

# POLIGONIZAÇÃO ADAPTATIVA DA FRONTEIRA DE SÓLIDOS CSG

ANTÔNIO LOPES APOLINÁRIO JÚNIOR


TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO EM ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por :



---

PROF. RONALDO CESAR MARINHO PERSIANO, D. SC.  
(PRESIDENTE)



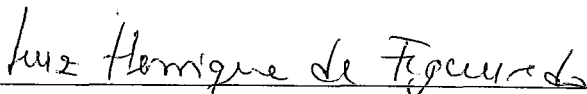
---

PROF. PAULO ROMA CAVALCANTI, D. SC.



---

PROF. ANTÔNIO ALBERTO FERNANDES DE OLIVEIRA, D. SC.



---

PROF. LUIZ HENRIQUE DE FIGUEIREDO, D. SC.

Rio de Janeiro, RJ - Brasil  
Março 1995

APOLINÁRIO, ANTÔNIO LOPES, JR.

Poligonização Adaptativa da Fronteira de Sólidos CSG. [Rio de Janeiro] 1995.  
viii, 107p. 29,7 cm (COPPE/UFRJ. M.Sc., Engenharia de Sistemas e Computação,  
1995)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Modelagem de Sólidos

2. Algoritmos Adaptativos

I. COPPE/UFRJ

II. Título (série)

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para obtenção do grau de Mestre em Ciências (M. Sc.)

## Poligonização Adaptativa da Fronteira de Sólidos CSG

Antônio Lopes Apolinário Júnior  
Março de 1995.

Orientador : Prof. Ronaldo Cesar Marinho Persiano  
Programa : Engenharia de Sistemas e Computação

Nesse trabalho estaremos interessados em construir uma aproximação poligonal da superfície de um sólido, representado no esquema CSG. Para alcançar esse objetivo iremos lançar mão da construção de uma triangulação adaptativa do espaço, no qual o sólido se encontra imerso. Partiremos de uma triangulação inicial, formada por seis tetraedros resultantes da decomposição da caixa envolvente do sólido. A partir desses elementos iniciaremos um processo de refinamento adaptativo dessa triangulação.

Esse processo de refinamento será guiado por dois critérios básicos : o seletivo e o adaptativo. O primeiro deve determinar se, dado um certo tetraedro da triangulação, a superfície do sólido intercepta seu interior. Dessa forma é possível concentrar o processo de refinamento nos elementos que podem melhorar a qualidade da aproximação quando refinados.

O segundo critério irá avaliar qual o erro de estaremos introduzindo na aproximação ao substituímos uma dada superfície por outra linear, dentro de um dado tetraedro. Com esse critério podemos abortar o processo de refinamento tão logo a aproximação linear seja suficientemente boa, dentro de uma determinada tolerância.

A poligonização resultante pode ser utilizada como representação intermediária dentro de um modelador CSG, para fins de visualização do modelo, bem como para a conversão do modelo CSG para uma representação por fronteira.

Abstract of Thesis presented to COPPE/UFRJ as partial fulfillment of the requirements for the degree of Master of Science (M.Sc.).

## Adaptive Polygonization of CSG Solids Boundary

Antônio Lopes Apolinário Júnior  
March 1995

Thesis Supervisor : Ronaldo Cesar Marinho Persiano  
Department : Computation and Systems Engineering

In this work, we are interested in constructing a polygonal approximation of a CSG solid boundary. In order to reach this goal we will use a three-dimensional adaptive triangulation of the space where the solid is immerse. We will start from an initial triangulation, composed by six simplices, produced by the decomposition of the solid bounding box. Based on those simplices an adaptive refinement process will begin.

Such process will be guided by two basic criteria : the selective and the adaptive. The first one will determine whether the solid boundary cuts the interior of some simplex. So we can concentrate the work on simplices which can improve the quality of the approximation.

The second criterion will evaluate the error when the original surface, defined by the boundary of the solid, is replaced by a planar surface. With this error value we can stop the refinement process as soon as this substitution fulfills our precision requirement.

That polygonization of an CSG solid boundary can be used as an intermediate representation in a solid modeling system, based on a CSG scheme, in order render the models or as a way to convert a CSG representation to a boundary representation.

## AGRADECIMENTOS

Várias foram as pessoas que, direta ou indiretamente, contribuíram para a realização desse trabalho. A elas gostaria de deixar aqui expressa, mesmo que em poucas palavras, minha gratidão.

Aos meus pais agradeço pela força e empenho com que sempre se dedicaram na tarefa, nem sempre simples, de me educar e preparar para enfrentar os desafios cotidianos.

À minha esposa Andréa, pela compreensão e incentivo que demonstrou desde minha decisão de tentar a pós-graduação até hoje.

A Ronaldo Marinho Persiano, meu orientador, por toda a experiência que pude adquirir durante esses três anos de convívio, tanto como aluno quanto orientado.

À Valéria pelas férteis discussões durante a fase de implementação, caronas e papos sobre os mais variados assuntos.

Ao nosso suporte Maurício Nardone, por ter sempre resolvido os diversos “pepinos” nas máquinas do LCG.

A Paulo Roma, pela ajuda e interesse demonstrados na fase final do trabalho.

Aos meus colegas de turma Daisy, José Maria, Adailton e Márcio, pelas várias horas de estudo e conversa fiada que tivemos durante o período de créditos.

Aos colegas Brito, Sara, Sonja, Maria Célia, Carla Godinho, Frederico, Nelma e Carla Cristina, pelas várias horas de descontração tão necessárias em alguns momentos.

Ao CNPq e CAPES que deram suporte financeiro no desenvolvimento desse trabalho.

Por fim a todos os meus familiares e amigos, que sempre me deram especial força para chegar ao término desse trabalho, com as tradicionais perguntas : “E então, já defendeu sua tese ?”, “Já marcou sua defesa ?”, etc.

# Índice

---

<b>INTRODUÇÃO</b>	<b>1</b>
<b>1. MODELAGEM DE SÓLIDOS</b>	<b>4</b>
1.1 INTRODUÇÃO	4
1.2 CONCEITO DE SÓLIDO	5
1.3 ESQUEMAS DE REPRESENTAÇÃO	7
1.4 ESQUEMA DE REPRESENTAÇÃO DE FRONTEIRA	9
1.4.1 PROPRIEDADES DO ESQUEMA DE REPRESENTAÇÃO POR FRONTEIRA	12
1.5 ESQUEMA DE DECOMPOSIÇÃO ESPACIAL	12
1.5.1 PROPRIEDADES DO ESQUEMA DE DECOMPOSIÇÃO ESPACIAL	14
1.6 ESQUEMA DE GEOMETRIA SÓLIDA CONSTRUTIVA	15
1.6.1 PROPRIEDADES DO ESQUEMA CSG	16
1.7 ESQUEMAS HÍBRIDOS	17
1.8 NOVOS ESQUEMAS	18
1.9 CONCLUSÃO	19
<b>2. POLIGONIZAÇÃO DE UMA SUPERFÍCIE</b>	<b>21</b>
2.1 INTRODUÇÃO	21
2.2 DETALHANDO O ESQUEMA CSG	22
2.2.1 NOVA ABORDAGEM PARA O PROBLEMA	23
2.3 SUBDIVISÃO ESPACIAL	24
2.4 DECOMPOSIÇÃO CELULAR	25
2.4.1 CÉLULA	25
2.4.2 SIMPLEXOS	26
2.4.3 DECOMPOSIÇÃO CELULAR	27
2.4.4 TRIANGULAÇÕES	28
2.5 MODELO DE INTERPOLAÇÃO LINEAR	29
2.5.1 GERAÇÃO DA POLIGONIZAÇÃO	30
2.5.2 ALGORITMO	32
2.6 NOVA PROPOSTA	33
<b>3. SUBDIVISÃO ESPACIAL SELETIVA</b>	<b>35</b>
3.1 INTRODUÇÃO	35
3.2 TRIANGULAÇÕES CFK	35
3.2.1 DEFINIÇÃO	35

