

## Um Algoritmo de Corte para Modelos Sólidos

Marco Rangel, Paulo Bernardes, Luís Marques, Rafael Bidarra,  
José C. Teixeira

**Grupo de Métodos e Sistemas Gráficos**  
Departamento de Matemática, Universidade de Coimbra  
Largo D. Dinis - Apartado 3008  
3000 COIMBRA - PORTUGAL  
Tel: 039 - 28097/8/9  
Fax: 039 - 32568

### Resumo

A operação de corte de modelos sólidos é de grande importância quer na fase de modelação, quer na de análise de modelos quer ainda na de apresentação (gráfica) dos modelos. Esta comunicação é destinada a apresentar um algoritmo eficiente para o corte de modelos sólidos armazenados numa estrutura Brep. Depois de referir sucintamente algumas vantagens do uso de uma ferramenta deste tipo num sistema de modelação sólida, são introduzidos alguns conceitos específicos do modelo computacional BRep (2-Manifold), uma vez que a implementação do algoritmo está vocacionada para o corte de modelos sólidos deste tipo.

É feita uma descrição detalhada das várias fases do algoritmo de corte referindo, sempre que necessário, algumas especificidades do mesmo. Apresentam-se algumas vantagens desta realização relativamente a outras aproximações e a generalidade dos modelos a que se pode aplicar o algoritmo é exemplificada através de figuras de modelos sólidos que foram sujeitos ao processo de corte (em Anexo).

## 1 - Introdução

A introdução de tecnologias CAD/CAM como suporte de actividades de *projecto* e fabrico industrial, tem sido uma constante nas mais diversas áreas. O reconhecimento da importância do uso destas tecnologias no meio industrial é um facto inegável, devido às grandes facilidades por elas proporcionadas, nomeadamente, a diminuição do tempo de desenvolvimento do produto, minimizando significativamente os custos associados a um determinado projecto e permitindo melhorar substancialmente o desempenho e a qualidade do produto final [ENJL-90], [PAGB-88].

Quando se usa um sistema CAD, torna-se necessário representar a forma tri-dimensional do objecto. São portanto necessárias técnicas para construir essas representações - técnicas de modelação geométrica. De entre estas assumem especial importância as técnicas de modelação sólida [REQA-80], [MORM-85]. O algoritmo aqui apresentado destina-se a operar sobre modelos sólidos. No âmbito da modelação sólida, é de salientar a importância de um algoritmo que permita efectuar o corte de modelos segundo um plano de corte definido interactivamente pelo utilizador. Uma ferramenta deste tipo apresenta-se bastante útil, não só na fase de modelação do objecto (podendo ser encarado como uma verdadeira ferramenta de modelação), como numa fase posterior, permitindo a visualização de características dos modelos (exemplo: visualização de secções de corte, técnica largamente utilizada em projectos de engenharia).

No desenvolvimento do algoritmo foi assumida a validade geométrica e topológica do modelo sólido inicial ao qual se pretende aplicar o processo de corte. A informação geométrica e topológica do modelo inicial vai sofrer alterações durante o processo de corte, por meio da criação, alteração e eventual eliminação de vértices, arestas e faces. O fecho do algoritmo é garantido pelo uso intensivo e ordenado dos Operadores de Euler [BAUB-74], sendo assim assegurada a validade de todos os (sub-)modelos sólidos resultantes do corte do modelo inicial, no que se refere à ausência de modelos auto-intersectáveis.

## 2 - A Representação de Sólidos pela Fronteira

De entre as várias representações de modelos sólidos existentes, o algoritmo desenvolvido foi direccionado para uma representação BRep, devido às vantagens por ela oferecidas, nomeadamente, o alto nível de interactividade com o utilizador, permitindo a execução de alterações locais. Este tipo de modelos são normalmente considerados de grande interesse em processos que envolvam saída gráfica, quer na visualização sob a forma de modelos de linhas, quer na aplicação de conhecidos algoritmos de sombreamento para obtenção de imagens foto-realísticas. O cálculo de propriedades integrais em modelos deste tipo é bastante simples, sendo esta mais uma das razões da importância deste algoritmo, uma vez que ele permite ao utilizador determinar áreas ou volumes de uma dada parte do modelo.

