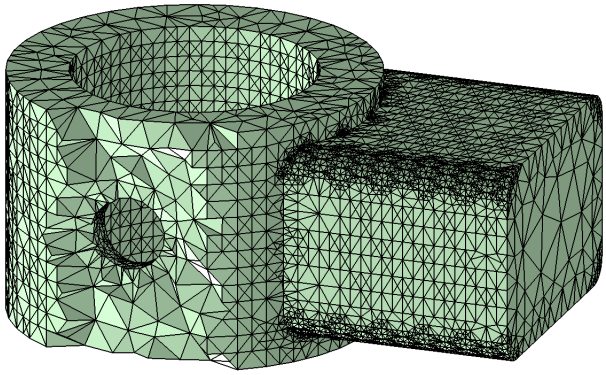
The volumetric meshes in Figure 5.12 where generated using the surface mesh shown in Figure 5.9, with a maximum radius-edge ratio restriction equal to 2.0. The number of elements in the volumetric meshes, which could be generated, decreased considerably. It started as 113117 in Figure 5.12(b), decreasing to 40924 in Figure 5.12(b), 31784 in Figure 5.12(c) and it achieved 18313 elements in Figure 5.12(d). The maximum radius-edge ratio also decreased from 1204.5 in the model without remeshing; to 35.72, for the model with polygonal faces in the intersection areas; 6.82, when the intersection areas where remeshed, and it achieved 3.63 when the model is fully remeshed. 5217 elements have their radius-edge ratio higher then 2.0 in the first mesh, 381 in the second, 349 in the third and 59 in the last. The minimum dihedral angle increased from 0.082° in the first model to 0.72° in the second, 2.37° in the third and 3.61° in the last model.

The volumetric meshes presented in Figures 5.12(b) and 5.12(c) were used in a electromagnetic simulation to exemplify the improvement on the conditioning of the global stiffness matrix. The stiffness matrix were generated for the wave equation discretization for the scattered problem, without considering the surface integral. The problem was simulated using the definitions on [Jin93]. The stiffness matrix for the mesh in Figure 5.12(b) was generated with 60579 equations. The number of equations for the mesh in Figure 5.12(c) was 47880. The condition number reduced from order of 10^9 in the first simulation to 10^5 in the second. As discussed in section 2.4, the numerical solution accuracy increases when the magnitude order of the condition number decreases. This proves that an improvement on the shape of the finite elements increases the accuracy of the electromagnetic solutions.

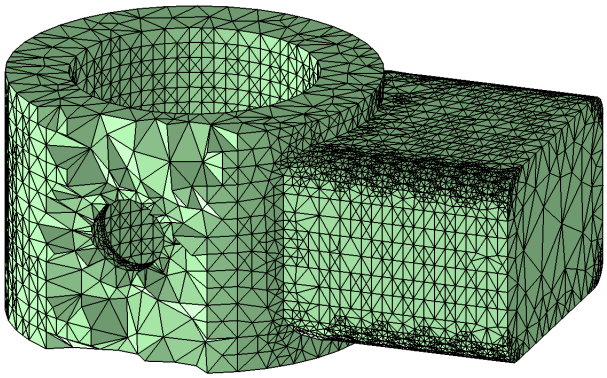
Other examples of resultant meshes of the Boolean and assembly operations application over simple primitives are show in Figures 5.13 and 5.14. Both examples presented good improvements on their surface meshes.



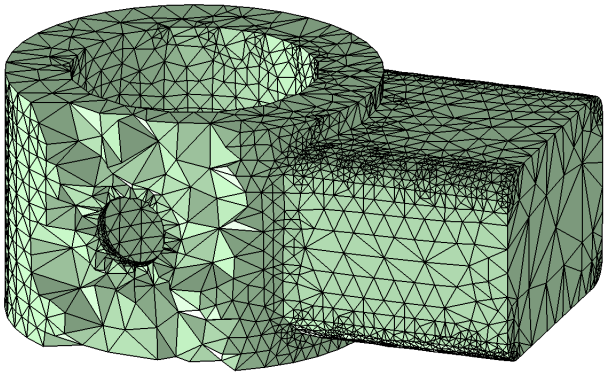
(a) without remeshing



(b) avoiding planar facets triangulation

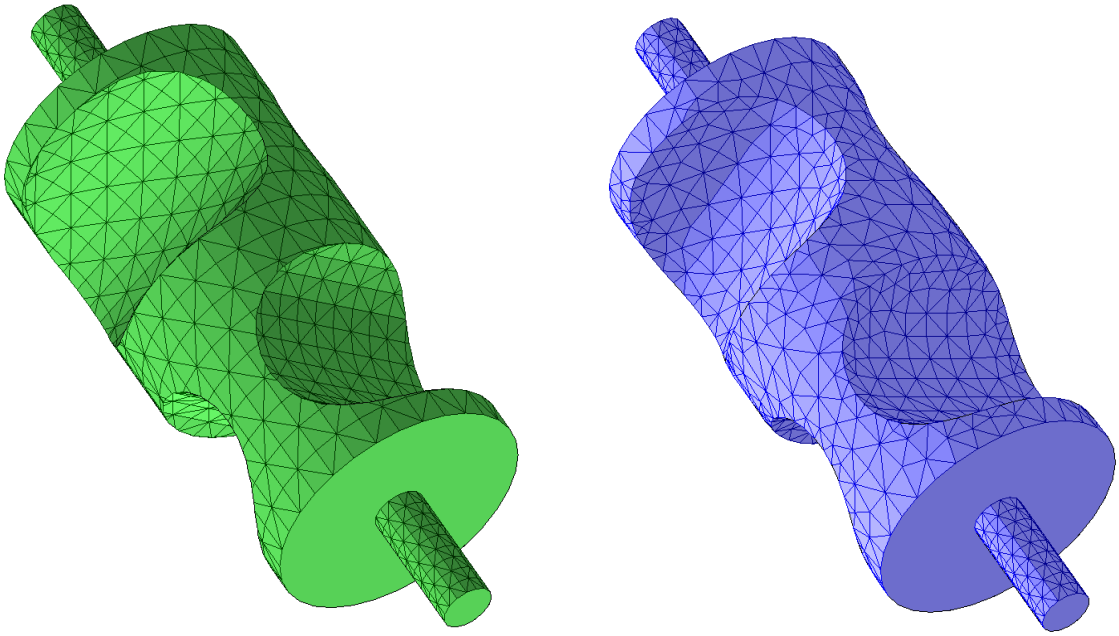


(c) remeshing the intersection regions



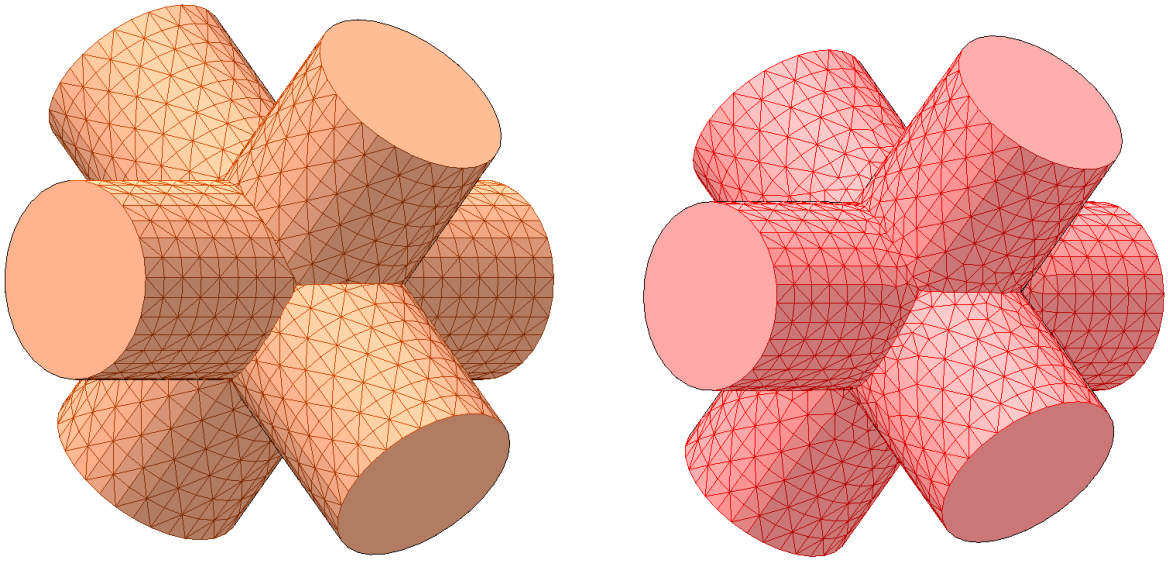
(d) fully remeshing

Figure 5.12: Volumetric mesh of a model generated by application of one union and two difference operations



(a) polygonal mesh (b) remeshed model

Figure 5.13: Surface meshes for the result of union and difference among cylinders with different radius; (a) has 2162 mesh faces and its minimum angle is 0.63°; (b) has 2358 mesh faces and minimal angle equal to 24.8°



(a) polygonal mesh (b) remeshed model

Figure 5.14: Surface meshes for the union of three cylinders; (a) has 2910 mesh faces and the minimum angle is 5.61°; (b) has 2994 mesh faces and its minimum angle is 11.63°

5.1.3 Discussion

The triangulation of all faces during the intersecting step of the Boolean and assembly operation application makes the volumetric mesh generation very hard. Attempting to improve the volumetric mesh quality, the volumetric mesh generator introduces a large number of new points in the regions with thin elements. However, it does not succeed: when the surface mesh is very degraded, the generated volumetric meshes have a very large number of elements and high maximum radius-edge ratio. Sometimes, even the surface mesh generation is compromised, as in the electrical machine model (Figure 5.4), that could not be generated.

Avoiding the planar facets triangulation is very import to guarantee the model surface generation and also its corresponding volumetric mesh generation. However, avoiding the planar facet triangulation is not enough to guarantee good results in the volumetric meshes of models that have curved areas, as the model presented in Figure 5.9(c). The short edges generated by the Boolean and assembly operations, not only the small angles, also lead to a volumetric mesh with poor quality degree. Then, the remeshing step is primordial to improve these surface meshes.

After the remeshing process, the resultant meshes had their minimum angle and shortest edge increased. The geometric approximation in the intersection areas were preserved, because the method take advantage of using the smooth approximation generated before the step of removing the undesired elements in the boundary evaluation. When the model is fully remeshed there are more improvements on the mesh.