3.2.2 VARIAÇÕES DE UMA TRIANGULAÇÃO CFK	37
<b>3.3 SUBDIVISÃO SELETIVA DO ESPAÇO</b>	<b>40</b>
3.3.1 A TRIANGULAÇÃO INICIAL	40
3.3.2 O PROCESSO DE SUBDIVISÃO SELETIVO	41
3.3.3 O ALGORITMO	41
3.3.4 CONSIDERAÇÕES SOBRE O ALGORITMO	42
<b>3.4 NOVA PROPOSTA</b>	<b>43</b>
<b>3.5 CONCLUSÕES</b>	<b>44</b>
<b><u>4. SUBDIVISÃO ESPACIAL ADAPTATIVA</u></b>	<b><u>45</u></b>
4.1 INTRODUÇÃO	45
4.2 POR QUE A ADAPTATIVIDADE ?	45
4.3 COMO GARANTIR A CONTINUIDADE ?	46
4.4 REFINAMENTO BÁSICO	47
4.5 MANUTENÇÃO DA ESTRUTURA DA TRIANGULAÇÃO	51
4.6 SUBDIVISÃO ADAPTATIVA	54
4.7 CONCLUSÃO	56
<b><u>5. CRITÉRIO DE INTERFERÊNCIA</u></b>	<b><u>57</u></b>
5.1 INTRODUÇÃO	57
5.2 REPRESENTAÇÃO DAS PRIMITIVAS	58
5.3 SUBDIVISÃO BINÁRIA DOS COEFICIENTES BÉZIER DE UMA PRIMITIVA	62
5.4 CONSTRUÇÃO DO CRITÉRIO DE INTERFERÊNCIA	64
5.4.1 CLASSIFICAÇÃO DA ÁRVORE CSG	67
5.4.2 LOCALIZAÇÃO DE UM SÓLIDO	68
5.4.3 ALGORITMO DE CLASSIFICAÇÃO	69
5.5 CONCLUSÃO	70
<b><u>6. CRITÉRIO PARA POLIGONIZAÇÃO</u></b>	<b><u>72</u></b>
6.1 INTRODUÇÃO	72
6.2 CRITÉRIO ADAPTATIVO	72
6.3 CÁLCULO DA APROXIMAÇÃO	77
6.3.1 VÉRTICES QUE ANULAM A FUNÇÃO CARACTERÍSTICA	78
6.3.2 ALGORITMO PARA POLIGONIZAÇÃO	80
6.4 QUALIDADE DAS FACES DA APROXIMAÇÃO	82
6.4.1 COLAPSO DE VÉRTICES	82
6.5 CONCLUSÃO	84
<b><u>7. AVALIAÇÃO E CONSIDERAÇÕES FINAIS</u></b>	<b><u>86</u></b>
7.1 INTRODUÇÃO	86
7.2 AVALIAÇÃO DO ESQUEMA PROPOSTO	87
7.2.1 ANÁLISE DA COMPLEXIDADE DO ALGORITMO	87

7.2.2 EXEMPLOS E RESULTADOS	91
7.3 CONCLUSÃO	102
7.4 SUGESTÕES PARA TRABALHOS FUTUROS	102
<b>REFERÊNCIAS</b>	<b>104</b>

---



# INTRODUÇÃO

---

Modelagem de Sólidos é a área da Computação Gráfica que trabalha com a representação, visualização, projeto e análise de modelos de objetos físicos tridimensionais. Nos últimos anos esta área de pesquisa vem tendo um grande crescimento fomentado tanto pelo desenvolvimento tecnológico quanto teórico. Esse crescimento vem alargando cada vez mais o uso da Modelagem de Sólidos em diversos ramos do conhecimento humano, tais como Arquitetura, Engenharia, CAD/CAM, robótica, etc [REQU92].

Vários esquemas de representação de sólidos têm sido propostos na tentativa de atender, da forma mais plena possível, aos requisitos das diversas áreas em que a Modelagem de Sólidos pode ser aplicada. Entre os mais “populares” estão os esquemas de *Representação por Fronteira (B-rep)* [MÄNT90], *Decomposição Espacial* [JACK80] e *Geometria Sólida Construtiva (CSG)* [HOFF89].

Dentro de nosso trabalho estaremos interessados particularmente no esquema CSG. De forma resumida, esse esquema representa um sólido a partir de outros sólidos básicos e pré-definidos, denominados *primitivas*. Aplicando sobre essas primitivas operações de adição/remoção de material e posicionamento no espaço é possível construir sólidos bastante complexos, de forma relativamente simples.

Um sólido CSG, assim construído, é plenamente descrito por uma expressão algébrica composta das primitivas como operandos e das operações de adição/remoção/posicionamento. Essa forma de representação, apesar de concisa e fácil de ser editada, torna algumas operações mais complicadas que em outros esquemas. Em particular, algoritmos que necessitem de informações a respeito da fronteira do sólido (como, por exemplo, os utilizados para visualização), tem seu trabalho aumentado já que ela não se encontra explicitamente representada na expressão algébrica associada ao sólido.

Nossa proposta visa construir uma representação formada por faces planares. Esse conjunto de faces representam uma superfície linear por partes, aproximação da superfície original. A essa representação auxiliar denominaremos *poligonização do sólido*.

Para isso iremos promover uma subdivisão do espaço, no qual o sólido está imerso, em células definidas por tetraedros, de forma que tenhamos uma triangulação cobrindo todo o espaço. Baseado nessa triangulação, definiremos um interpolador linear, a partir do qual seremos capazes de calcular, em cada tetraedro, uma face poligonal, que aproxima a superfície dentro daquela região.

Os sólidos em geral não ocupam todo o espaço no qual se encontram. Portanto, nossa triangulação do espaço deve ser inteligente o suficiente para concentrar esforços subdividindo apenas regiões que contêm parte da superfície a ser poligonalizada. A essa característica denominaremos *seletividade*.

Como em geral objetos do mundo real não possuem uma fronteira completamente uniforme - com respeito à curvatura da superfície que os definem - é interessante que nosso esquema seja capaz de levar em conta tal fato. Para tanto nossa proposta é construir uma triangulação que possa ser mais refinada (células de menor tamanho) em regiões onde a curvatura da superfície a ser aproximada seja alta. De forma análoga, a triangulação deve ser menos refinada (células com tamanho maior) em regiões onde a curvatura da superfície a ser aproximada seja baixa, o que significa que a própria superfície é próxima a um plano. Dizemos que uma triangulação que atende a essa característica é *adaptativa*.

Para que a idéia de seletividade possa efetivamente dar resultado, é importante poder contar com um critério confiável e de fácil aferição para a classificação de uma célula, no que tange a passagem ou não em seu interior da superfície a ser aproximada. Da mesma forma, para que nosso critério de adaptatividade possa dar resultado, precisamos definir um critério preciso e relativamente simples para determinação do erro cometido na substituição da região da superfície dentro de um tetraedro por um plano. Claramente tal critério deverá levar em conta uma análise local da curvatura da superfície.

Nossa proposta é fornecer um algoritmo, que construa tal triangulação adaptativa, guiado por um critério seletivo e outro adaptativo, que sejam eficientes e de baixo custo computacional. [BLOO88], [WARR90] e [VELH90] apresentam proposta semelhante no contexto da construção de aproximações de superfícies implícitas.

A apresentação de nosso trabalho será feita em sete partes. Na primeira apresentaremos uma visão geral da área de modelagem de sólidos, seus principais paradigmas e suas características. Procuraremos destacar os principais pontos de cada esquema de representação, ressaltando sua utilização e discutindo seus problemas. Tentaremos apresentar também, ainda que de forma resumida, alguns esquemas que tentam ampliar o poder de expressão dos esquemas “clássicos”.

No segundo capítulo apresentaremos um primeiro algoritmo, bastante simples e sem nenhum dos dois critérios, para a geração da poligonização de uma superfície genérica. Formalizaremos o conceito de triangulação e mostraremos porque esse tipo de subdivisão espacial nos será útil. Também apresentaremos uma primeira formulação para o interpolador linear, a ser definido dentro do domínio da superfície, que servirá de base para a construção da poligonização.

No capítulo subsequente discutiremos uma primeira forma de promover a construção de uma triangulação do espaço levando em consideração o primeiro dos critérios : o seletivo. Para tanto utilizaremos uma classe de triangulações baseada nas triangulações CFK. Abordaremos o algoritmo proposto por [SALI92] discutindo suas

principais restrições para que possa incorporar também o critério adaptativo durante o processo de construção da triangulação do espaço.