A estrutura de dados BRep utilizada é a "*Halfedge*", na qual cada aresta tem associado um par de *halfedges*. Cada um deles representa a orientação em que a aresta é percorrida, quando se efectua um percurso completo e fechado em cada uma das duas faces que ela separa. O percurso em torno de uma face designa-se por "*Loop*".

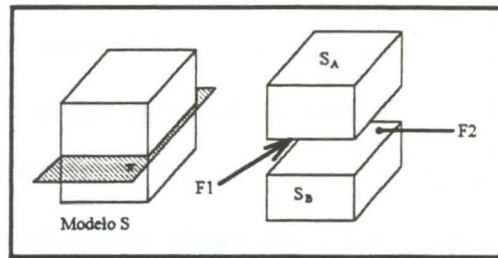
A cada conjunto de faces adjacentes que envolve uma parte do modelo, dá-se o nome de "*Shell*".

## 3 - Descrição do Algoritmo

O algoritmo apresentado tem como finalidade a resolução global, optimizada do problema do corte de modelos sólidos BRep. Como dados do problema temos um modelo sólido  $S$  e um plano de corte  $\pi$ . O objectivo é o de determinar duas famílias de modelos sólidos  $F_A$  e  $F_B$  resultantes do corte do modelo inicial  $S$  pelo plano secante  $\pi$ , sendo o conjunto  $F_A$  definido como o conjunto das *shells* resultantes do processo de corte, que se situam acima do plano  $\pi$  e  $F_B$  o conjunto das *shells* que se encontram abaixo do referido plano. Os modelos poliédricos sobre os

quais vai operar o algoritmo de corte podem ser obtidos por um qualquer modelador sólido BRep.

O esquema global de funcionamento do algoritmo pode ser descrito do seguinte modo: consideremos o modelo S definido na Figura 1 e um plano de corte  $\pi$ , que o secciona. Pretende-se que da aplicação do processo de corte, resultem duas *shells* distintas: a que se situa acima do plano  $\pi$ , que designamos por  $S_A$  e a que se encontra abaixo deste,  $S_B$ . Este resultado é obtido inserindo o par de faces F1 e F2 ao longo da secção poligonal do modelo, originada pelo plano de corte. Uma das faces, F1, fecha  $S_A$  e a outra face, F2, fecha  $S_B$ .



Figural - Corte de um Modelo

De modo a simplificar a abordagem do algoritmo, este foi sub-dividido em várias fases permitindo cada uma delas um tratamento individual, obedecendo a sua execução sucessiva a uma ordem lógica que preserva o interface entre cada fase (Figura 2).

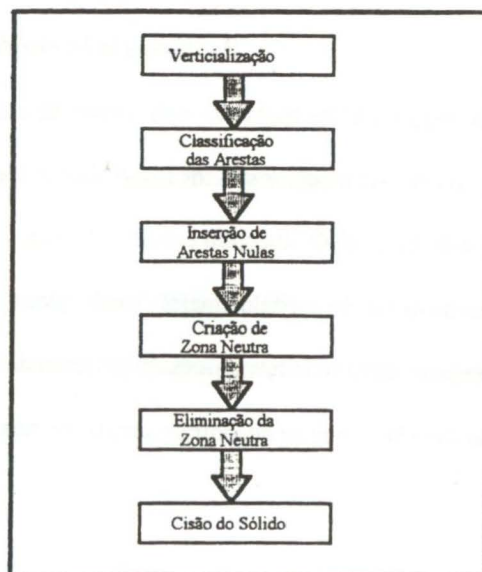


Figura 2 - Fases do Algoritmo de Corte.



O modelo usado para exemplificar o algoritmo de corte é o apresentado na Figura 3 :

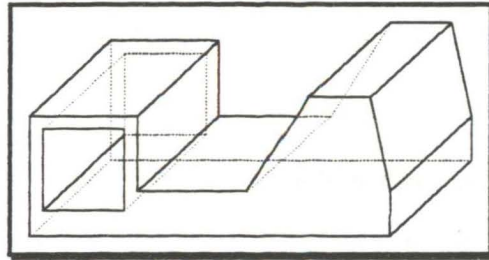


Figura 3 - Modelo base para o Processo de Corte

### 3.1 - Verticalização:

A verticalização consiste na obtenção do conjunto de vértices  $V_{VC}$  coplanares com o plano de corte  $\pi$ , resultantes da intersecção do referido plano com a fronteira 0D (vértices) e 1D (arestas) do modelo. Um caso particular é a situação em que uma aresta é coplanar com  $\pi$ , sendo então considerados apenas os seus vértices e excluindo-se os pontos interiores. Para cada aresta intersectada pelo plano  $\pi$  é calculado o ponto de intersecção, pois ele vai fazer parte do conjunto de vértices desejado (Figura 4).

