

6 Conclusões

Este trabalho apresenta um algoritmo para operações booleanas em ambientes de modelagem baseados em representação de fronteira (B-Rep) e com domínio representacional *non-manifold*. A idéia principal é permitir a aplicação das operações booleanas em entidades topológicas de qualquer dimensão, em qualquer número e com qualquer descrição geométrica.

Operações booleanas constituem o paradigma de modelagem de sistemas com representação CSG. Estes sistemas permitem que se combinem objetos sólidos primitivos por meio das operações booleanas para formar objetos sólidos mais complexos. Contudo, as primitivas sólidas possuem formas pré-definidas e sua caracterização geométrica completa depende apenas da instanciação de certos parâmetros pelo usuário do sistema.

Se os objetos sólidos passíveis de serem representados só puderem ser *manifold*, então o conjunto das operações booleanas não é fechado. Isto porque mesmo quando os objetos que estão sendo combinados são *manifold*, não há garantia alguma de que o resultado da operação booleana também será *manifold*. Para que as operações booleanas possam ser inseridas em sistemas de modelagem com domínios *manifold*, é necessário que se adote algum procedimento que evite o subconjunto de resultados *non-manifold* que podem surgir após a aplicação de uma operação booleana sobre dois sólidos. Uma maneira é simplesmente restringir os resultados que podem ser gerados, ignorando-se aqueles que não podem ser representados pelo sistema. Este procedimento pode ser eficiente, mas é também um tanto limitante. Outra maneira é a criação de artifícios que simulem uma topologia *manifold* em resultados que sejam *non-manifold*, como a duplicação de arestas ou vértices *non-manifold*. Este tipo de procedimento é menos restritivo que o anterior, mas pode ocasionar problemas de consistência entre topologia e geometria.

Requicha e Voelcker [11] propuseram uma maneira de se evitar um subconjunto dos resultados *non-manifold* que podem surgir pela aplicação das operações booleanas entre sólidos. São as operações booleanas regularizadas, que permitem que somente resultados com volumes preenchíveis sejam representados. Neste processo, entidades topológicas de mais baixa dimensão

que se encontram soltas ou pendentes são desconsideradas no resultado final. Mesmo assim, o processo de regularização ainda não garante o fechamento das operações booleanas, pois ainda é possível se obter resultados *non-manifold* a partir da aplicação de uma operação booleana regularizada entre dois sólidos.

A utilização de sistemas de modelagem com domínios *non-manifold* permite que se represente qualquer resultado de uma operação booleana, fazendo assim com que elas passem a formar um conjunto fechado de operações. A vantagem da modelagem *non-manifold* está no fato de que ela agrega todos os tipos de representação que a antecederam historicamente, ou seja, a modelagem por arames, a modelagem por superfícies e a modelagem de sólidos.

Em ambientes de modelagem que se baseiam em representação de fronteira (B-Rep), o uso da topologia como elemento-chave é justificado por três razões primordiais: estabilidade do sistema de modelagem, erros numéricos passíveis de existirem na representação geométrica de curvas e superfícies e separação das informações topológicas e geométricas como base para uma maior organização e sistematização na implementação do sistema de modelagem. A topologia de adjacência é a mais utilizada no campo da modelagem geométrica. Ela permite que se tenha acesso à conectividade dos elementos topológicos presentes.

Baseando-se nestas informações, buscou-se elaborar um algoritmo que permitisse a utilização de uma ferramenta típica de sistemas de modelagem CSG em sistemas de modelagem B-Rep. Buscou-se também estabelecer uma maneira completa e consistente de se tratar as operações booleanas em domínios *non-manifold*. Além disso, uma das principais premissas do algoritmo proposto é a utilização de informações essencialmente topológicas, evitando-se ao máximo a dependência da geometria dos modelos durante os passos de modelagem.

Para que as operações booleanas pudessem ser aplicadas não somente entre dois sólidos, mas entre conjuntos de entidades topológicas quaisquer, criou-se o conceito de grupo, que passou a representar um conjunto de entidades que necessitam ser agrupadas para que possam receber o mesmo tratamento, de acordo com o objetivo do usuário do sistema de modelagem. Um grupo pode conter um número qualquer de entidades topológicas, e estas podem ter qualquer dimensão. Deste modo, as operações booleanas não ficam restritas a apenas dois sólidos.