Em seqüência iremos propor um novo algoritmo que incorpore tanto o critério seletivo quanto o adaptativo. Esse algoritmo irá se basear em um procedimento primeiramente apresentado por [RIVA92] e mais tarde adaptado por [BARB94] para o refinamento adaptativo de uma triangulação baseada em extensões das triangulações CFK.

Os capítulos cinco e seis darão conta da formalização da construção dos critérios seletivo e adaptativo, respectivamente. Nesses capítulos entraremos em detalhes de como podemos classificar um tetraedro, com respeito à sua interseção com a superfície a ser aproximada, bem como à forma pela qual podemos avaliar o erro cometido na substituição de uma superfície por uma outra linear, no interior de um tetraedro.

Finalizando nosso trabalho, apresentaremos uma análise crítica dos resultados alcançados por nosso método. Discutiremos alguns detalhes ligados à implementação computacional e propostas para trabalhos futuros.

# 1. MODELAGEM DE SÓLIDOS

---

## *1.1 Introdução*

Um modelo pode ser visto como uma forma de representação de um dado objeto ou fenômeno, pertencente ao mundo real, com o objetivo de tornar possível o estudo de suas características ou interações com o meio em que se encontra, de forma mais simples.

Para alcançar esse objetivo a definição de um modelo em geral propõe algumas simplificações das situações encontradas no mundo real. Portanto, ao definirmos um modelo, estamos estabelecendo um certo escopo do mundo real sobre o qual esse modelo poderá representar os objetos ou fenômenos. Em outras palavras, uma vez definido o limite de atuação de nosso modelo, este será válido apenas para uma determinada classe de objetos ou problemas, nos quais as simplificações impostas no modelo não o invalidem.

Dentro desse quadro, podemos definir a Modelagem de Sólidos como um conjunto de teorias, técnicas e algoritmos que permitem construir, analisar e manipular esquemas não ambíguos que representam objetos sólidos (tridimensionais) do mundo real.

Nesse capítulo discutiremos, de forma resumida, alguns dos principais conceitos relacionados com os esquemas de representação de sólidos tradicionalmente utilizados.

Além disso apresentaremos alguns desses paradigmas, não só colocando em discussão suas características, mas também tentando fazer uma análise comparativa entre eles.

Por fim apresentaremos alguns esquemas mais recentes, que pretendem tornar possível a representação de uma classe mais ampla de objetos comparados com os esquemas tradicionais.

## 1.2 Conceito de Sólido

Os primeiros sistemas computacionais propostos para a automação do processo de modelagem de objetos tridimensionais eram desprovidos de qualquer tipo de modelo de representação formal. Os objetos eram representados por um conjunto de linhas que “definiam” sua superfície.

Esse tipo de representação, conhecido por *Wireframe*, possui uma série de problemas. Uma única representação *wireframe* pode corresponder a mais de um sólido, ou seja, a representação do sólido é ambígua. Além disso nem todas as representações possíveis de construção usando apenas linhas representam um objeto sólido real (não há garantia de validade da representação). Exemplos dessas situações são apresentados na figura 1.1.

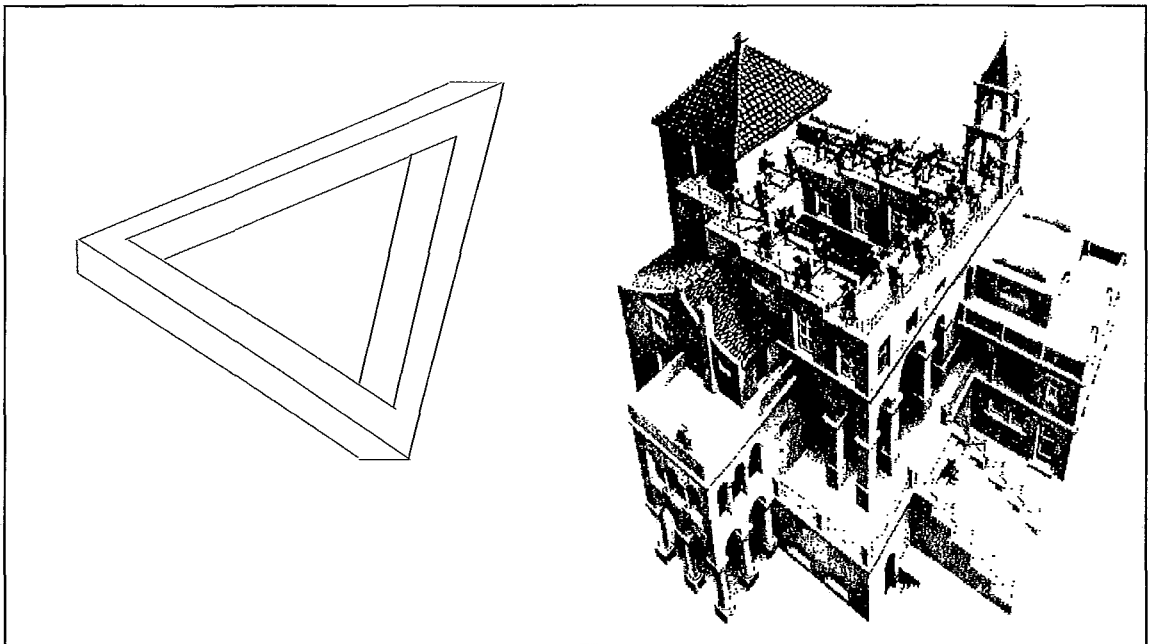


Figura 1.1 Exemplo de duas situações onde objetos são representados apenas através de linhas. Em ambos os casos tais objetos não podem ser construídos em suas versões tridimensionais. A figura à direita apresenta uma ilustração de M.C.Escher.

No sentido de garantir a validade da representação, entre outras propriedades, os primeiros esforços na área de modelagem de sólidos se concentraram na construção de uma base teórica, que pudesse servir de alicerce para a definição de modelos para representação de objetos tridimensionais.

O primeiro modelo formal para representar objetos sólidos foi proposto por [REQU80] e define um sólido como um subconjunto de pontos do espaço Euclidiano de dimensão 3. Nem todos os subconjuntos de  $E^3$ , porém, são adequados para representar

sólidos. Algumas características, inerentes aos objetos reais, devem ser respeitadas por essa classe de subconjuntos do  $E^3$ , das quais podemos destacar :

- A forma do sólido deve ser independente de sua localização e orientação.
- Um sólido deve conter material em seu interior e sua fronteira não deve possuir elementos isolados de dimensão menor que 3. Ou seja, um sólido deve ser homogeneamente tridimensional.
- Um sólido deve ocupar uma porção finita do espaço.
- Não deve haver alteração na forma do sólido quando este for movimentado pelo espaço através de movimentos rígidos.

Com base nessas características, podemos definir que classe de subconjuntos do  $E^3$  irá respeitar tais propriedades : subconjuntos fechados (contenham sua fronteira), limitados, regulares (fronteira sem partes "penduradas") e semi-analíticos (fronteira "bem comportada"). A essa classe de subconjuntos denominamos *r-sets*.

As operações passíveis de aplicação sobre um sólido (*r-set*) serão as operações usualmente definidas sobre conjuntos, ou seja, união, interseção e diferença. No contexto da modelagem de sólidos denominamos tais operações como *Operações Booleanas*, pela sua associação com os operadores da álgebra booleana.

Um problema surge nesse momento : os *r-sets* não são algebricamente fechados sobre os operadores booleanos normalmente definidos. A figura 1.2 mostra dois sólidos (*r-sets*), que, quando compostos por um desses operadores, resultam em um sólido que não é um *r-set*.