The number of volumetric mesh elements decreased drastically from the model without any form of improvement to the model fully remeshed. For instance, in Figure 5.12 the number of volumetric elements decreased from 113117 in the original mesh to only 18313 elements in the fully remeshed model. Also, the maximum radius-edge ratio became 330 times smaller than same metric in the first model.

The condition number associated to the stiffness matrix calculated for the mesh in Figure 5.12(b) is smaller then the condition number evaluated for the mesh in Figure 5.12(c). This result proves that an improvement on the shape of the finite elements increases the accuracy of the electromagnetic solutions by FEM, as discussed in 2.4.

The results showed good improvements in the surface mesh quality of models generated by the application of the Boolean and assembly operations over primitives, and better ones in the volumetric finite element mesh. They also demonstrated that the geometric characteristics of the original models remained correct after the remeshing process execution.

5.2 Models Generated by Acquisition Process

In this section, examples are provided to illustrate the improvements achieved on surface mesh quality of reconstructed models. The models presented here were downloaded from the AIM@SHAPE Shape Repository (A.I.M.A.T.S.H.A.P.E. - Advanced and Innovative Models And Tools for the development of Semantic-based systems for Handling, Acquiring, and Processing knowledge Embedded in multidimensional digital objects), [AIM06].

To evaluate the surface mesh quality and compare the meshes, Table 5.2 shows the statistics for the surface meshes of the 3D models that are presented in Figures 5.15, 5.19, 5.23 and 5.27. The number of vertices, number of faces, percentage of elements with minimum angle lower than 30° and percentage of elements with minimum angle higher than 40° are presented for the initial surface meshes and the remeshed models.

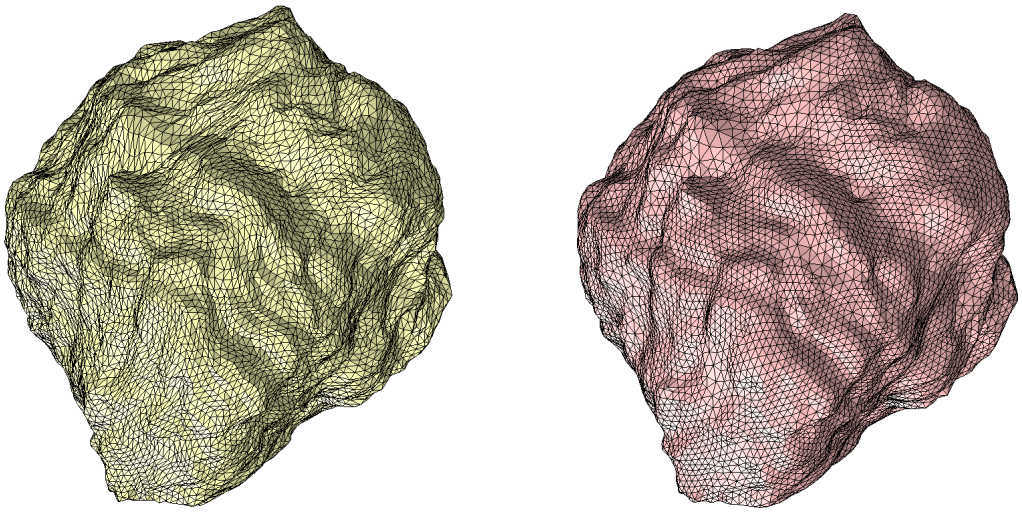
Table 5.2: Statistics for the original and the remeshed models

Model	Figure	vertices	faces	elem. $< 30°$	elem. $> 40°$
Asteroid (initial)	5.15(a)	10242	20480	30%	34%
Asteroid (remeshed)	5.15(b)	9634	19264	≈0%	95%
Rabbit (initial)	5.19(a)	16760	33515	28%	32%
Rabbit (remeshed)	5.19(b)	15169	30334	≈0%	95%
Human Torso (initial)	5.23(a)	12766	25528	52%	25%
Human Torso (remeshed)	5.23(b)	10099	20194	<0.5%	92%
Stanford Bunny (initial)	5.27(a)	20011	40018	75%	12%
Stanford Bunny (remeshed)	5.27(b)	13376	26749	<0.5%	90%

The number of surface mesh elements in the improved meshes compared to the initial meshes decreased for all models. The percentage of elements with minimum angle higher than 40° substantially increased and the number of thin elements drastically decrease achieving nearly zero for all models.

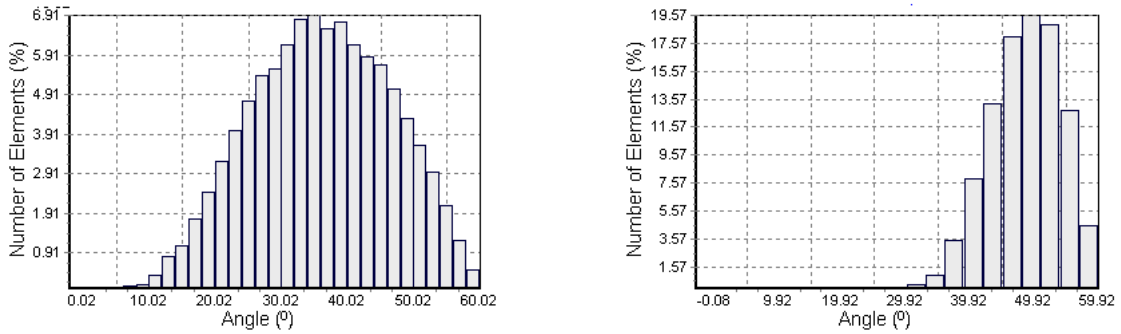
The better resultant mesh is the one for the rabbit model (Figure 5.19), because its features are very smooth. The more expressive improvement is in the remeshed model of the Stanford bunny model: it has 12% of its elements with minimum angle higher than 40° and after the remeshing process, this number jumped to 90%.

Figure 5.15 illustrates the asteroid model before and after the remeshing process being applied. Figure 5.16 presents the corresponding graphs for the distribution of the minimum angles for these surface meshes.



(a) Initial surface mesh (b) Remeshed Model

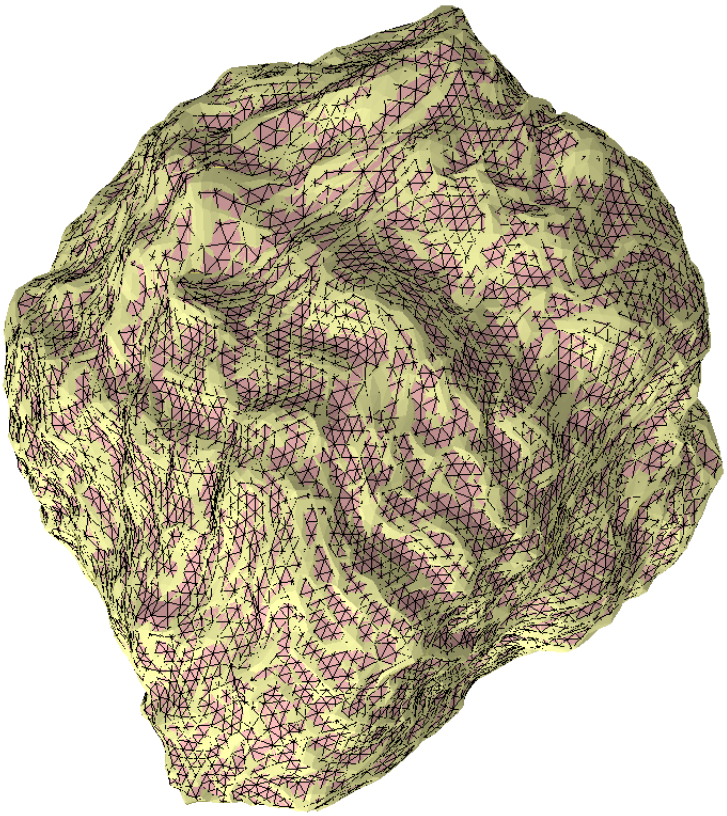
Figure 5.15: Surface meshes of the asteroid model



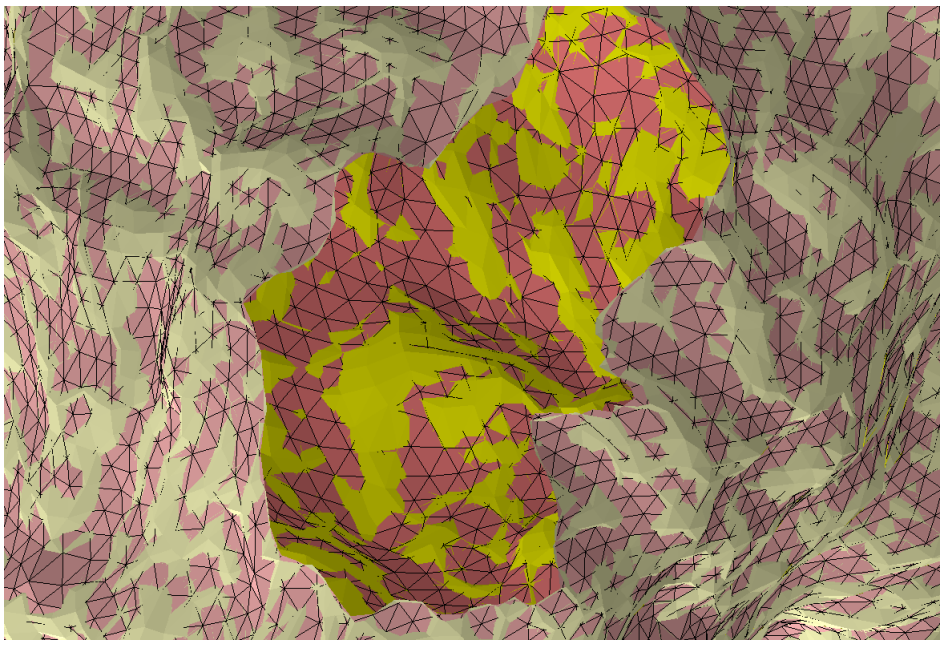
(a) Initial surface mesh (b) Remeshed Model

Figure 5.16: Surface mesh quality graphs considering the minimum angle measure for the asteroid model

To illustrate the control of the deviation during the remeshing process, Figure 5.17 shows the asteroid initial mesh and the improved mesh at the same location. The initial mesh (Figure 5.15(a)) is in yellow and its edges were not drawn, the remeshed model (Figure 5.15(b)) is represented with red. Due to the smooth approximation the sharp features were smoothed a little, but the majority of the mesh elements are on top of the initial mesh and geometric characteristics were preserved.