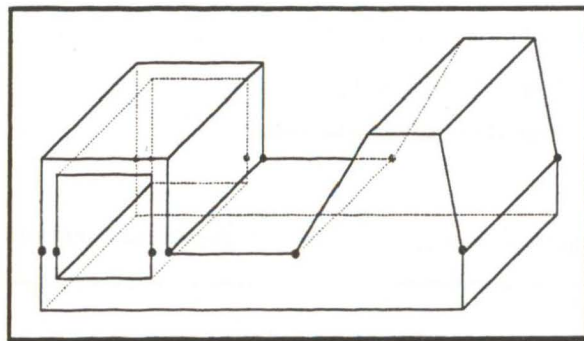


Figura 4 - Verticalização do modelo.

Formalmente temos :

$S \rightarrow$  Modelo;

$\pi \rightarrow$  Plano de Corte;

$V_{vc} \rightarrow$  Conjunto dos Vértices Coplanares ao plano  $\pi$ ;

$E(S) \rightarrow$  Conjunto das arestas do modelo  $S$ ;

$V(S) \rightarrow$  Conjunto de vértices do modelo  $S$ ;

$$V_{vc} = \{ v : v \in (V(S) \cap \pi) \} \cup \{ v' : v' \in (E(S) \cap \pi) \};$$

$e \in E(S)$  então:  $e$  tem como vértices extremos  $V_1$  e  $V_2 : e \langle V_1 V_2 \rangle$

O algoritmo da verticalização é o seguinte :

**Para**  $\forall e \in E : e \langle V_1 V_2 \rangle$

- calcula a distância  $d_1$  ( com sinal ) de  $v_1$  ao plano de corte;
- calcula a distância  $d_2$  ( com sinal ) de  $v_2$  ao plano de corte;
- **Se** (  $d_1 < 0$  e  $d_2 > 0$  ) ou (  $d_1 > 0$  e  $d_2 < 0$  )

**Então** { *aresta não coplanar intersectada* }

- calcula ponto  $V_{int}$ , de intersecção da aresta e com o plano  $\pi$ ;
- MEV ( subdivisão da aresta )<sup>(1)</sup>
- adiciona  $V_{int}$  a  $V_{vc}$ ;

**Senão** {  *$v_1$  ou  $v_2$  é coplanar* }

- **Se**  $d_1 = 0$  **Então** adiciona  $v_1$  a  $V_{vc}$ ;
- **Se**  $d_2 = 0$  **Então** adiciona  $v_2$  a  $V_{vc}$ ;

**FimPara**;

É a partir do conjunto de vértices  $V_{vc}$  que se vai desenrolar a segunda fase do algoritmo.

<sup>1</sup>- Operador de Euler que divide uma aresta em duas por meio de um novo vértice.

### 3.2 - Classificação das Arestas

A segunda fase do algoritmo permite efectuar a classificação das arestas (adjacentes aos vértices anteriormente determinados) em três grupos: arestas situadas acima, abaixo e sobre o plano de corte  $\pi$ .

O grupo das arestas coplanares com  $\pi$  é ainda sujeito a um segundo processo de reclassificação, de forma que à saída desta fase existam apenas dois grupos de arestas: as situadas acima e abaixo do plano de corte. O processo de reclassificação consiste em dividir as arestas coplanares, pelos dois conjuntos anteriores, de modo a que as mesmas sejam consideradas como se estivessem sempre do lado do material do modelo S, evitando assim o aparecimento de arestas remanescentes.