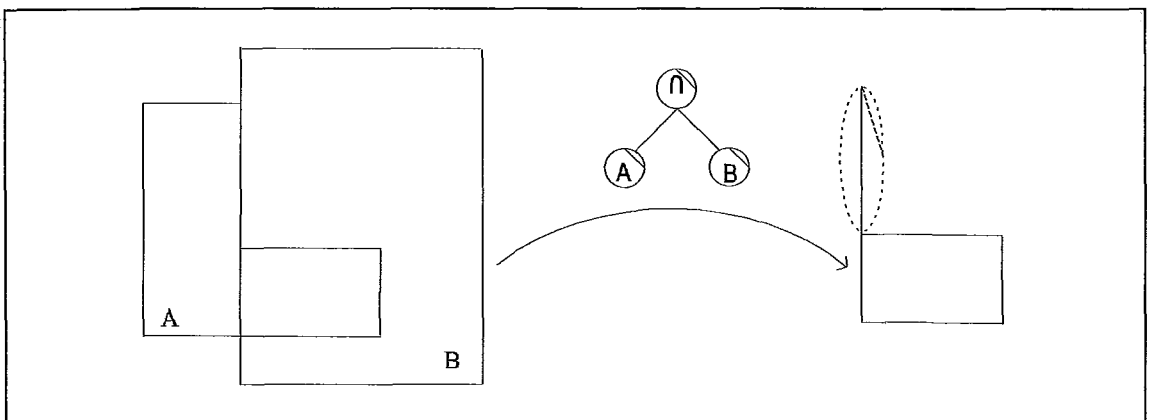


Figura 1.2 Exemplo de situação em que, partindo-se de sólidos regulares, e aplicando-se um operador booleano tradicional não se obtêm como resultado um sólido regular.

A solução encontrada para esse problema está na redefinição das operações booleanas. Denominamos operações booleanas regularizadas ( $op^*$ ) a seguinte seqüência de operações :

$$A \text{ op}^* B = \text{fecho}(\text{interior}(A \text{ op } B))$$

onde  $A$  e  $B$  representam os sólidos e  $op$  uma operação booleana convencional.

### 1.3 Esquemas de representação

Agora que possuímos um modelo formal dos objetos que queremos representar, devemos nos preocupar em formular abordagens que permitam representá-los computacionalmente.

Podemos definir formalmente um *Esquema de Representação* como sendo uma relação  $s$  que mapeia objetos de um *espaço de modelos*  $M$  (ao qual pertencem os *r-sets*) em um *espaço de representações*  $R$  (representando objetos válidos dentro de nosso modelo). Essa definição pode ser representada como o gráfico da figura 1.3.

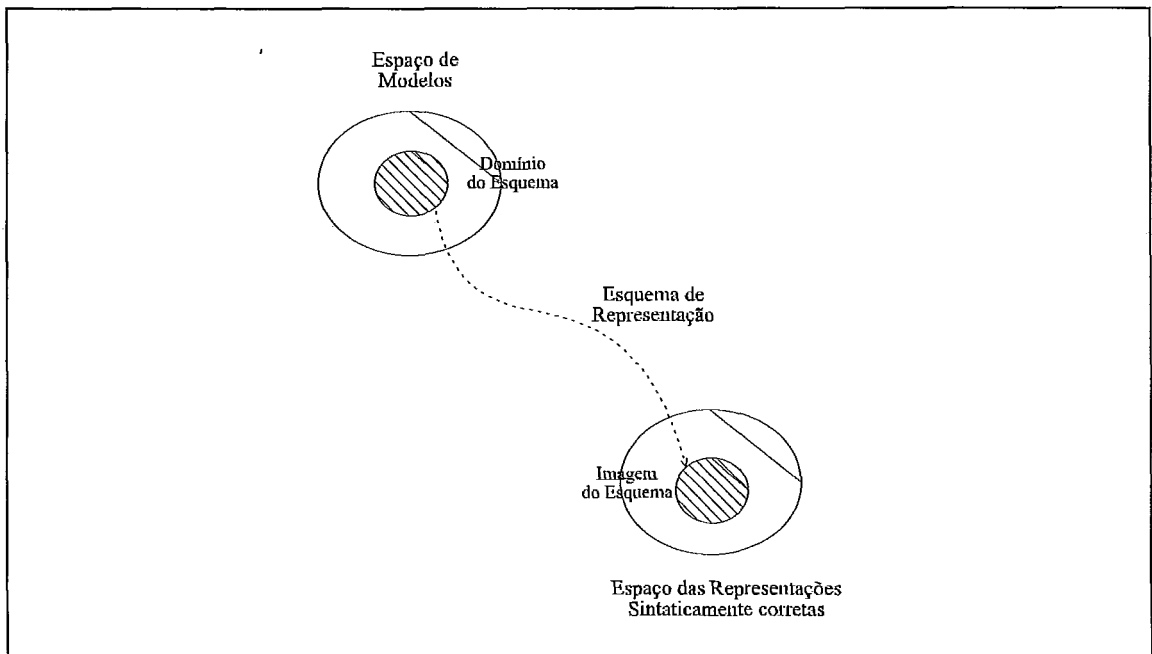


Figura 1.3 Modelo formal de um Esquema de Representação [REQU80].

Vale ressaltar que nem todos os elementos de  $M$  poderão ser representados através da relação  $s$ . De forma análoga, nem todas as representações válidas de  $R$  serão obtidas a partir de  $M$ .

Analisando os modelos sobre esse enfoque, ou seja, como relações, podemos de forma simples definir algumas propriedades importantes de um esquema de representação :

- i. Caso  $s$  seja uma função, ou seja, a cada elemento do domínio  $D$  de  $s$  esteja associado a um único elemento da imagem  $V$  de  $s$ , podemos dizer que tal esquema de representação é *único*.
- ii. Caso  $s^{-1}$  seja uma função, isto é, cada elemento da imagem  $V$  de  $s$  corresponde a apenas um elemento do domínio  $D$  de  $s$ , o esquema representado por  $s$  é dito *completo* ou *não ambíguo*.

Outras propriedades também são desejáveis em um esquema de representação :

- O *poder de expressão* de um esquema de representação, que está diretamente ligado ao seu domínio, ou seja, ao conjunto de entidades que este esquema pode representar.
- Um esquema deve garantir a *validade* dos objetos por ele representados.
- Deve ser possível, a partir de uma representação do esquema, ter acesso a informações suficientes para que se possa calcular qualquer propriedade geométrica desse objeto. A essa característica definimos *completude*.

Outras propriedades, sem terem uma conotação formal, mas igualmente importantes são

- A concisão do esquema, ou seja, quanta informação deve ser necessária para se construir um objeto do mundo real.
- Quão simples é a tarefa de criação da representação de um objeto no esquema.
- Robustez do esquema para a aplicação de algoritmos diversos, de acordo com os requisitos das aplicações que o utilizam.

Passaremos agora a discutir alguns dos principais esquemas de representação utilizados em Modelagem de Sólidos. Procuraremos também analisar cada um com respeito as propriedades, formais ou não, definidas.



## 1.4 Esquema de Representação de Fronteira

O *Esquema de Representação por Fronteira (B-rep)* surgiu baseado em um dos primeiros esquemas de representação de objetos tridimensionais : o *Wireframe*.

Como já mencionamos, o esquema *Wireframe* é bastante simples e fornece poucos recursos para operar com os objetos por ele representados. Além disso é um esquema ambíguo.

A idéia básica do esquema B-Rep reside no fato de um sólido poder ser descrito de forma não ambígua pelo conjunto de faces que compõe sua fronteira. Cada face por sua vez pode ser descrita pelo conjunto de arestas que a limitam. De forma análoga as arestas do modelo podem ser descritas pelos vértices que definem seus extremos.

Podemos, por conseguinte, enxergar um modelo B-rep como um modelo em camadas, onde cada vez que nos aprofundamos encontramos elementos de uma dimensão inferior. A figura 1.4 mostra esse esquema.