(a) Remeshed model is in red and the initial mesh of the asteroid model is in yellow



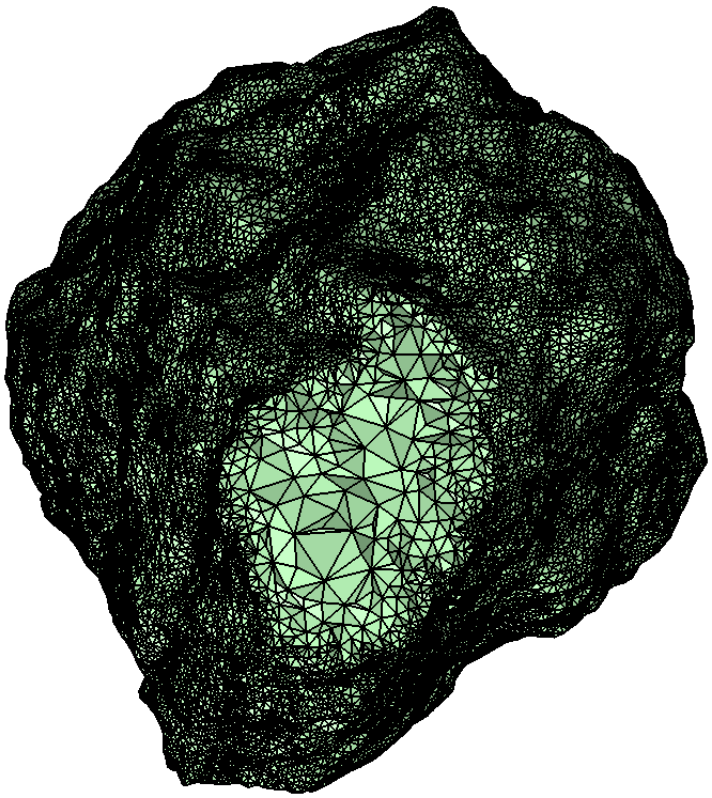
(b) Zoom into the meshes

Figure 5.17: Initial mesh of the asteroid model on top of the remeshed model

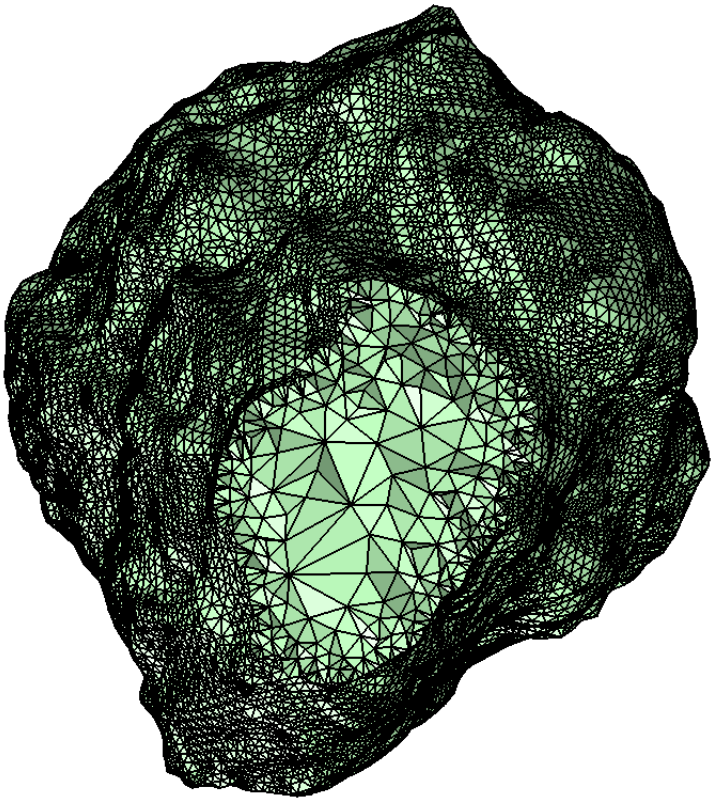
The volumetric meshes of the asteroid initial mesh and the improved mesh are presented in Figure 5.18. Both meshes were generated with the maximum radius-edge ratio limited to 2.0. The number of elements generated when the initial mesh was used as input is 303070 and its maximum radius-edge ratio is 56.42. The improved surface mesh produced a volumetric mesh with less than half elements of the first, 126739, and maximum radius-edge ratio equal to 55.56. The number of elements that exceeded the maximum radius-edge of 2.0 is 1881 for the first volumetric mesh and 178 for the second.

Figure 5.19(a) presents the initial rabbit model and Figure 5.19(b) illustrates the remeshed rabbit model. The distribution of the minimum angles of both surface meshes are shown in Figure 5.20(a) and Figure 5.20(b). Figure 5.21 illustrates a zoom in the rabbit model ears.

The volumetric mesh for the remesh rabbit model is presented in Figure 5.22. It was generated with maximum radius-edge ratio limited to 2.0. It has 147653 elements with the radius-edge ratio variating from 0.6220 to 65.8229. 61 elements have radius-edge ratio higher than 2.0. The volumetric mesh, using the initial mesh of the rabbit as input, was not generated. The initial mesh have some faces that intersects each other in points other than its vertices, which is an invalid configuration for a volumetric mesh generator.

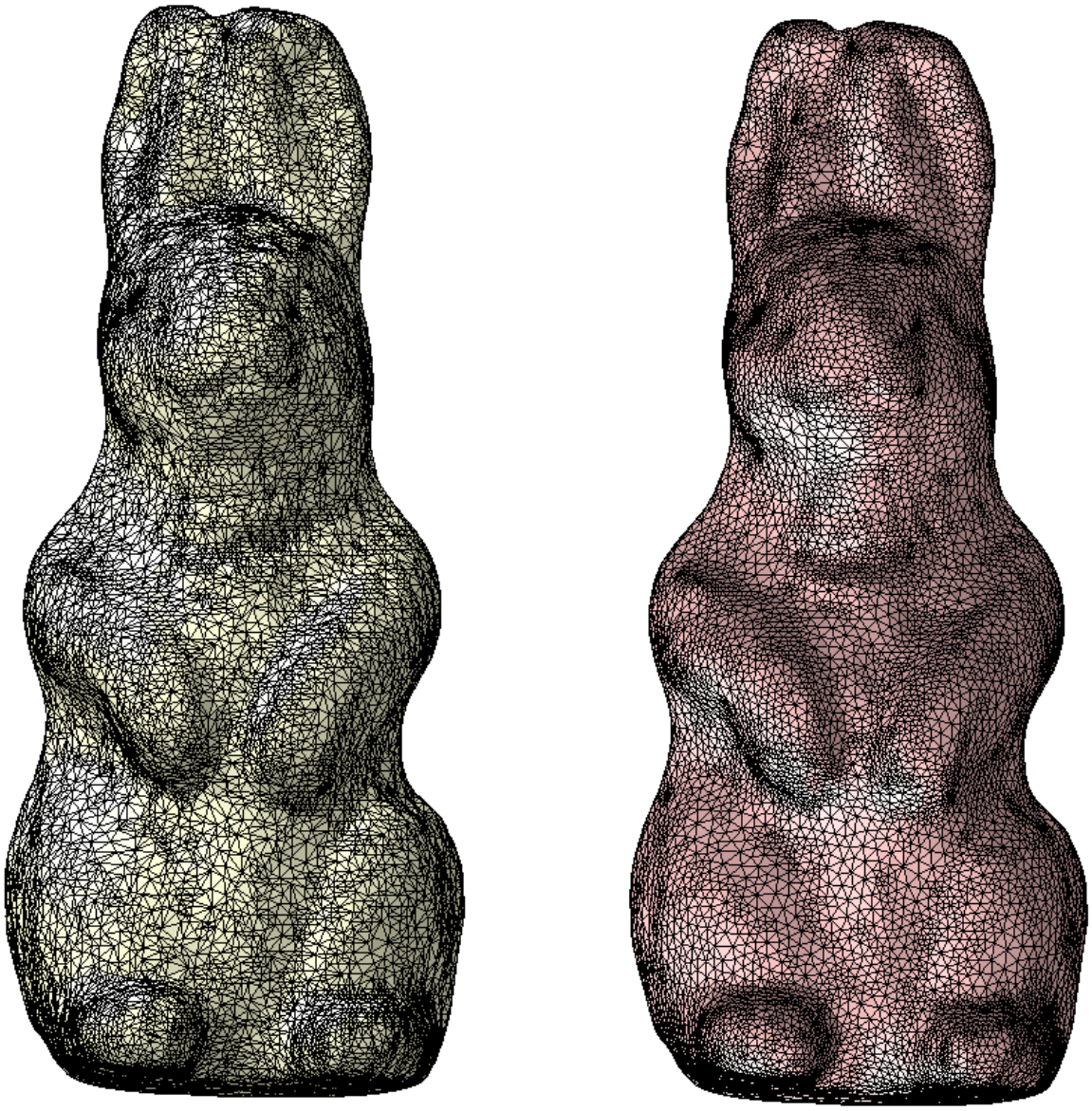


(a) Initial mesh



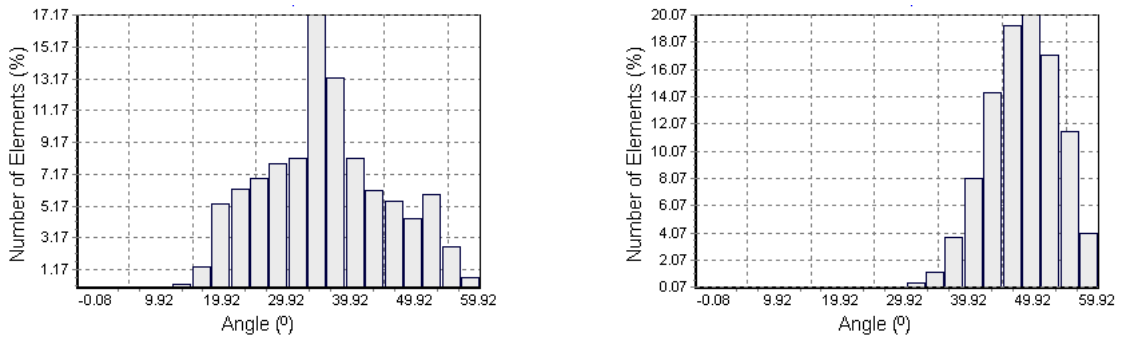
(b) Remeshed Model

Figure 5.18: Volumetric meshes of the asteroid model using as input the surface meshes of Figure 5.15