Neste contexto, criou-se uma metodologia de abordagem em relação às estruturas pendentes ou soltas internamente às regiões. Como em diversos problemas de engenharia estruturas internas representam restrições ao domínio destes problemas, influenciando, por exemplo, na geração de malhas de elementos finitos superficiais ou sólidas, buscou-se padronizar um procedimento que respeitasse a existência destas estruturas.

Para que o algoritmo proposto pudesse ser implementado e testado, utilizou-se um modelador geométrico tridimensional pré-existente, o MG, que atende aos requisitos mencionados quanto à forma de representação por fronteira (B-Rep) e domínio representacional *non-manifold*. Este modelador já possuía algumas ferramentas necessárias para garantir que as condições de aplicabilidade do algoritmo pudessem ser respeitadas, como a interseção entre superfícies com geometria arbitrária e o reconhecimento automático de multi-regiões. O enfoque híbrido do modelador, que possui uma estrutura de dados própria que se intercomunica com a estrutura *Radial Edge* de Weiler [17] implementada na biblioteca de classes CGC, permite que se usufrua das funcionalidades desta biblioteca apenas nas etapas de modelagem em que a costura topológica do modelo se faz necessária, como no reconhecimento de regiões.

Contudo, algumas limitações da estrutura de dados do modelador não permitem que as operações booleanas possam ser implementadas exatamente como proposto, além de não permitirem que se possam testar alguns casos patológicos, como a superposição de superfícies.

6.1. Principais contribuições

Operações booleanas são ferramentas de modelagem bastante exploradas. Podem-se encontrar diversos programas e aplicativos que utilizam as operações booleanas como uma forma de se combinar sólidos mais simples para gerar sólidos mais complexos. Mesmo em sistemas de modelagem B-Rep, nos quais a implementação destas operações torna-se algo um pouco mais complicado, as operações booleanas podem existir como forma de modelagem complementar [52].

Este trabalho reúne os conceitos fundamentais de modelagem *non-manifold*, estruturas de dados topológicas e operações booleanas entre sólidos para apresentar um algoritmo robusto e eficiente para se combinar entidades

topológicas quaisquer por meio das operações booleanas. As principais contribuições deste trabalho são:

- Apresentação de um algoritmo para realização de operações booleanas em ambientes de modelagem B-Rep com domínio *non-manifold*.
- Sistematização do tratamento de entidades soltas ou pendentes interiormente a faces e regiões.
- Possibilidade de se aplicar o algoritmo a grupos de entidades topológicas, onde cada grupo pode conter um número qualquer de entidades.
- Tratamento diferenciado para faces, arestas e vértices, garantindo assim a possibilidade de se gerar resultados com qualquer dimensão.
- Utilização de informações essencialmente topológicas no algoritmo, o que garante a sua aplicabilidade em modelos com geometria qualquer.
- Inclusão de mais uma ferramenta de modelagem dentro do modelador MG, tornando-o mais completo e eficiente.

6.2. Sugestões para trabalhos futuros

O algoritmo proposto pode ser considerado como genérico no sentido de não depender do tipo de estrutura de dados utilizado pelo sistema de modelagem para poder ser descrito. Os elementos topológicos que são manipulados - *vértices*, *arestas* e *faces* - são elementos utilizados por qualquer sistema de modelagem baseado em representação de fronteira.

Para que o algoritmo possa ser aplicado, é necessário um pré-processamento das informações de entrada, de forma que as condições de aplicabilidade do algoritmo sejam respeitadas. Isto significa que o sistema de modelagem que utilizar este algoritmo deve possuir ferramentas para o cálculo de interseções entre curvas e superfícies e para o reconhecimento de regiões.

O algoritmo consiste basicamente no armazenamento, classificação e consulta a entidades topológicas destes três tipos, para determinar quais devem permanecer e quais devem ser eliminadas no resultado de uma operação booleana qualquer.

Destas três etapas mais gerais, a única que utiliza informações geométricas do modelo é a classificação das entidades topológicas. Esta etapa requer a chamada de funções para se determinar se um ponto é interior ou

exterior a uma região, e se um ponto pertence ou não a um retalho de superfície. Para se manter a generalidade do algoritmo, entretanto, é necessário que estas funções aceitem como parâmetros de entrada retalhos de superfície com geometria arbitrária (incluindo os que formam as fronteiras das regiões).