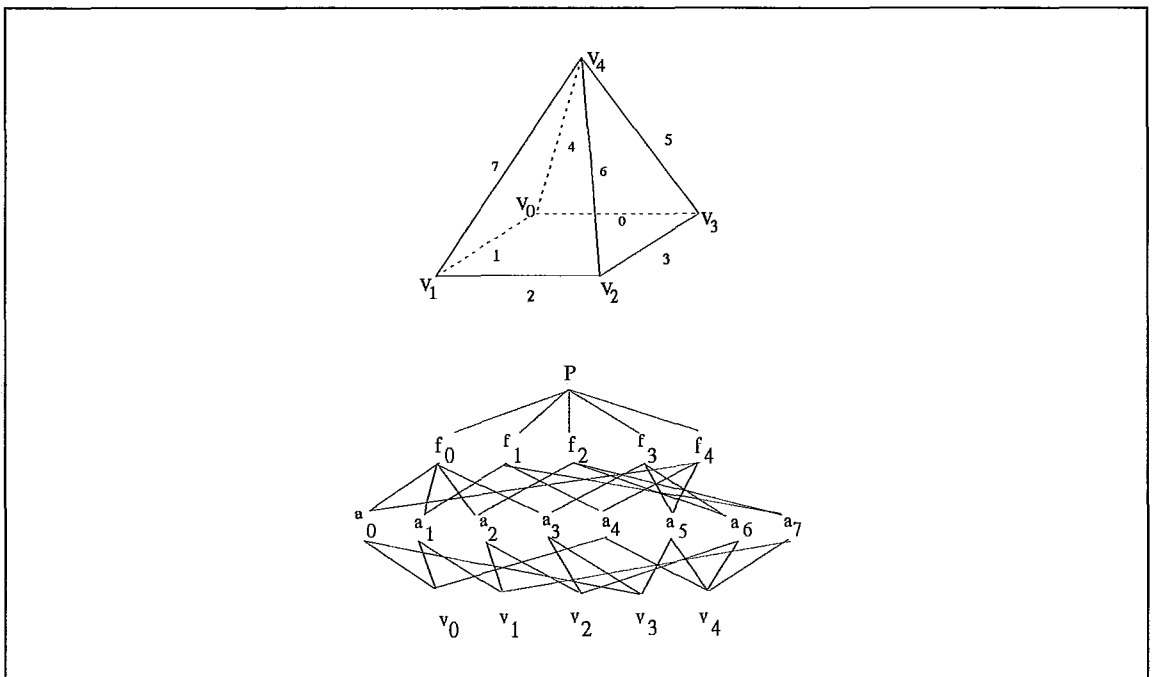


Figura 1.4 Exemplo de uma pirâmide de base quadrada descrita através do esquema de Representação por Fronteira (B-Rep).

Uma face pode definir diversos tipos de superfícies, como por exemplo planos, superfícies quadráticas, quárticas ou superfícies paramétricas. Além disso, a princípio, nenhuma restrição é imposta sobre a geometria das faces.

Podemos identificar em uma face dois lados : o interno e o externo. Portanto quando descrevemos o conjunto de arestas que compõem uma face devemos levar em conta a orientação da superfície, ou seja, se estamos sobre a face de fora ou de dentro do sólido.

As faces que irão definir a fronteira do sólido devem obedecer a algumas propriedades :

1. Uma face do objeto é um subconjunto de sua fronteira;
2. A união de todas as faces que compõem o objeto deve ser igual à sua fronteira;
3. Faces devem ter área e ser homoganeamente bidimensionais.

Para garantir a validade do arranjo de faces, mais informações devem ser armazenadas além das relacionadas à geometria das faces. Para tanto, informações topológicas, que dão conta das relações de conectividade entre os diversos elementos (faces, arestas e vértices) da representação, são acrescentadas ao modelo. Em outras palavras, as relações de incidência e adjacência entre os componentes do modelo serão explicitadas.

Isso permite que o modelo seja percorrido através de seus diversos elementos. Por exemplo podemos ter conhecimento sobre todas as arestas incidentes em um determinado vértice, ou que arestas formam determinada face.

Uma das formas mais conhecidas para representação da topologia de um modelo B-rep é a Winged-Edge [BAUM74]. Essa representação baseia-se na representação das informações combinatórias (ou topológicas) das arestas do objeto.

São armazenadas referências às faces adjacentes esquerda e direita (que compartilham a aresta) e as arestas anterior e posterior nos sentidos horário e anti-horário (referentes as faces esquerda e direita). A figura 1.5 mostra tal estrutura.

Para garantir a consistência topológica do esquema, Baumgart propõe a utilização dos *Operadores de Euler* [MANT88]. A função desses operadores é atuar diretamente sobre a estrutura topológica, promovendo alterações, mas garantindo a sua validade. É através desses operadores que a criação e edição de representações nessa estrutura são feitas.

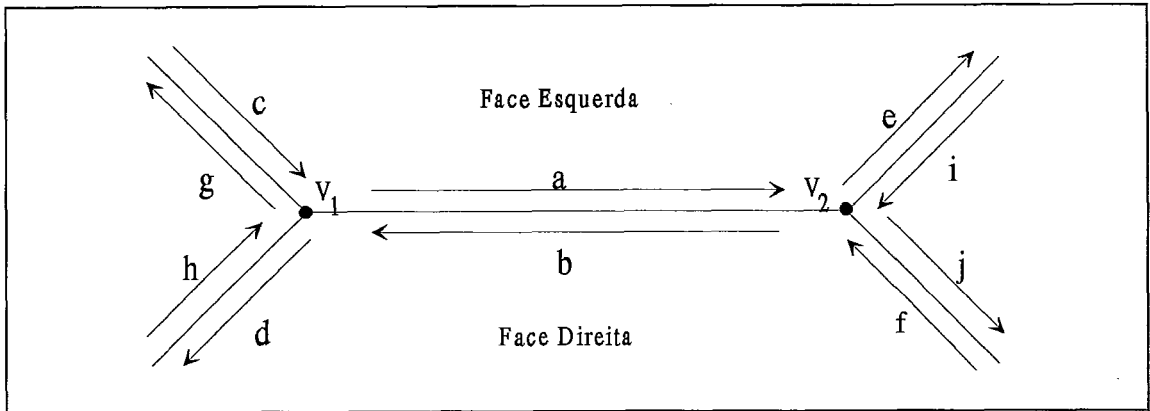


Figura 1.5 Modelo Winged-Edge, proposto por Baumgart, para representação da topologia em um esquema de representação por fronteira.

Apesar de garantirem a coerência topológica do objeto, os operadores de Euler são muito pouco intuitivos, ou seja, para se obter um alteração simples sobre a representação, em geral, vários operadores devem ser acionados em sequência.

Em [GUIB85] foi proposta uma estrutura alternativa, bem mais simples, para representar um superfície topológica : a *quad-edge*. Nessa estrutura, a única informação armazenada é relativa à aresta. Diferenciam-se, para cada aresta geométrica, duas arestas topológicas (uma para cada face). A consistência da representação é mantida através de uma álgebra : a *álgebra de arestas*.

[HOTT92] se vale dessa estrutura para a criação de uma representação B-Rep dentro da proposta de algoritmo para conversão CSG → B-Rep. Como a estrutura de dados é bem mais simples, as operações definidas para a manipulação do modelo são apenas 9, e denominadas *Funções de Aresta*. Tais funções na verdade são escritas com base em apenas duas funções. A figura 1.6 mostra a estrutura usada nesse esquema com seus nove operadores associados.

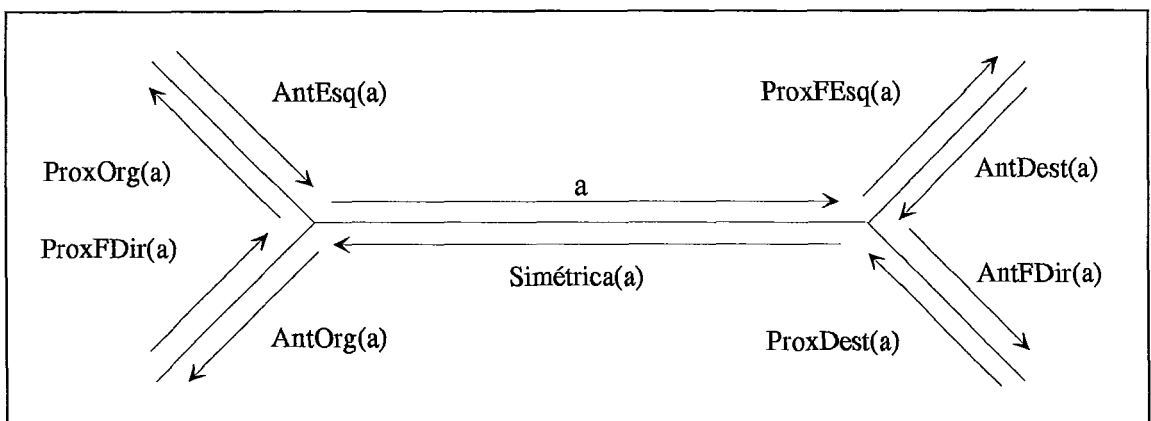


Figura 1.6 Modelo Aresta Orientada, proposto por Guibas e Stolfi, para representação das informações topológicas em um esquema de representação por fronteira.