A classificação das arestas pode ser ilustrada pela Figura 5:

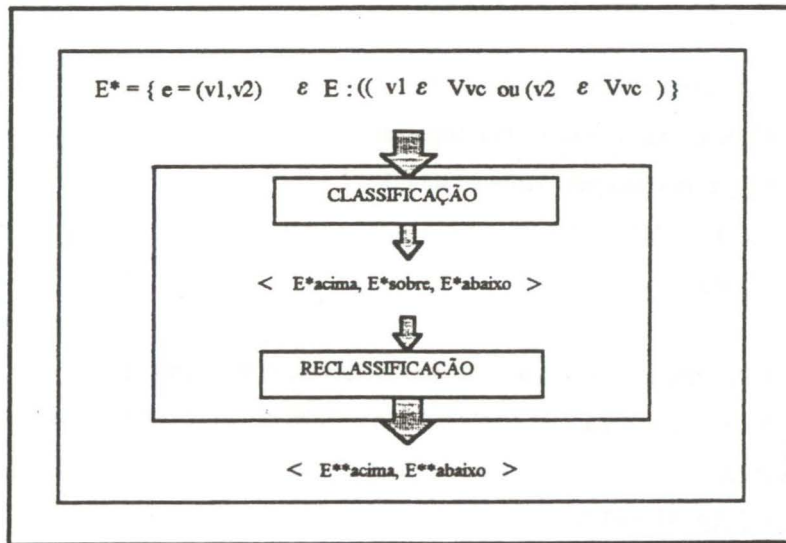


Figura 5 - Classificação das Arestas.

Algoritmo de classificação das arestas:

**Para**  $\forall v \in V_{vc}$  { vértice base }

**Para**  $\forall e \in E: e \langle V V_t \rangle$

- determina vértice terminal  $v_t$ ;
- calcula distância  $d$  ( com sinal ) de  $v_t$  ao plano  $\pi$ ;
- **Caso d Seja**
  - $> 0$  : ACIMA(e);
  - $= 0$  : SOBRE(e);
  - $< 0$  : ABAIXO(e);

**FimCaso;**

**FimPara;**

**FimPara;**

Reclassificação das arestas :

$\forall e \in E^*$  sobre :  $e \langle V V_t \rangle$

- Determina se  $e$  pertence a uma face coplanar com  $\pi$ ;
- **Se sim**

**Então**

- Determina Normal do plano  $\pi$ ;
- Determina Normal da face coplanar;
- **Se** as normais têm o mesmo sentido

**Então** ABAIXO(e)

**Senão** ACIMA(e);

**Senão**

- Determina as outras duas arestas incidentes nesse vértice coplanar;
- Calcula as distâncias com sinal dos vértices extremos de cada uma

dessas arestas;

- **Caso**  $d_1$  e  $d_2$  sejam :
  - $> 0$  : ACIMA(e);
  - $< 0$  : ABAIXO(e)
- Caso contrário : ACIMA(e); (2)

**FimCaso;**

**FimSe;**

<sup>2</sup>- Neste caso poderia ter sido classificada como ACIMA ou ABAIXO. optou-se pela classificação ACIMA.



Os dois conjuntos de arestas obtidas são o contributo da fase de classificação para a fase seguinte.

### 3.3 - Inserção de Arestas Nulas

O objectivo desta fase é a preparação do modelo para a posterior criação da "Zona de Ninguém" ou Zona Neutra, que permite a separação das partes da vizinhança arestal de todos os vértices coplanares com  $\pi$ , ( $V_{VC}$ ) que se situam acima do plano  $\pi$ , daquelas que ocorrem abaixo deste. O objectivo desta fase pode ser visualizado através da Figura 6.

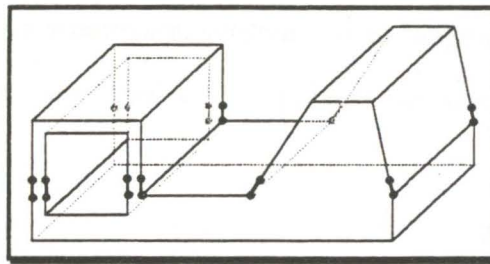


Figura 6 - Inserção das Arestas Nulas.