Tendo em vista estas informações, uma sugestão seria implementar o algoritmo de operações booleanas proposto como uma biblioteca de classes extensível. Esta biblioteca poderia conter uma única classe, *BoolOp*, responsável pela aplicação das operações booleanas sobre as entidades topológicas. Poderia haver um método responsável pelo armazenamento das informações dos grupos, que seriam passadas para este método, por exemplo, como vetores de informações geométricas sobre as entidades topológicas. A descrição geométrica deveria ser comum para o cliente da biblioteca e para a biblioteca. Isto poderia ser feito através de uma interface comum, baseada, por exemplo, numa representação NURBS [27,28]. A biblioteca poderia conter uma estrutura de dados topológica própria, que utilizasse as informações geométricas recebidas para criar os elementos topológicos. Além disso, métodos *virtuais puros* (métodos de uma classe que não são implementados dentro do escopo desta classe, mas que necessariamente devem ser implementados dentro do escopo das classes derivadas desta) responsáveis pelos algoritmos geométricos de detecção de ponto em região e de detecção de ponto sobre face poderiam ser declarados, para que o cliente da biblioteca pudesse implementar uma classe derivada da classe *BoolOp* onde estes métodos seriam implementados. Os métodos responsáveis pelas operações booleanas retornariam vetores com as informações geométricas das entidades que deveriam ser removidas.

Poderiam ser pesquisados algoritmos de detecção de ponto em região e de ponto sobre retalho de superfície mais eficientes para serem usados na etapa de classificação das entidades topológicas do algoritmo proposto neste trabalho. Dependendo da complexidade do modelo em termos de número de entidades topológicas presentes, esta etapa pode demandar um tempo excessivo.

Também podem ser feitas algumas sugestões relativas ao modelador MG. Quanto à sua estrutura de dados, novas classes e métodos poderiam ser criados para que as informações providas da biblioteca CGC pudessem ser devidamente interpretadas.

Uma classe representando uma casca poderia ser criada, de tal forma que os objetos desta classe possuíssem uma lista de listas de referências para entidades topológicas quaisquer (objetos da classe *Topology*), cada lista representando uma casca de uma determinada região. Poderia haver uma

referência para uma *parentShell*, semelhante ao *parentLoop* existente para um *patch2d*, representando uma eventual casca mais externa que contém aquela casca no seu interior. Para manter a consistência, os objetos da classe *patch3d* possuiriam uma lista de cascas associadas àquele *patch3d*. Isto permitiria que estruturas soltas internamente a uma região, bem como vazios internos, pudessem ser identificados.

Na lista de *patches3d* de um determinado *patch2d*, um mesmo *patch3d* poderia aparecer duas vezes no caso do *patch2d* estar pendente internamente a este *patch3d*. Isto inclusive faz sentido, já que este retalho de superfície é visitado duas vezes quando se percorre a lista de retalhos de superfície associada a esta região na biblioteca CGC. Ou seja, na estrutura de dados da *Radial Edge*, os dois *usos* desta face estão associados a uma mesma casca. Isto permitiria identificar os retalhos de superfície pendentes internamente às regiões.

As arestas pertencentes à fronteira de um determinado *patch2d* formando um *loop* interno conexo e fechado poderiam ser diferenciadas das arestas soltas ou pendentes internamente ao *patch2d*, sendo armazenadas numa outra lista, ou mesmo na lista *_lsegments*, juntamente com outras as arestas da fronteira deste *patch2d*. Poderia haver também uma lista de vértices representando vértices soltos no interior do *patch2d*.

Outra mudança que poderia ser feita é a desvinculação de um *patch2d* das arestas (objetos da classe *Segment*) originalmente utilizadas para gerar aquele *patch2d*. Quando um *patch2d* fosse interceptado por outros que o dividissem em dois ou mais novos *patches2d*, as malhas associadas aos novos *patches2d* poderiam ser re-geradas de forma que dependessem exclusivamente das novas arestas das fronteiras destes *patches2d*.

Além disso, o algoritmo de interseção de superfícies poderia ser estendido de forma que a superposição de faces fosse devidamente tratada. Isto poderia ser feito identificando-se o problema da superposição antes do cálculo da interseção, por meio de testes geométricos, usando-se uma tolerância conveniente para o modelo em estudo. Ao ser solicitado o cálculo da interseção entre as superfícies superpostas, uma malha única seria gerada. Isto seria de grande utilidade na geração de modelos que utilizam somente faces planas, onde esta situação pode ocorrer com mais frequência.