### 1.4.1 Propriedades do Esquema de Representação por Fronteira

O esquema de representação por fronteira não é um esquema único, já que vários conjuntos de faces podem representar o mesmo sólido. Porém, uma vez fixado o conjunto de faces, este define um e somente um sólido, ou seja, não há ambigüidade no esquema.

A criação de um sólido nesse tipo de esquema de representação é uma tarefa bastante complexa, já que cabe ao usuário fornecer além das informações geométricas, as relações de adjacência entre seus elementos.

A falta de concisão da representação B-rep e sua instabilidade, no que tange a garantia de validade geométrica, são alguns dos pontos que pesam de forma negativa contra o modelo. Outro problema é o custo de armazenamento e manipulação das estruturas de dados utilizadas. A realização de operações booleanas em modelos B-Rep também não é uma tarefa simples, exigindo algoritmos bastante complexos [HOFF89] [MÄNT90].

O esquema B-rep é particularmente interessante em aplicações que necessitem de visualização rápida. Sua estrutura de dados possibilita a aplicação da maioria dos algoritmos de visualização, de forma simples e direta.

## 1.5 Esquema de Decomposição Espacial

Os modelos de decomposição espacial procuram descrever o sólido como uma coleção de células, cuja união fornece uma aproximação para o sólido.

Em geral, as células utilizadas para construir tal aproximação são cubos, uma vez que sua geometria simétrica possibilita uma série de simplificações nos algoritmos e na estrutura de dados usadas por esse esquema.

Um primeiro processo para a obtenção desse conjunto de células é a *Enumeração Exaustiva ou Decomposição Regular*. Nesse método define-se uma célula cúbica inicial que contenha o objeto, e a partir da qual construiremos a aproximação. Essa célula inicial, também chamada Caixa Envolvente (*Bounding-Box*) tem suas faces paralelas a um dos planos XY, XZ e YZ, definidos pelos eixos do Sistema Cartesiano. A partir dessa célula promovemos cortes paralelos a cada um dos plano acima descritos gerando 8 novas células. Para cada nova célula criada aplicamos os mesmos cortes recursivamente até que o tamanho das células geradas seja adequado para o grau de precisão requerido. Ao final desse processo as células são classificadas com relação ao sólido, se estão no interior, exterior ou na fronteira do sólido. Dessa classificação uma

lista das células que estão total ou parcialmente contidas no objeto é gerada. As células dessa lista irão formar a aproximação do sólido.

Tal processo de decomposição pode ser visto como a construção de uma árvore octal completa, onde a raiz é a célula ou cubo inicial, e cada nível representa um nível do processo de decomposição.

Apesar de simples, a *enumeração exhaustiva* é bastante custosa no que diz respeito ao número de células geradas. É simples observar que a cada novo nível criado na árvore completa, o número de células é duplicado. Como, em geral, para se conseguir uma aproximação razoável do objeto as células tem tamanho reduzido, a quantidade de células gerada é bastante elevada.

A partir da análise do método exposto podemos elaborar uma estratégia que tente minimizar o número de células geradas. Tal objetivo pode ser facilmente alcançado se levarmos em conta que, em muitos casos, podemos abortar o processo de decomposição antes de alcançar o tamanho de célula especificado pela precisão. Isso se deve ao fato de que o tamanho mínimo de célula é importante apenas na fronteira do sólido. Células localizadas no interior do sólido não tornam a aproximação melhor quando tem seu tamanho reduzido. Células exteriores não interessam na aproximação do sólido.

Em outras palavras, se decompuermos apenas as células de fronteira, que são efetivamente as células que determinam a precisão do modelo, podemos reduzir drasticamente o número de células geradas.

O esquema que representa tal idéia é o *Octree* [SAME]. Esse esquema também pode ser visto como a criação de uma árvore octal, porém não completa. A figura 1.7 mostra como essa árvore é gerada. Utilizamos um exemplo bidimensional para facilidade de representação. Nesse caso temos uma árvore quaternária ou *quadtree*.

Cada célula (correspondendo a um nó da árvore octal) é classificado como :

- Interior (negro)
- Exterior (branco)
- Fronteira (cinza)

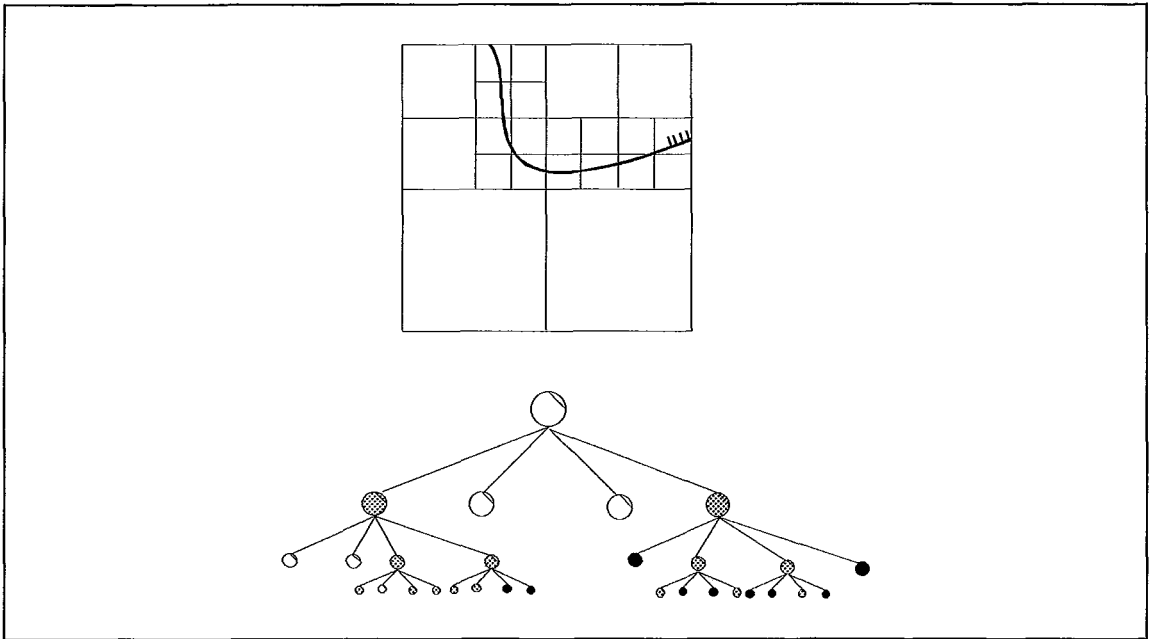


Figura 1.7 Exemplo de construção de uma aproximação da curva através do esquema de representação Decomposição Espacial. Nesse caso a estrutura da decomposição forma uma árvore quaternária (Quadtree).

Fica evidente que a lista de células geradas ao fim do processo, e que irá caracterizar a aproximação do sólido, irá diminuir. Apenas as folhas da árvore octal gerada, e os nós classificados como interiores, irão ser incluídos. Como a árvore não é mais completa a tendência é que o número de células seja menor.

Os modelos Octrees, portanto, são mais econômicos no número de células do que os modelos de Decomposição Regular. Porém ainda assim seu custo de armazenamento é elevado, no caso de modelos muito refinados.

### 1.5.1 Propriedades do Esquema de Decomposição Espacial

O poder de expressão do esquema de Decomposição Espacial é baixo no sentido que sempre iremos construir uma aproximação do sólido.

O esquema de Decomposição Espacial não é único, uma vez que um sólido pode ser descrito por diversas aproximações dependendo da definição da célula inicial, que é peça chave no processo de decomposição. Uma vez construída, a aproximação define o sólido de forma não ambígua.

Uma das dificuldades encontradas no esquema é a sua edição, já que a aplicação de movimentos rígidos é uma tarefa que pode se tornar bastante custosa.

Uma das razões que tornam esse esquema atraente é sua possibilidade de paralelização. Cada uma das 8 células iniciais pode ser entregue a um processador para gerar a aproximação daquela porção do espaço.