O algoritmo para a inserção das arestas nulas é o seguinte:

**Para**  $\forall v \in V_{VC}$  **Faz**

• **Se**  $\exists e_i \in E^{**}$  Acima e  $\exists e_j \in E^{**}$  Abaixo

**Então**

- Identifica aresta  $e_j$  Abaixo à cabeça;
- Identifica aresta  $e_i$  Acima à cauda;
- Insere uma aresta nula;

**FimSe;**

**FimPara;**

A identificação das arestas Abaixo à cabeça e Acima à cauda é ilustrada pela Figura 7. A correcta identificação destas duas arestas é crucial, uma vez que essa escolha vai permitir demarcar correctamente, o ciclo de arestas Acima que ficam em torno de um vértice, do ciclo de arestas Abaixo que se devem situar em torno do outro vértice da aresta nula (Figura 8).

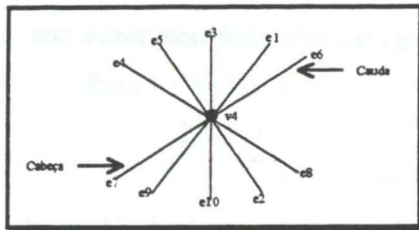


Figura 7 - Ident. da Cabeça e da Cauda

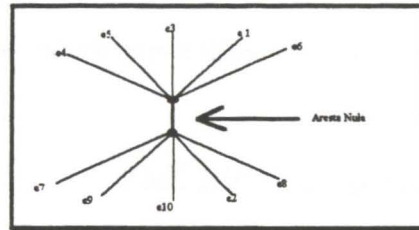


Figura 8 - Inserção da Aresta Nula.

O modo sistemático como é feita a inserção da aresta nula (sempre no sentido da aresta que se situa abaixo do plano, para a aresta que se situa acima deste), permite que numa segunda fase, se possa efectuar a "colagem" de dois modelos anteriormente cortados. O conjunto das arestas nulas inseridas, vai servir como entrada para a fase seguinte do algoritmo.

### 3.4 - Criação da Zona Neutra

Esta fase consiste basicamente, em combinar as arestas nulas, originadas pelo passo anterior, de forma a obter um conjunto de faces nulas. À sequência de faces nulas assim obtidas dá-se o nome de "Zona Neutra". Esta zona vai permitir demarcar a parte do modelo que se situa abaixo do plano de corte, da que se encontra acima deste. Um dos passos desta fase do processo é a da criação dos ciclos de interligação. Um ciclo de interligação é definido como uma sequência finita de arestas nulas, que permitem definir sem ambiguidade uma zona neutra (Figura 9).

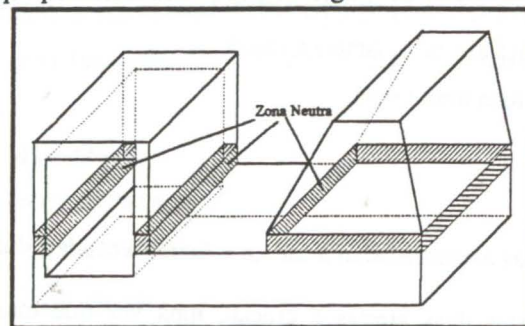


Figura 9 - Criação da zona neutra.

O algoritmo para a criação da zona neutra pode ser descrito do modo seguinte:

- rdenação lexicográfica das arestas nulas;
- **Enquanto** existirem arestas nulas **Faz**
  - Determina um ciclo de interligação;
  - Determina a primeira aresta nula do ciclo de interligação;
  - **Enquanto** não chegar ao fim do ciclo de interligação **Faz**
    - Determina a próxima aresta nula;
    - **Se** as arestas estão na mesma face e no mesmo *loop*
- **Então**
  - MEF (vértices iniciais das arestas nulas);<sup>(3)</sup>
  - MEF (vértices finais das arestas nulas);
- **Senão** {se as arestas estão apenas na mesma face};
  - MEKR (vértices iniciais das arestas nulas);<sup>(4)</sup>
  - MEF (vértices finais das arestas nulas);
- **FimEnquanto**;
- Retira as arestas já utilizadas em ciclos de interligação;
- **FimEnquanto**;

### 3.5 - Eliminação da Zona Neutra

A eliminação da zona neutra criada na fase anterior, é o passo do algoritmo que antecede o corte final do sólido. Nesta fase vai-se proceder à remoção da zona neutra, criada na fase anterior.

A remoção da zona neutra é feita para cada ciclo de interligação, destruindo as arestas nulas e as faces a elas associadas. Em cada ciclo a última face nula não é removida, porque é necessária para a criação de um *loop* interior que coincide com o *loop* contorno da face. A zona neutra eliminada pode ser visualizada na Figura 10.