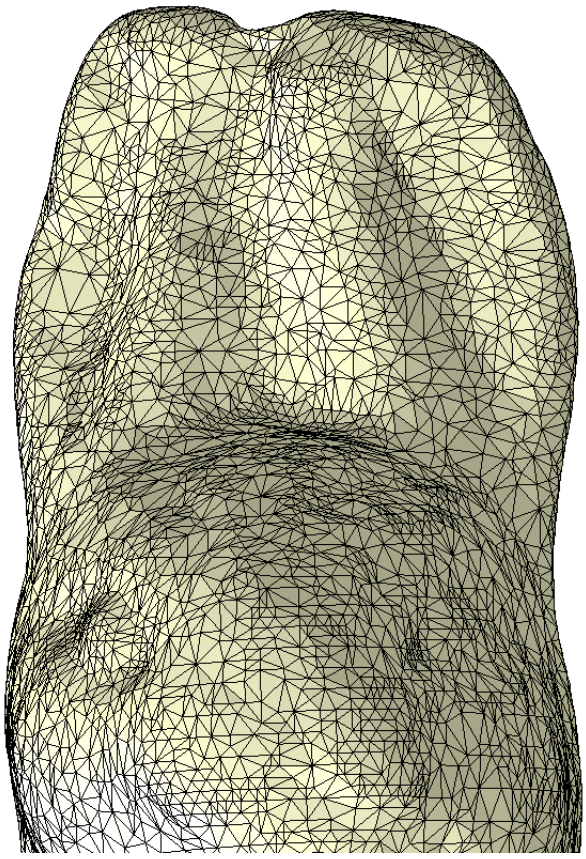
(a) Initial surface mesh (b) Remeshed Model

Figure 5.19: Surface meshes of the rabbit model

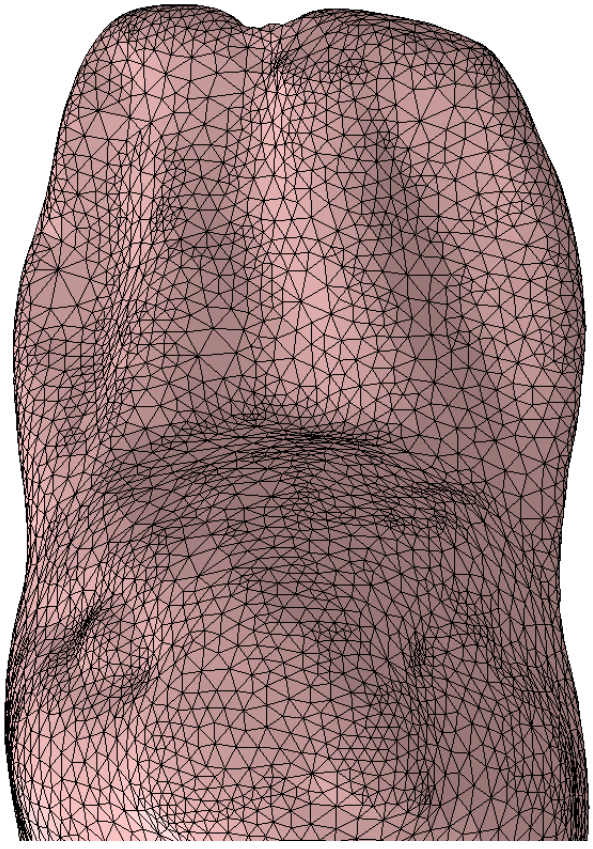


(a) Initial surface mesh (b) Remeshed Model

Figure 5.20: Surface mesh quality graphs considering the minimum angle measure for the rabbit model



(a) Initial surface mesh

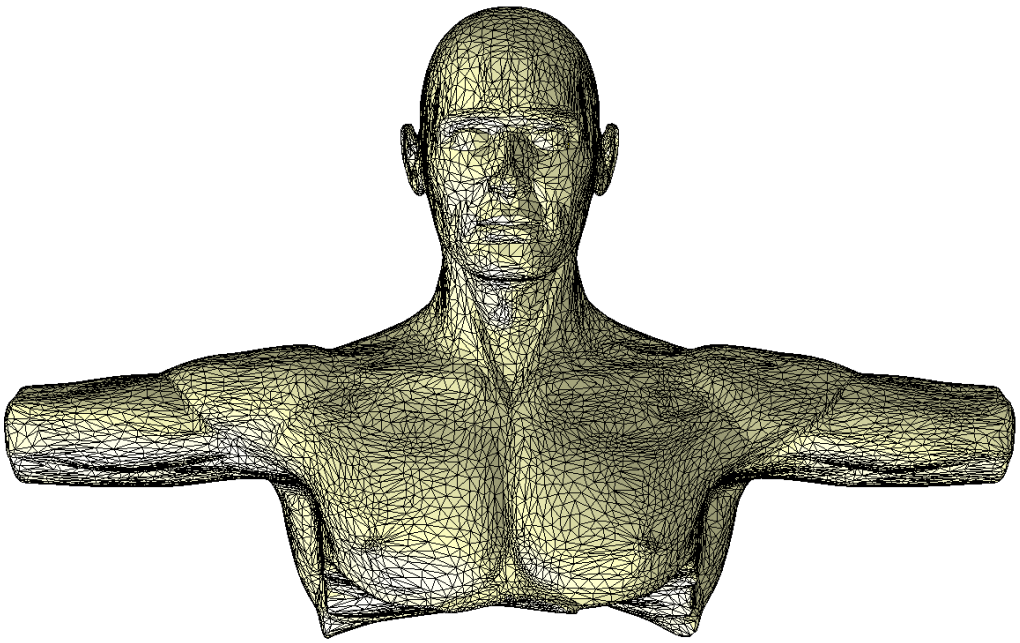


(b) Remeshed Model

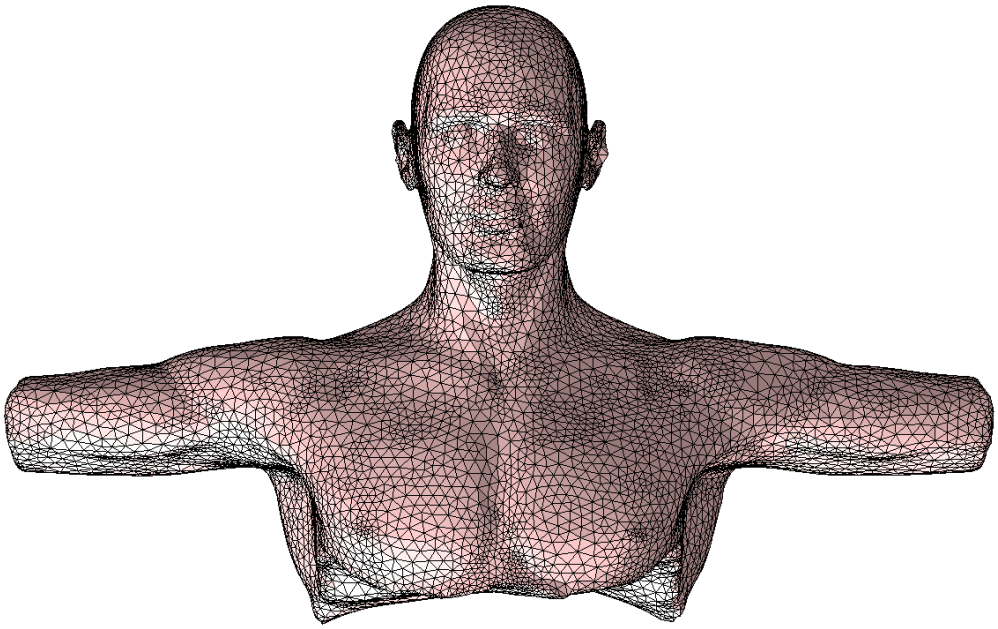
Figure 5.21: Zoom in the ears of the rabbit model presented in Figure 5.19

Figure 5.22: A cut in the volumetric mesh of the remeshed rabbit model (Figure 5.19(b))

Figure 5.23 shows the human torso model before and after the application of the remeshing process. Figures 5.25(a) and 5.25(b) present the distribution of the minimum angles of meshes in Figures 5.23(a) and 5.23(b), respectively. Figure 5.24 illustrates a zoom in the human model faces.