Outra característica bastante atraente nesse esquema é a facilidade de implementação das operações booleanas.

## 1.6 Esquema de Geometria Sólida Construtiva

O Modelo de Geometria Sólida Construtiva parte do princípio de que sólidos complexos podem ser construídos a partir de sólidos simples, aplicando operadores de adição e remoção de material e movimentos rígidos.

Esse processo de "montagem" de um sólido CSG pode ser representado por uma árvore binária. Cada folha dessa árvore corresponde a um sólido simples (primitiva). Em seus nós intermediários são representadas as operações aplicadas sobre esses sólidos primitivos ou, no caso de serem raiz de subárvores, aplicadas sobre os sólidos intermediários. A raiz dessa árvore está associada à representação do sólido final.

Essa árvore por sua vez pode ser interpretada de modo a produzir uma expressão algébrica, cujos operadores são as operações de adição/remoção de material e de posicionamento, enquanto que os operandos são as primitivas.

Os operadores de adição e remoção de material são associados diretamente aos operadores de manipulação de conjuntos : união, interseção e diferença. Os operadores sobre conjuntos são tradicionalmente denominados, nesse contexto, de operadores Booleanos.

É evidente que parte do poder de expressão do modelo CSG está associado à diversidade de primitivas utilizadas para construção dos sólidos. Em geral, as primitivas são compostas de elementos tais como cubos, esferas, toros, cilindros, etc.

Modeladores mais genéricos no entanto fornecem como primitivas básicas *semi-espaços*. Um semi-espaço é definido por uma função analítica definida da seguinte forma :

$$\begin{array}{ll}
 f: E^3 \rightarrow R & \\
 \forall p \in E^3 : f(p) > 0 & p \text{ é exterior} \\
 f(p) = 0 & p \text{ é fronteira} \\
 f(p) < 0 & p \text{ é interior}
 \end{array}$$

Modeladores que definem primitivas dessa forma possuem grande flexibilidade, pois com relação à função  $f$  não é imposta nenhuma restrição a não ser que esta seja analítica. Portanto podemos definir semi-espaços, em princípio, de qualquer grau. Na

figura 1.8 mostramos um exemplo da construção de uma primitiva utilizando essa abordagem.

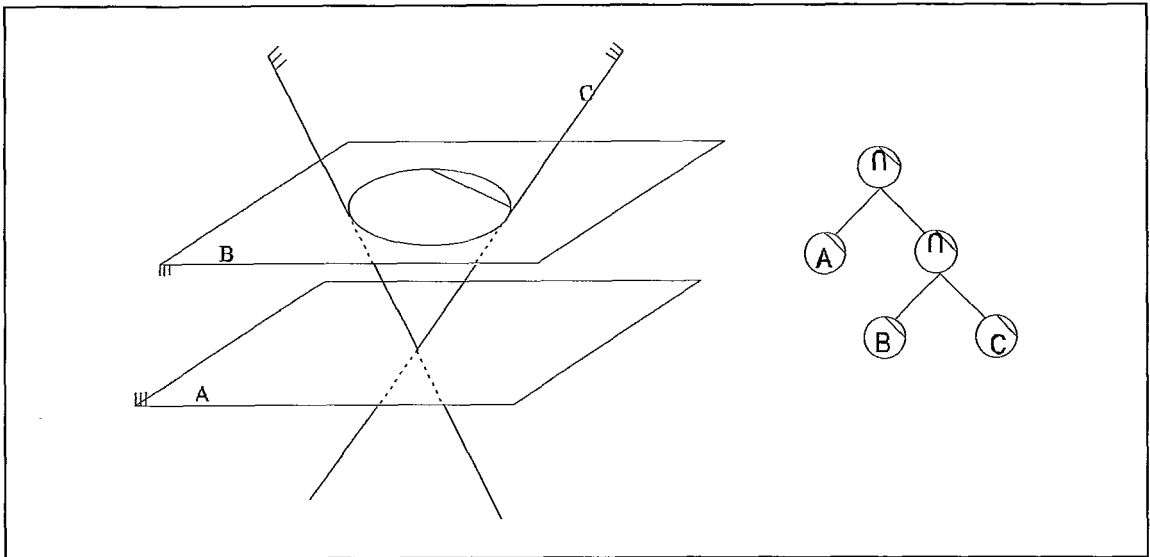


Figura 1.8 Representação de uma primitiva CONE decomposta em semi-espacos.

Uma restrição que devemos lembrar é que, para nós, sólidos são subconjuntos do  $E^3$  com algumas restrições (*r-sets*). Uma dessas restrições é que os *r-sets* são limitados, o que não acontece com os semi-espacos. Tal restrição pode ser facilmente contornada se impusermos um “limite” para o nosso universo, ou seja, trabalharmos com uma caixa envolvente dentro da qual nosso sólido está definido.

A construção de sólidos CSG está diretamente associada a uma linguagem, através da qual especificaremos como os sólidos básicos podem ser associados. Essa linguagem em geral é de fácil entendimento, o que torna a construção de um objeto pelo esquema CSG uma tarefa relativamente simples. Um exemplo de definição de uma linguagem para um modelador de sólidos CSG pode ser encontrado em [ESPE90]. Exemplos de definições de sólidos utilizando essa linguagem podem ser encontrados na figura 7.1 do capítulo 7 desse trabalho.

### 1.6.1 Propriedades do Esquema CSG

O poder de expressão de um esquema CSG está diretamente ligado ao rol de primitivas fornecido.

A validade do esquema é garantido visto que todas as operações booleanas são regularizadas.



O esquema de representação CSG tem por característica a não ambigüidade, porém seus modelos não são únicos. A figura 1.9 mostra um exemplo de tal situação. Essa característica é bastante crítica pois torna mais complexas tarefas como detectar quando uma representação corresponde a um objeto nulo. Esse problema é particularmente útil para detecção de contato entre dois objetos, pois se a interseção dos dois for nula não há contato. Soluções para esse problema são propostas em [TILO84].

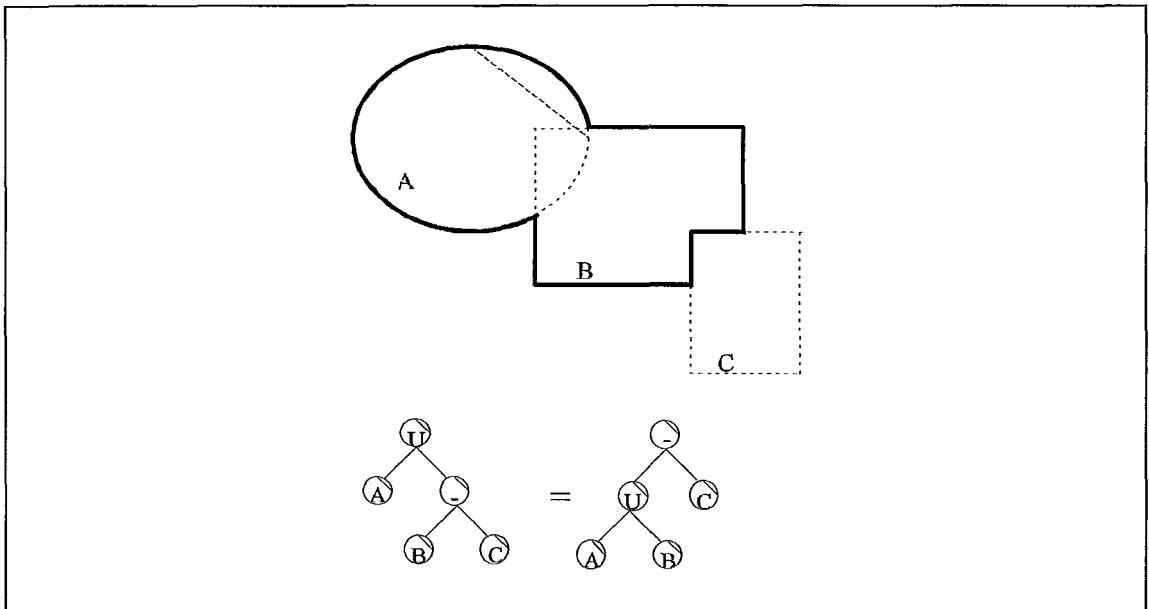


Figura 1.9 Não unicidade do esquema de representação CSG : as duas árvores representam o mesmo sólido.