<sup>3</sup>- Operador de Euler que divide uma face em duas, por meio de uma nova aresta.

<sup>4</sup>- Operador de Euler que permite destruir um *loop* interior por meio de uma nova aresta, que o liga ao *loop* contorno.

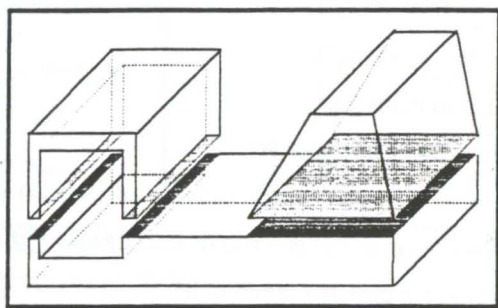


Figura 10 - Eliminação da Zona Neutra.

O algoritmo para a eliminação da zona neutra pode ser apresentado do modo seguinte:

**Para** todos os ciclos de interligação **Faz**

- Determina uma aresta nula do ciclo de interligação a ser analisado;
- **Enquanto** não chegar ao penúltimo elemento do ciclo **Faz**
  - KEF; <sup>(5)</sup>
  - Determina a próxima aresta nula;

**FimEnquanto**;

- KEMR;( para a última aresta nula do ciclo de interligação actual) <sup>(6)</sup>

**FimPara**;

### 3.6 - Cisão do Sólido

A etapa final do algoritmo de corte diz respeito à divisão efectiva do modelo seccionado, em vários modelos desconexos. É de extrema importância o conjunto das faces nulas, fornecido pela fase anterior do algoritmo. A cada uma dessas faces nulas, é aplicado o operador de Euler MFKRH<sup>(7)</sup>, que cria uma nova face, cujo contorno é o loop interior da face nula.

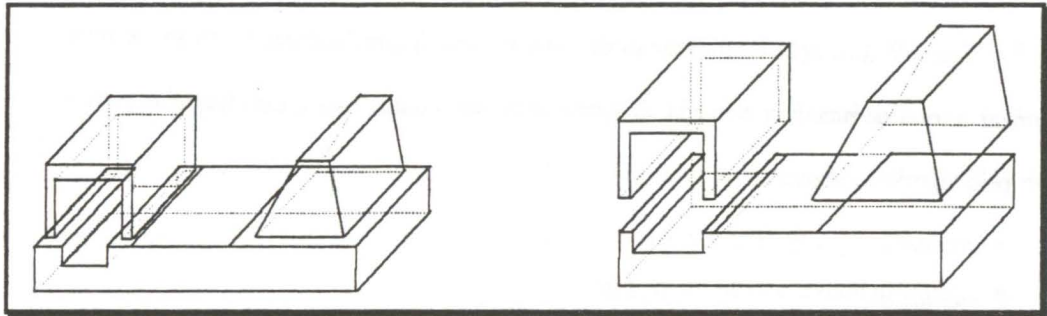
---

<sup>5</sup> - Operador de Euler que destroi uma face por destruição de uma aresta.

<sup>6</sup> - Operador de Euler que cria um *loop* interior por destruição de uma aresta.

<sup>7</sup> - Operador de Euler "Make Face Kill Ring and Hole".





(a)

(b)

Figura 11 - Resultado final da aplicação do Algoritmo de Corte.

Sempre que o operador MFKRH é aplicado, é feita uma reclassificação do modelo inicial, de modo a reconstruir os conjuntos de faces, arestas e vértices, de cada uma das *shells* resultantes (visualizadas na Figura 11 (a)), para obtermos um conjunto de modelos correctos após o algoritmo de corte. Este processo é realizável usando as relações topológicas de adjacência entre faces. Após reclassificadas as faces, o processo de reclassificação das arestas e dos vértices é trivial.

O resultado final da aplicação do algoritmo é o ilustrado pela Figura 11 (b).

#### 4 - Comparação Com Outras Aproximações

Relativamente a outras aproximações, nomeadamente Mäntylä [MANM-88], existem algumas melhorias essencialmente nas fases de Inserção de Arestas Nulas, Formação e Eliminação da Zona Neutra e Cisão do Sólido.