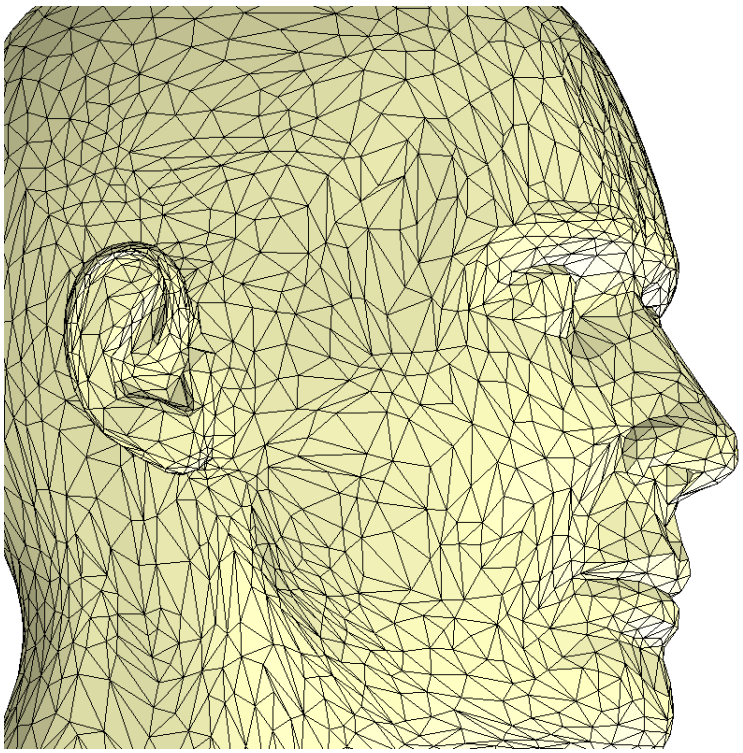


(a) Initial surface mesh

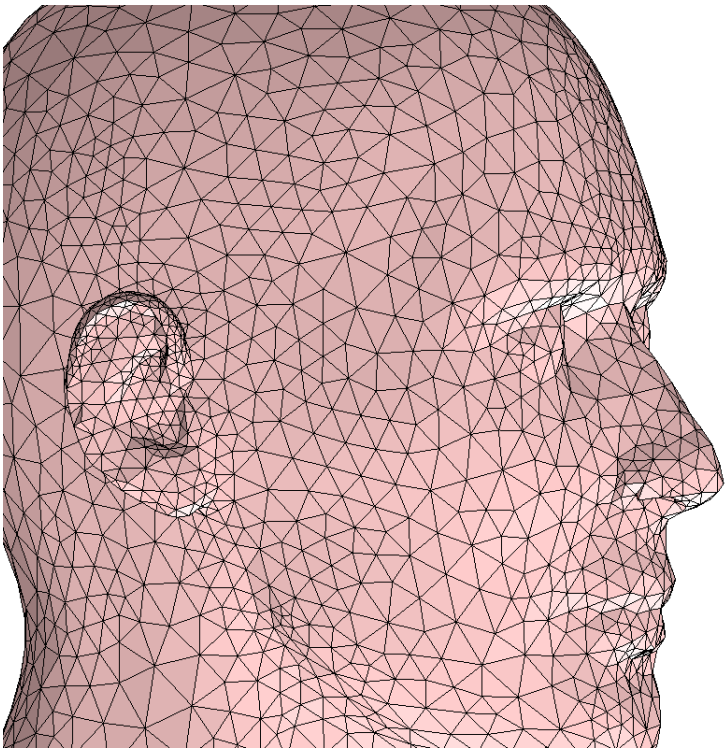


(b) Remeshed Model

Figure 5.23: Surface meshes of the human torso model

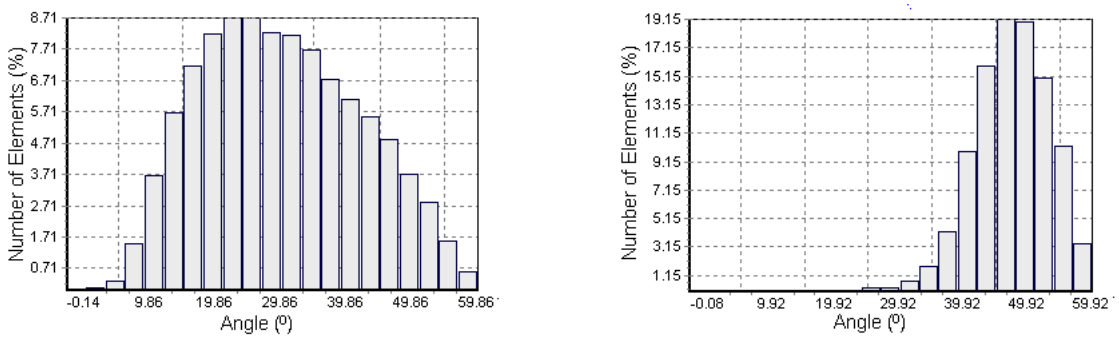


(a) Initial surface mesh



(b) Remeshed Model

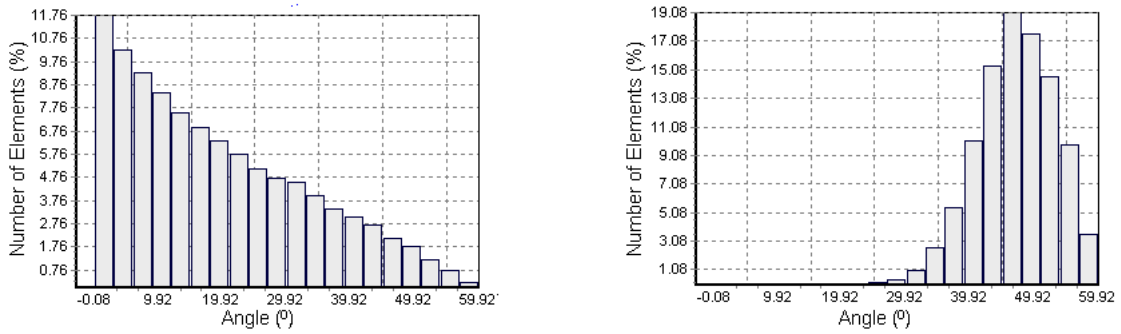
Figure 5.24: Zoom in the Face of the human torso model presented in 5.23



(a) Initial surface mesh (b) Remeshed Model

Figure 5.25: Surface mesh quality graphs considering the minimum angle measure for the human torso model

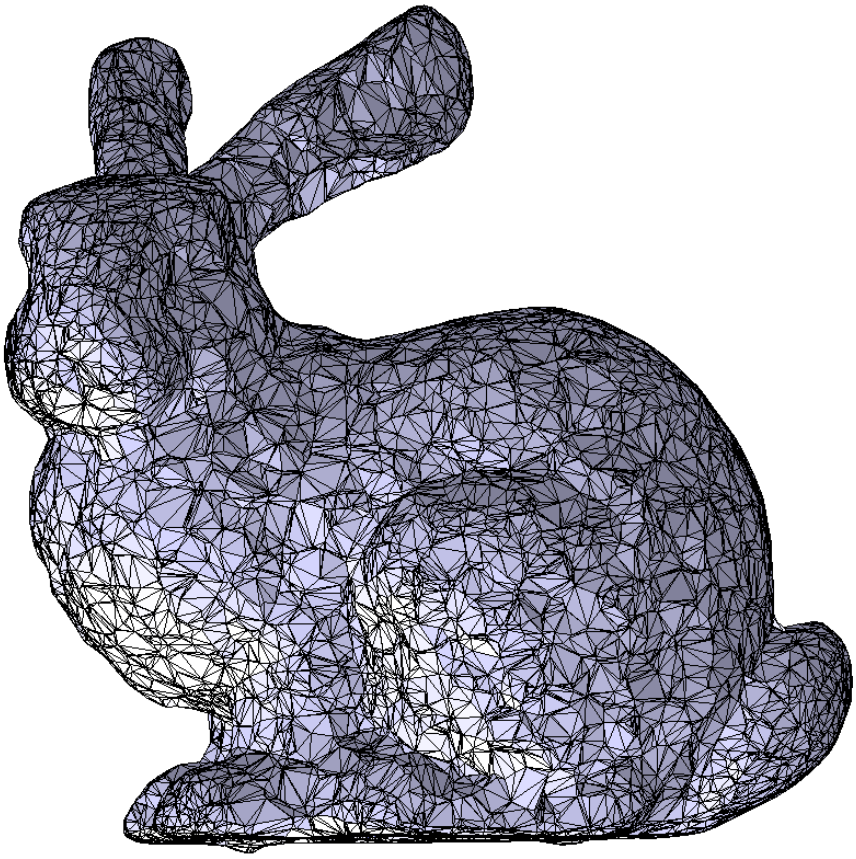
Figure 5.27 illustrates the Stanford bunny model before and after the improvement of its surface mesh. Figure 5.26 shows the corresponding graphs for the distribution of the minimum angles for both meshes.



(a) Initial surface mesh (b) Remeshed Model

Figure 5.26: Surface mesh quality graphs considering the minimum angle measure for the Stanford bunny model

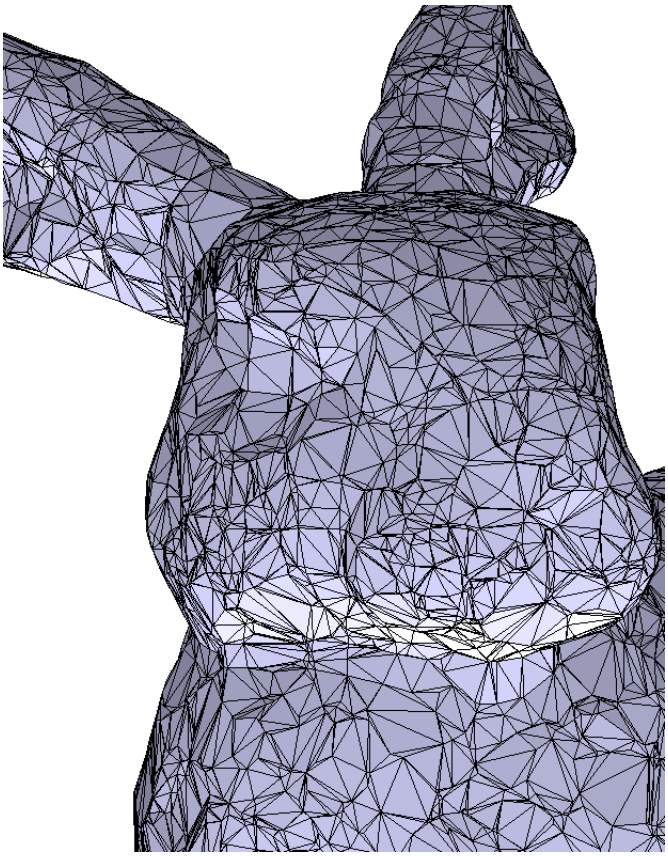
To better view the surface mesh of the Stanford bunny before and after the remeshing process, Figure 5.28 enlarges the faces of the meshes in Figure 5.27.



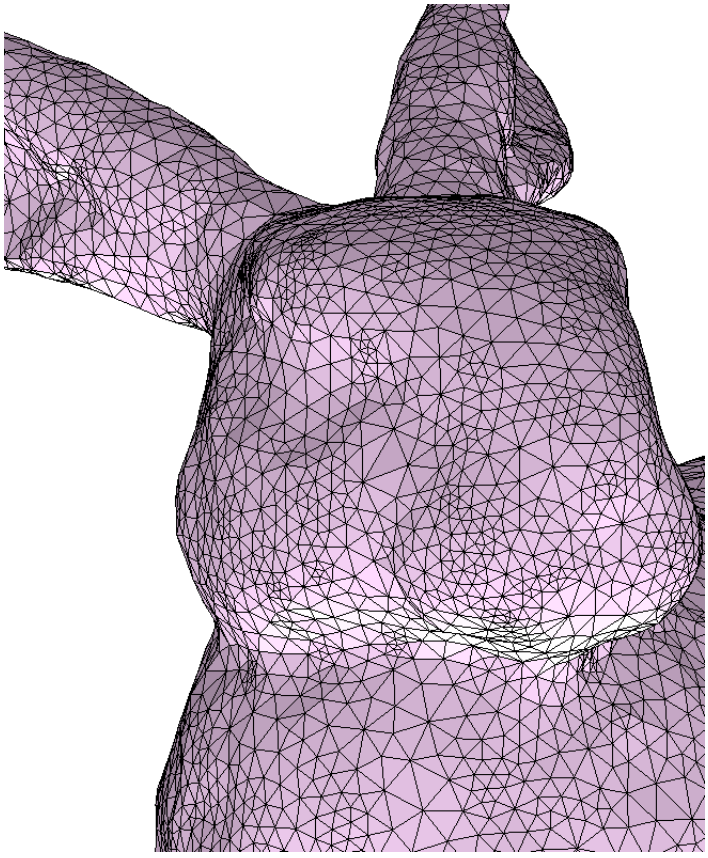
(a) Initial surface mesh

(b) Remeshed Model

Figure 5.27: Surface meshes of the Stanford bunny model



(a) Initial surface mesh



(b) Remeshed Model

Figure 5.28: Zoom in the face of the Stanford bunny model presented in 5.27

5.3 Discussion

In all Figures of section 5.2, it is easy to see that regions with bigger curvature have more elements than the regions with low curvature degree. As discussed in section 4.2.2, the deviation value is evaluated as an average of the deviation between the mesh faces barycenters and the associated smooth approximation. This value is increased or decreased by a factor that is inversely proportional to the local curvature of the surface model. If the user was able to set the expected deviation, the remeshing process could be used to decimate the mesh, reducing more the number of vertices and degrading the geometric characteristics; or it could be used to enrich the initial surface mesh, increasing the model vertex sampling rate.