Na inserção das arestas nulas, a diferença consiste no modo como são identificados as *halfedges* (semi-arestas) que, aquando da criação da aresta nula, permitem demarcar o ciclo de arestas que ficam em torno de um dos vértices, do ciclo das que ficam em torno do outro vértice.

As fases de formação e eliminação da zona neutra, foram fundidas numa só, de modo a otimizar o total de memória ocupada. O modo como estas duas fases foram fundidas pode-se resumir pelo algoritmo seguinte:

- Ordenação lexicográfica das Arestas Nulas;
- **Enquanto** houver arestas nulas **Faz**
  - Determina um ciclo de interligação;
  - Forma a zona neutra correspondente ao ciclo;
  - Elimina a zona neutra criada;
  - Remove as arestas já utilizadas nos ciclos de interligação;
- **FimEnquanto**;

Finalmente na fase da cisão, os melhoramentos são significativos, uma vez que possibilitam uma grande economia de memória, bastante importante devido ao elevado espaço de memória ocupado por um modelo BRep minimamente complexo. O processo de construção das *shells* é realizado com base numa interrogação otimizada às relações de adjacência entre faces.

Em anexo são apresentados alguns exemplos de aplicação do algoritmo descrito sobre modelos criados com um modelador de sólidos (M3D) em desenvolvimento no Grupo de Métodos e Sistemas Gráficos.

## 5 - Conclusões

O presente trabalho destinou-se a apresentar um algoritmo de corte de modelos sólidos resultante da optimização de outras aproximações, algoritmo implementado e testado no âmbito dos trabalhos de I&D do Grupo de Métodos e Sistemas Gráficos (Dep. de Matemática - FCTUC). Esta ferramenta pode ser utilizada nas fases de modelação dos modelos sólidos, de visualização (nomeadamente de secções e/ou de cortes) e de análise (ex. cálculo de propriedades integrais). A sua grande generalidade, em relação ao leque de modelos sólidos a que pode ser aplicado, deverá ser realçada.

É de realçar o carácter particular deste algoritmo como uma primeira abordagem ao problema das operações booleanas entre modelos sólidos BRep [REAV-83], podendo mesmo ser encarado com um caso particular de uma operação booleana entre dois modelos, um dos quais é composto por apenas uma face, de dimensões infinitas.

A robustez do algoritmo só pode ser cabalmente comprovada se for implementado sobre Operadores de Euler implementados de forma otimizada, uma vez que deles depende em grande medida. Por esse motivo, a par da migração da implementação existente para diferentes plataformas (PC, UNIX) e diferentes linguagens de programação (Pascal, C/C++), continuará a ser dedicada atenção à optimização dos próprios Operadores de Euler. Por outro lado, através da utilização de algoritmos de conversão de representações adequados, pretende-se utilizar esta ferramenta com outros tipos de modelos.

#### *Agradecimentos*

Os autores agradecem ao Abel Gomes e a todos os que vêm contribuindo para a implementação do M3D, tornando assim possível a obtenção dos modelos apresentados.

### Referências

- [BAUB-74] **Baumgart, B.** " *Geometric Modeling for Computer Vision* ", PhD thesis, Stanford University, 1974.
- [ENJL-90] **Encarnação, J. , Lindner, R. e Schlechtendahl, E. ,** " *Computer Aided Design: Fundamentals and System Architectures*", Springer-Verlag, Berlim, 1990.
- [MANM-88] **Mäntylä, Martti,** " *An Introduction to Solid Modeling*", Computer Science Press, Maryland, USA, 1988.
- [MORM-85] **Mortenson, M.,** " *Geometric Modeling*", John Wiley and Sons, New York. 1985.
- [PAGB-88] **Pahl, G. e Beitz, W.** " *Engineering Design*", Springer-Verlag, London, 1988.
- [REAV-83] **Requicha, A., A. e Voelcker, H. B.,** " *Boolean Operations in solid modelling: Boundary evaluation and merging algorithms*", *Proc. IEEE*, 3(7), Oct. 1983, pag. 30-44.
- [REQA-80] **Requicha, A., A., G.,** " *Representations of Solid Objects-Theory, Methods and Systems*", *ACM Computing Surveys*, Dec. 1980.



Anexo - Exemplos de Aplicação