The results showed significant improvements on surface mesh quality of models generated by reconstruction algorithms. Unfortunately, the generation of the volumetric meshes of the reconstructed models was not possible for some of them. We generated them only for the asteroid model (Figure 5.18) and for the rabbit model (Figure 5.22). Due to the lack of free memory in the system, the volumetric mesh generated for the human torso and Stanford bunny were not evaluated. The B-rep data structures for these models are very big and demand a lot of memory, which makes the evaluation and visualization of the volumetric meshes impossible for models with a very large number of elements.

The models generated by Boolean and assembly operations preserve the sharp features, due to the degree of freedom that the vertices receive during their creation, as the cone example illustrated. However, in the reconstructed model all the vertices can move in any direction on the surface and the sharp features are smoothed, as one can see in the asteroid model (Figure 5.17) and human torso (Figure 5.23). The remeshed models are smoother than the corresponding initial meshes. This happen due to our consideration that the points used in the model reconstruction were sampled from a smooth surface, as stated in section 2.3.3.3, and we approximate the model by a set of smooth patches.

The new remeshing method showed to be appropriated to receive the resulting meshes from the application of the Boolean and assembly operations in solid modelers or from surface reconstruction methods. It proved to be effective in the improvement of surface meshes with any degree of initial quality. The resultant meshes have its surface meshes in agreement with the initial model geometric characteristics and the requirements for a good finite element volumetric mesh generation.

Chapter

6

Conclusions

The finite element method is a good choice for solving partial differential equation over complex domains. FEM numerically approximate the solution of linear or nonlinear PDE by replacing the continuous system of equations with a finite number of coupled linear or nonlinear algebraic equations. Then, the problem domain must be partioned into small pieces of simple shape, constructing a finite element mesh. This mesh must conform the domain of study and its elements should satisfy constraints on size and shape.

Simplicial meshes (triangular and tetrahedral) are one of the most popular representations for models with complex geometric shapes. They can give good surface approximation to complex models due to their flexibility.

An important goal of mesh generation is that the finite element mesh should be formed by relatively well shaped elements. Elements with small angles may degrade the quality of the numerical solution, because they can make the system of algebraic equations ill-conditioned, which compromises the solution accuracy.

The finite element volumetric mesh used in the finite element analysis is directly obtained from the surface mesh. The vertices, edges, faces, slits and holes from the input surface mesh are part of the resulting volumetric mesh. The volumetric mesh generators can try to improve their results by inserting new points on the facets, but they must not remove any of the input geometric entities on the received surface mesh. Then, if the

surface mesh quality is poor, the volumetric mesh can not be generated or a poor quality volumetric mesh is obtained.

The surface meshes generated by methods like the sphere and swept primitives discretization, usually, respect all the quality requirements, such as regularity, connectivity and have its elements internal angles relatively close to 60°. However, these methods produce only a small set of simple models, like spheres, cylinders and torus. Aiming to generate models with high complexity degree, two options are frequently chosen: the application of the Boolean and assembly operations over predefined models; or the generation of models from an acquisition process. Unfortunately, both methods often generate meshes with badly shaped elements.

The reconstruction algorithms, responsible for processing the cloud of unorganized points and turning it into a triangulated surface, do not care about the element shape quality or their connections. Their responsibilities are to guarantee correct geometry, topology and features. As a consequence, the resultant mesh is usually composed by many elements with acute angles (see Figure 1.2).

The same problem of elements with low quality occurs in the resultant meshes after the application of the Boolean and assembly operations over simpler models. They frequently contains a large number of thin elements (see Figure 1.1). As it was discussed in section 2.2.4, the badly shaped triangles appear during the operations application, because many triangles from the intersection region are split into degenerate ones. Each triangle from one object can be intersected by more than one triangle in the other object. The small and badly shaped triangles appear when the intersection points and edges are included in the meshes of both regions and the triangulation is regenerated. The new triangles form the resulting mesh.

On the other hand, as already stated, surface meshes that maximize internal angles are a necessary condition for the generation of high quality volumetric meshes. In this context, improving the surface mesh quality is critical for the generation of a good volumetric mesh suitable for use in electromagnetic simulations by the finite element method.

To improve the surface mesh quality, different post-processing methods can be used, such as surface smoothing, cleaning-up, refining, adaptive and others. The smoothing methods do not change the elements connectivity, but can give good results when the

elements distribution is not too degraded. The mesh vertices are just relocated to improve elements quality. The clean-up methods are able to improve element connectivities, but usually they are used in conjunction with smoothing. The refinement methods, frequently, are used as part of an adaptive solution process. None of these methods, by themselves, are able to improve the kind of surface meshes we have.

The mesh adaptation process turns out to be a very interesting approach. The adaptive methods combine the smoothing, cleaning-up and refining methods to guarantee a better resultant mesh. We presented and discussed two adaptive remeshing methods. Both methods applies series of local mesh modification to improve the mesh. The Frey's method approximates the model surface by a quadric surface to guarantee the model geometric characteristics. Firstly, the geometric surface mesh is extracted from an initial triangulation with bounded distance. Then, a geometric support is built to govern the mesh modification operations during the construction of the normalized unit mesh. This method input is considered to be a large triangular mesh and the method to evaluate the approximation is the same for the entire model.

Surarzhsky and Gotsman's method is based on Frey's work with some strategies alterations. Its input is a large triangular mesh that represents a C^1-continuous surface that is approximated by a set of triangular cubic Bézier patches. The mesh adaptation process alternates the application of operators for area equalization and angle improvements. The regularization of the connectivity is realized at the end.

Both methods are applied only for oversampled triangular meshes representing smooth surfaces. Their works do not guarantee improvement on meshes with low element density or meshes of models composed by curved and planar areas. All the faces are always approximated in the same way, quadric surface approximation for Frey´s method and triangular cubic Bézier patches for Surazhsky and Gotsman´s method.

The resultant meshes of the Boolean and assembly operations application over simpler models are usually represented by a scarce data set; and the meshes generated from surface reconstruction often are a dense data set. The meshes also can have planar facets that need to remain as so when the mesh quality is being improved. In order to address both cases indistinctly, we proposed the combination of two approaches to improve the surface mesh quality: i) the modification of the boundary evaluation process of the Boolean and assembly operations application to avoid the planar facets triangulation, allowing the

surface mesh to be a polygonal mesh; and ii) a remeshing method driven by smooth surface approximation of mesh nodes.

The volumetric mesh generators can receive polygonal meshes as input to describe the model geometry. The higher is the degree of freedom of the volumetric mesh generator, the better is the quality of the resulting mesh. Then, an easy way to increase this degree of freedom is to avoid the planar facet triangulation during the boundary evaluation process, which is our first proposed approach. This approach consists, mainly, in the modification of the primitive mesh generation and intersecting process steps of the boundary evaluation to work with polygons, instead of working only with triangles. Then, some edges that do not add any additional geometric information to the model are suppressed from the resultant mesh.

When the planar facet triangulation is avoided, volumetric meshes for different 3D models were obtained with better quality, smaller number of elements, better memory usage and processing time. The processing time reduced for the volumetric mesh generation and also for the Boolean and assembly operations application. The new approach also made possible the modeling of more complex geometries for which, with the previous system, neither the volumetric meshes nor even their surface meshes could be generated. However, this idea guarantees improvements only for the volumetric meshes with planar facets.

Since the majority of the models are composed by planar facets and curved facets, which are still approximated by triangular faces, a second approach to improve the mesh is proposed. It is more general than the first and it is based on the works presented in chapter 3 to improve surface meshes independent of their source. Our remeshing scheme performs series of local mesh modifications driven by a smooth approximation of the model surface. We also proposed a new approach to evaluate the model surface approximation considering the mesh nodes. This method is able to give good approximation for models generated by application of Boolean and assembly operation over 3D primitives and models reconstructed from a set of points. It generates a set of smooth patches, where each mesh face has a corresponding patch associated to it. The patches are generated using the face vertices and the vertices around it. The approximation of the model as a patches set and the use of the face vertices and its neighbors vertices reduces the approximation errors in the resulting model surface.

The surfaces meshes had their minimum angle and shortest edge increased, after the application of our remeshing process during the Boolean or assembly operations evaluation or to the overall mesh. The geometric approximation in the intersection areas were preserved, because the method take advantage of using the smooth approximation generated before the step of removing the undesired elements in the boundary evaluation. When the model is fully remeshed there were even more improvements on the mesh.

The condition number associated to the stiffness matrix were improved when the quality of the finite element mesh increased. This result proves that an improvement on the shape of the finite elements increases the accuracy of the numerical solutions generated by FEM, as discussed in 2.4.

For the models generated by reconstruction algorithms, the results showed significant improvements on the surface mesh quality. The number of elements with minimum angle lower than 30° considerably decreased and the number of elements with minimum angle higher than 40° increased a lot from the initial mesh to the remeshed models. The geometric characteristics of all models were also preserved.

The improvements on the volumetric finite element mesh were very expressive. They start from making the volumetric mesh generation possible, improvements on the radius-edge ratio metric and also an expressive reduction on the mesh elements number.

The results also illustrated that regions with bigger curvature have more elements than the regions with low curvature degree. The sharp features were preserved in the models generated by Boolean and assembly operations, due to the degree of freedom that the vertices receive during its creation. The reconstructed model had their sharp features a bit smoothed.

We have shown that our remeshing method is appropriated to receive the resulting meshes from the application of the Boolean and assembly operations in solid modelers or from surface reconstruction algorithms. It proved to be effective in the improvement of surface meshes with any degree of initial quality. The resultant meshes have their surface meshes in agreement with the initial model geometric characteristics and the requirements for electromagnetic simulation.

6.1 Summary of the Accomplished Work

We have proposed a novel remeshing scheme to improve surface meshes independent of their origins. More specifically, the major contribution of our work is the model surface approximation by smooth surface patches, which drives the local mesh modification operators during the adaptive process of mesh improvement.

During this work development, some components of independent interest in the scientific community were produced. They can be summarized as follows:

- Identification of the origin of the surface mesh elements lack of quality;
- Modification of the boundary evaluation steps of primitive mesh generation and intersecting process to avoid the planar facet triangulation and work with polygonal elements;
- Evaluation of a surface approximation suitable for models generated by the Boolean and assembly operations applied to simple primitives and models obtained from reconstructing algorithms, such as scanning devices;
- Implementation of the four local mesh modification operators, edge-collapsing, edge-swapping, edge splitting and vertex relocation, driven by the surface approximation of the model surface, using the Euler operators to modify the B-rep representation;
- Development of the surface mesh adaptation scheme based on the implemented local mesh modification operators;
- Implementation of functions to build the B-rep representation of genus 0 models stored in different file formats, such as:
  - PLY (Polygon File Format): describes an object as a collection of vertices, faces and other elements, along with properties such as color and normal direction that can be attached to these elements;
  - OBJ (3D Object Format): it supports both polygonal objects and free-form objects. Polygonal geometry uses points, lines, and faces to define objects while free-form geometry uses curves and surfaces;

– OFF (Object File Format): it also describes an object as a collection of vertices, faces and other elements.

The following publications were generated based on the results that we have obtained during this thesis development:

1. Nunes, C. R. S., Mesquita, R. C., Ribeiro, T. R., Chaves, R. D. and Lowther, D. A., "Building finite-element meshes from reconstructed surfaces", *Submitted to The 16th Conference on The Computation of Electromagnetic Fields*, Aachen, Germany, June 2007.

2. Nunes, C. R. S., Mesquita, R. C. and Lowther, D. A., "Remeshing driven by smooth surface approximation of mesh nodes", *IEEE Transactions on Magnetics*, to appear, April 2007.

3. Nunes, C. R. S., Mesquita, R. C. and Lowther, D. A., "Refinamento da malha superficial baseado na aproximação suave da superfície do modelo", in: *Anais do VII Congresso Brasileiro de Eletromagnetismo*, Belo Horizonte, MG, Brazil, August 2006.

4. Nunes, C. R. S., Mesquita, R. C., Lowther, D. A. and Toledo, R. G., "Remeshing driven by smooth surface approximation of mesh nodes", in: *The 12th Biennial IEEE Conference on Electromagnetic Field Computation*, Miami, Florida, USA, digest CD, April 2006.

5. Nunes, C. R. S., Mesquita, R. C., Lowther, D. A. and Terra, F. M., "Volumetric mesh progress by improving the surface mesh quality", in: *Proceedings of The 15th Conference on The Computation Of Electromagnetic Fields*, vol. I, pp. 132–133, Shenyang, Liaoning, China, June 2005.

6. Nunes, C. R. S., Mesquita, R. C., "Numerical robustness in a solid modeler applied to electromagnetic problems", in: *Anais do XXIV Congresso Ibero Latino Americano de Métodos Computacionais em Engenharia*, Ouro Preto, MG, Brazil, October 2003.

7. Nunes, C. R. S., Mesquita, R. C., Magalhães, A. L. C. C, Mol, C. L. L., Samora, H. F. M. and Falqueto, T. S., "Implementation of boolean and assembly operations in a solid modeler", in: *Proceedings of The 14th Conference on The Computation Of*

Electromagnetic Fields, vol. III, pp. 90–91, Saratoga Springs, New York, USA, July 2003.

8. Nunes, C. R. S., Mesquita, R. C., Magalhães, A. L. C. C., Mol, C. L. L., Samora, H. F. M., Falqueto, T. S., Bastos, J. P. A., "Implementação das operações booleanas e de montagem em um modelador de sólidos", in: *Anais do V Congresso Brasileiro de Eletromagnetismo*, Gramado, RS, Brazil, November 2002.

The first paper presents our remeshing method being used to improve the mesh quality of models generated by surface reconstruction process.

The next three papers introduce the remeshing method driven by the smooth surface approximation of mesh nodes. They present the improvements in the surface mesh and the volumetric mesh of models generated by Boolean and assembly operations, when our mesh post-processing method is applied.

The fifth paper presents the method of avoiding the planar facets triangulation to improve the surface mesh and, consequently, the volumetric mesh of models generated by Boolean and assembly operations.

At the beginning of this thesis, we started working on precision and numerical robustness to improve the Boolean and assembly operations applications. This idea is presented in the sixth paper. The process of coupling a precision library to the GSM was started, but some problems were found. The GSM works evaluating and comparing very huge expressions, that are result of many calculations and transformations. The evaluation of these huge expressions using precision libraries became very slow and inviable.

The last two papers present the implementation steps of Boolean and assembly operations, which are results of my master´s work. The implementation of Boolean and the assembly operations was not an easy task, but it made possible the creation of complex objects using any kind of preexistent model, increasing the GSM descriptive power. In particular, the assembly operation, which, to our knowledge, is an exclusive feature of GSM, improves the representation of electromagnetic problems, making their description much easier than using other solid modelers.

6.2 Future Work

Following the investigations described in this thesis, a number of projects could be taken up. Some of them are:

- Implementation of an algorithm to choose the next edge to be modified by the local mesh operators, instead of testing all edges from the edges set;
- Make possible to the user to set the tolerance and use the remeshing method to enrich or simplify a given surface mesh;
- Identify and preserve sharp features in reconstructed models;
- Limit the area for the smooth patches construction, instead of limiting the number of points. This would make the approximation evaluation more accurate;
- Build the B-rep representation of reconstructed models with arbitrary genus.

Bibliography

[AIM06] AIM@SHAPE, "Aim@shape shape repository v3.0", `http://shapes.aim-at-shape.net/index.php`, site visited in December 2006, 2006.

[Ame99] Amenta, N. and Bern, M., "Surface reconstruction by voronoi filtering", *Discrete and Computational Geometry*, vol. 22, pp. 481–504, 1999.

[Ame01] Amenta, N., Choi, S. and Kolluri, R., "The power crust, unions of balls, and the medial axis transform", *Computational Geometry: Theory and Applications*, vol. 19(2-3), pp. 127–153, `http://www.cs.utexas.edu/users/amenta/powercrust/`, 2001.

[Arb90] Arbab, F., "Set models and boolean operations for solids and assemblies", *IEEE Computer Graphics and Applications*, vol. 10(6), pp. 76–86, November 1990.

[Ber99] Bern, M. W. and Plassmann, P., *Mesh Generation*, chap. 6, Elsevier Science, 1999.

[Fre98] Frey, P. and Borouchaki, H., "Geometric surface mesh optimization", in: *Computing and Visualization in Science*, vol. 1, pp. 113–121, Springer-Verlag, 1998.

[Fre00] Frey, P. J., "About surface remeshing", in: *Proceedings of the 9th International Meshing Roundtable*, pp. 123–136, `http://www.andrew.cmu.edu/user/sowen/abstracts/Fr739.html`, 2000.

[Fre05] Freire, J. A., *Reconstrução de Superfície a Partir de um Conjunto Não-Organizado de Pontos*, Master's thesis, Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Minas Gerais, Belo Horizonte, Outubro 2005.

[Fri72] Fried, I., "Condition of finite element matrices generated from nonuniform meshes", *AIAA Journal*, vol. 10(2), pp. 219–221, February 1972.

[Goi04] Gois, J. P., *Reconstrução de superfícies a partir de nuvens de pontos*, Master's thesis, Instituto de Ciências Matemáticas e de Computação - ICMC-USP, São Carlos, Abril 2004.

[Hrá03] Hrádek, J., *Methods of Surface Reconstruction from Scattered Data*, Ph.D. thesis, University of West Bohemia in Pilsen, Department of Computer Science and Engineering, Czech Republic, technical Report No. DCSE/TR-2003-02, March 2003.

[Hui92] Hui, K. C. and Tan, S. T., "Construction of a hybrid sweep-csg modeler: The sweep-csg representation", *Engineering with Computers*, vol. 8(2), pp. 101–119, March 1992.

[Ida97] Ida, N. and Bastos, J. P. A., *Electromagnetics and Calculation of Fields*, Springer-Verlag, New York, 2nd ed., 1997.

[Jin93] Jin, J., *The Finite Element Method in the Electromagnetics*, John Wiley & Sons, Inc., New York, USA, 1993.

[Kuo05] Kuo, C.-C. and Yau, H.-T., "A delaunay-based region-growing approach to surface reconstruction from unorganized points", *Computer-Aided Design*, vol. 37(8), pp. 825–835, 2005.

[Mag98] Magalhães, A. L. C. C. and Mesquita, R. C., "Requirements for a solid modeler coupled to a finite element mesh generator", *IEEE Transactions on Magnetics*, pp. 3347–3350, September 1998.

[Mag99] Magalhães, A. L. C. C., Mesquita, R. C. and Shimizu, C. A., "An strategy to generate the finite element mesh in defined solids by sweep - Uma estratégia para gerar malha de elementos finitos em sólidos definidos por varredura", in: *XX Congresso Ibero Latino Americano de Métodos Computacionais em Engenharia*, p. 272, 1999.

[Mag00a] Magalhães, A. L. C. C., *Estudo, Projeto e Implementação de um Modelador de Sólidos Voltado para Aplicações em Eletromagnetismo*, Ph.D. thesis, Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Minas Gerais, UFMG, Belo Horizonte, 2000.

[Mag00b] Magalhães, A. L. C. C. and Mesquita, R. C., "Exploring inner boundaries in solid modelers applied to electromagnetic problems", *IEEE Transactions on Magnetics*, pp. 1682–1686, July 2000.

[Mav00] Mavriplis, F., "Cfd in aerospace in the new millenium", *Canadian Aeronautics and Space Journal*, vol. 46(4), pp. 167–176, December 2000.

[Mor85] Mortenson, M. E., *Geometric Modeling*, John Wiley & Sons, Inc., New York, 763 pp., 1985.

[Muu91] Muuss, M. J. and Butler, L. A., "Combinatorial solid geometry, boundary representations and n-manifold geometry", in: *State of the Art in Computer Graphics: Visualization and Modeling*, pp. 185–223, Springer-Verlag, New York, 1991.

[Nun02] Nunes, C. R. S., *Implementação das Operações Booleanas e de Montagem num Modelador de Sólidos Voltado para Aplicações em Eletromagnetismo*, Master's thesis, Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal de Minas Gerais, Belo Horizonte, Março 2002.

[Nun03] Nunes, C. R. S., Mesquita, R. C., de Castro Magalhães, A. L. C., Mol, C. L. L., Samora, H. F. M. and Falqueto, T. S., "Implementation of boolean and assembly operations in a solid modeler", in: *Proceedings Of The 14th Conference On the Computation Of Eletromagnetic Fields*, vol. III, pp. 90–91, Saratoga Springs, New York, USA, July 2003.

[Nun05] Nunes, C. R. S., Mesquita, R. C., Lowther, D. A. and Terra, F. M., "Volumetric mesh progress by improving the surface mesh quality", in: *Proceedings Of The 15th Conference On the Computation Of Eletromagnetic Fields*, vol. I, pp. 132–133, Shenyang, Liaoning, China, June 2005.

[Nun06a] Nunes, C. R. S., Mesquita, R. C. and Lowther, D. A., "Refinamento da malha superficial baseado na aproximação suave da superfície do modelo", in: *7º CBMAG Congresso Brasileiro de Eletromagnetismo*, Belo Horizonte, Minas Gerais, Brasil, Agosto 2006.

[Nun06b] Nunes, C. R. S., Mesquita, R. C., Lowther, D. A. and Toledo, R. G., "Remeshing driven by smooth surface approximation of mesh nodes", in: *The 12th Biennial IEEE Conference on Electromagnetic Field Computation*, Miami, Florida, USA, digest CD, April 2006.

[Nun07] Nunes, C. R. S., Mesquita, R. C. and Lowther, D. A., "Remeshing driven by smooth surface approximation of mesh nodes", *IEEE Transactions on Magnetics*, to appear, April 2007.

[Owe98] Owen, S. J., "A survey of unstructured mesh generation technology", in: *Proceedings of 7th International Meshing Roundtable*, pp. 239–267, `http://www.andrew.cmu.edu/user/sowen/survey/`, October 1998.

[Pot03] Pottmann, H. and Leopoldseder, S., "A concept for parametric surface fitting which avoids the parametrization problem", *Computer Aided Geometric Design*, vol. 20(6), pp. 343–362, 2003.

[Req80] Requicha, A. G., "Representations for rigid solids: Theory, methods and systems", *ACM Computing Surveys*, vol. 12(4), pp. 437–464, December 1980.

[Req85] Requicha, A. G. and Voelcker, H. B., "Boolean operations in solid modeling: Boundary evaluation and merging algorithms", *Proceedings of the IEEE*, vol. 73(1), pp. 30–44, January 1985.

[Riv97] Rivara, M. C., "New longest-edge algorithms for the refinement and/or improvement of unstructured triangulations", in: *International Journal for Numerical Methods in Engineering*, vol. 40, pp. 3313–3324, Wiley, 1997.

[Rup93] Ruppert, J., "A new and simple algorithm for quality 2-dimensional mesh generation", in: *SODA '93: Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*, pp. 83–92, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1993.

[She96] Shewchuk, J. R., "Triangle: Engineering a 2d quality mesh generator and delaunay triangulator", in: *FCRC '96/WACG '96: Selected papers from the Workshop on Applied Computational Geometry, Towards Geometric Engineering*, pp. 203–222, Springer-Verlag, London, UK, 1996.

[She97] Shewchuk, J. R., *Delaunay Refinement Mesh Generation*, Ph.D. thesis, School of Computer Science, Carnegie Mellon University, Philadelphia, Pennsylvania, United States, available as Technical Report CMU-CS-97-137, May 1997.

[She02] Shewchuk, J. R., "What is a good linear element? interpolation, conditioning, and quality measures.", in: *Proceedings of 11th International Meshing Roundtable*, pp. 115–126, `http://www.andrew.cmu.edu/user/sowen/abstracts/Sh878.html`, September 2002.

[Si04] Si, H., "A quality tetrahedral mesh generator and three-dimensional delaunay triangulator", Tech. rep., WIAS, Berlin, Germany, v1.3 User's Manual. Software available at `http://tetgen.berlios.de/`, site visited in December 2006, 2004.

[Sid01] Siddiqui, M. and Sclaroff, S., "Surface reconstruction from multiple views using rational b-splines", Tech. Rep. 2001-009, Image and Video Computing Group, Com-

puter Science Dept., Boston University, http://www.cs.bu.edu/techreports/pdf/2001-009-rational-bspline-surface.pdf, June 2001.

[SIN06] SINTEF, "Least squares approximation of scattered data with b-splines", http://www.sintef.no, site visited in December 2006, 2006.

[Sur03] Surazhsky, V. and Gotsman, C., "Explicit surface remeshing", in: *Proceedings of Eurographics Symposium on Geometry Processing*, pp. 17–28, Aachen, Germany, June 2003.

[Til80] Tilove, R. B., "Set membership classification: A unified approach to geometric intersection problems", *IEEE Transactions on Computers*, pp. 874–883, October 1980.

[Tsu97] Tsukerman, I. and Bossavit, A., "Shape of finite elements and approximation in electromagnetics", in: *Proceedings of the International Symposium ISTET'97*, Palermo, Italy, June 1997.

[Tsu98] Tsukerman, I., "A general accuracy criterion for finite element approximation", *IEEE Transactions on Magnetics*, vol. 34(5), pp. 2425–2428, September 1998.

[Vla01] Vlachos, A., Peters, J., Boyd, C. and Mitchell, J. L., "Curved pn triangles", in: *Proceedings of the 2001 Symposium on Interactive 3D graphics*, pp. 159–166, ACM Press, 2001.

[yH05] Øyvind Hjelle and Dæhlen, M., "Multilevel least squares approximation of scattered data over binary triangulations", *Computing and Visualization in Science*, vol. 8(2), pp. 83–91, http://home.simula.no/~oyvindhj/pub/lsbintrSpringer.pdf, April 2005.

Index

Acquisition Process, 3, 29, 100
Adaptive Process, 59, 70, 83
Avoiding the Planar Facets Triangulation, 71, 85, 118, 122

Boundary Evaluation, 16, 71

CAD, 1, 2, 9, 68
CAE, 1
CAM, 1, 9
Clean-up Methods, 54, 116
  Shape Improvement, 54
  Topological Improvement, 55
Condition Number, 32–35, 95, 98, 119
Continuity
  C^0, 41
  C^1, 42, 64, 68
  C^n, 42
  G^0, 41
  G^1, 41, 59, 64
  Geometric, 41
  Parametric, 41

Delaunay
  Tetrahedralization, 25
  Triangulation, 24, 30
Dihedral Angle, 38, 64
discretization, 1, 8
Divide and Conquer, 16

edge-collapsing, 57, 74, 78
edge-splitting, 57, 74, 79
edge-swapping, 57, 74, 79
Euler Operators, 14, 17, 20, 72, 81

FEM, 1, 2, 7, 8, 15, 32, 34, 115
Frey's Method, 59, 117
  Geometric Support, 61
  Geometric Surface Mesh, 60
  Unit Surface Mesh, 62

Galerkin Formulation, 32
Genus, 63
GSM, 80
  Subsystem
    Interface, 80
    Kernel, 80
    Meshing, 81
    Modeling, 80

Hybrid Methods, 58, 67

Local Feature Size, 59
Local Mesh Modification Operators, 56
LSMG, 81

M-matrix, 35

manifold, 64
Mesh
Generation, 21
Sphere Discretization, 28, 116
Surface, 27
Surface Reconstruction, 29
Swept Primitives Discretization, 29, 116
Two Dimensional, 24
Volumetric, 25
Quality Measures, 31
Aspect Ratio, 36
Aspect ratio, 38
Dihedral Angle, 38
Distortion Metrics, 36
Minimum/Maximum Angle, 36
Radius-edge ratio, 37
Model
Asteroid, 100
Human Torso, 100, 108
Rabbit, 100, 103
Stanford Bunny, 100, 110
Modeling
Geometric, 9
Solid, 10
Surface, 10
Wireframe, 10

New Remeshing Scheme, 83
Algorithm, 77
Approximation of the Model Surface, 75
Implementation Characteristics, 80

OBJ, 120
OFF, 121
Operations
assembly, 3, 7, 11, 15, 70, 71, 73, 78, 85, 116, 118
Boolean, 3, 7, 11, 15, 70, 71, 73, 78, 85, 116, 118
Difference, 15
Intersection, 15
Union, 15
Boolean Evaluation, 20
Elements Classification, 18
Elimination of Undesired Elements, 20
Intersecting Process, 17
Primitive Mesh Generation, 17

PDE, 8, 35
PLC, 25–27, 49
PLSG, 25
PLY, 120
Poisson equation, 32

raycasting, 19
Reconstruction Algorithms, 29, 116
Crust, 31
Delaunay-based, 31
implicit surface, 30
Power Crust, 31
Software, 82
region-growing, 30
Refinement Methods, 55, 116
Edge Bisection, 55
Point Insertion, 56
Remeshing, 5, 58, 59, 65, 68, 69, 71, 73, 81, 89, 98, 101, 114, 118–120, 122, 123
Representation
B-rep, 12
CSG, 11
Sweep, 12

Shape Repository, 100
Smoothing Methods, 52, 116
    Averaging, 53
    Optimization-Based, 53
    Physically-Based, 54
Stiffness Matrix, 32–34, 95, 119
Surazhsky and Gotsman's Method, 63, 117
    Geometric Background, 64
    Remeshing, 65
Surface, 38
    Algebraic Form, 40
    B-Spline, 44, 46, 76
    B-Splines, 38
    Bezier, 38, 42
    Explicit, 40
    Implicit, 39
    Parametric, 39, 40
    patch, 39–41, 43–47
    Representation, 39
    Smooth Approximation, 46
Surface Mesh Post-Processing, 51

TetGen program, 82
Tetrahedron
    Badly-shaped, 37
    Cap, 36
    Needle, 36
    Sliver, 36–38
    Wedge, 36
    Well-shaped, 37
Triangle program, 82

vertex relocation, 57, 74, 79