Uma dificuldade adicional desse esquema aparece na aplicação dos algoritmos usuais para visualização de objetos tridimensionais. Tal fato se deve à falta de uma representação explícita da fronteira no esquema.

Voltaremos a discutir mais detalhes sobre este esquema nos próximos capítulos.

## 1.7 Esquemas Híbridos

A partir dos três modelos apresentados acima, podemos concluir que nenhum deles é versátil o suficiente para se adequar a todo o tipo de aplicação.

A construção de um sólido pelo esquema CSG é bem mais simples do que no esquema B-rep. Porém para a sua visualização o enfoque B-rep se torna bem mais atrativo do que o CSG. A possibilidade de paralelização bem como a facilidade para o cálculo de algumas propriedades dos objetos (como volume) favorecem o esquema de Decomposição Espacial.

Dessa forma, fica claro que um esquema híbrido, ou seja, que possa trabalhar com diferentes representações do mesmo sólido, torna-se uma solução bastante interessante na busca de suprir os pontos fracos de cada esquema.

Podemos pensar portanto, em um modelador que mantém uma representação CSG do sólido para a fase de construção/edição. Quando necessário visualizar o sólido, este pode ser convertido para o modelo B-rep para então ser processado por um algoritmo específico. Se quisermos calcular alguma propriedade do sólido podemos converter a representação para o esquema de Decomposição Espacial.

Dentro desse enfoque fica patente a necessidade de promover de forma eficiente e segura a conversão entre as várias representações possíveis dentro de um esquema híbrido. Tal tarefa no entanto não é das mais simples.

Algumas conversões são bastante simples. Por exemplo, um esquema de representação por Decomposição Celular pode ser convertido para uma representação B-rep de maneira simples. Outras conversões apesar de possíveis fazem uso de algoritmos complexos, como por exemplo a conversão B-Rep para CSG [SHAP91a][SHAP91b] ou mesmo a conversão CSG para B-rep [HOTT92].

A conversão entre alguns esquemas pode se tornar inviável. Nos casos em que desejamos converter esquemas com poder de representação distintos nem sempre essa conversão é possível nos dois sentidos. Esse é o caso, por exemplo, da conversão de um modelo B-rep para um modelo de Decomposição Celular.

Como exemplo de um esquema híbrido podemos citar as politrees e Extended Octrees [BRUN90] e o esquema Sweep-CSG [HUI94].

## 1.8 *Novos Esquemas*

Em geral com o objetivo de ampliar o poder de expressão dos esquemas de representação “clássicos”, várias novas propostas tem sido apresentadas.

A partir da análise do esquema CSG podemos concluir que uma de suas características é a garantia de validade quanto as representações criadas. Em outras palavras, partindo de sólidos válidos (r-sets) através das operações booleanas regularizadas garantimos que o sólido final gerado também será um sólido regular.

Porém, para determinadas aplicações, tal fato, ao invés de ser uma característica desejável, se torna uma restrição de representação bastante forte. Não que a característica de validade seja restritiva, mas sim a impossibilidade de representar objetos não homogeneamente tridimensionais.

Em diversas aplicações se deseja representar objetos formados a partir de diversas partes, cada uma composta de um material diferente, como aviões ou circuitos integrados. Nesses casos as superfícies de separação entre materiais deve ser representada para uma análise precisa [ROMA92].

Com o objetivo de permitir a representação dessas situações, [ROSS91] propõe um esquema que permite que objetos não homogêneos sejam representados, denominado *Constructive Non-Regularized Geometry (CNRG)*.

Esse esquema está baseado na definição de sólido a partir de agregados de regiões mutuamente disjuntas. Cada região é um subconjunto de pontos do  $R^n$ . Tais regiões não são necessariamente conexas, regulares ou de dimensão homogênea.

Um sólido CNRG é formado a partir da composição de várias regiões através de operações booleanas convencionais (não regularizadas) e de operações topológicas, como fecho, interior e fronteira. A seqüência dessas operações pode ser expressa a partir de uma árvore binária. Suas folhas representam as regiões primitivas e seus nós internos associações entre os sólidos intermediários e as operações definidas.

Como podemos observar, o esquema CNRG em muito se assemelha ao esquema CSG. Em vários aspectos, ele pode ser considerado uma extensão do esquema CSG. Por exemplo, ele estende o conceito de primitivas para objetos não sólidos. Também aumenta a possibilidade de composições apresentando operadores booleanos convencionais e topológicos, o que permite que as operações regularizadas também sejam representadas.

Os algoritmos geométricos concebidos para serem aplicados ao esquema CSG podem ser adaptados para o esquema CNRG.

## ***1.9 Conclusão***

Nesse capítulo apresentamos uma breve panorâmica da área de Modelagem de Sólidos : suas bases teóricas, paradigmas clássicos e novos esquemas de representação.

Os esquemas clássicos apresentados não são completos, no sentido de não atenderem de forma satisfatória a todos os requisitos básicos das aplicações que utilizam a Modelagem de Sólidos como ferramenta.

Uma idéia bastante usada é tentar combinar esquemas de representação distintos num mesmo sistema de modelagem. Com isso podemos contrabalançar as falhas de uns com as vantagens de outros, dando origem aos chamados esquemas híbridos.

Esforços também têm sido feitos no sentido de ampliar o poder de representação dos esquemas atuais, como o caso do esquema CNRG.

A partir do próximo capítulo daremos maior ênfase no esquema CSG de representação de sólidos. Estaremos interessados em atacar um dos pontos fracos do esquema : a determinação da fronteira de um sólido expresso nesse esquema.

Definiremos uma estratégia para resolver um problema mais genérico, a avaliação de uma superfície definida implicitamente. Iremos computar uma aproximação linear por partes dessa superfície promovendo uma subdivisão adaptativa do seu domínio.

Apesar de mais amplo, esse problema pode ser trazido para o contexto da Modelagem de Sólidos e ser usado, por exemplo, em um sistema de modelagem híbrido para conversão do esquema CSG para o esquema B-Rep, ou mesmo como ferramenta de visualização para modeladores CSG puros.

No próximo capítulo começaremos a apresentar nosso trabalho discutindo como deve ser essa subdivisão do espaço, como construí-la e quais as propriedades que esta deve possuir.

## 2. POLIGONIZAÇÃO DE UMA SUPERFÍCIE

---

### 2.1 Introdução

Nesse capítulo apresentaremos as idéias básicas que serão utilizadas em nossa proposta para a construção de uma poligonização da superfície de um sólido, representado no esquema CSG.

Algoritmos que necessitam avaliar a fronteira de um sólido CSG em geral se deparam com duas dificuldades. A primeira é inerente a própria complexidade da superfície do sólido. A segunda diz respeito à ausência de uma representação explícita da fronteira do sólido no esquema CSG.

Uma forma de tornar mais simples a utilização desses algoritmos dentro do esquema CSG é criar uma estrutura alternativa, de menor complexidade, que possa representar a superfície do sólido de forma explícita. A essa estrutura denominaremos *Poligonização da fronteira do sólido*. Ela será composta por um conjunto de polígonos planos, construídos de modo a formarem uma aproximação linear por partes da superfície original do sólido.

Para que possamos trabalhar com a poligonização, ao invés da superfície original, teremos de garantir que o erro cometido nessa substituição esteja dentro de uma certa tolerância preestabelecida.

A proposta que apresentaremos irá se concentrar em duas questões principais : a forma através da qual construiremos a poligonização e em como avaliar o erro cometido na substituição da superfície original por uma outra linear por partes.

Para alcançar esses objetivos voltaremos a tratar, agora com mais detalhamento, o esquema de representação CSG. Apresentaremos a seguir alguns conceitos básicos ligados ao processo de construção da poligonização. Discutiremos também, mesmo que de forma superficial nesse momento, como será a construção das faces (polígonos) que irão compor a poligonização da superfície do sólido.

## 2.2 Detalhando o Esquema CSG

Como já mencionado no capítulo anterior, o esquema CSG se utiliza de sólidos básicos, denominados primitivas, e operadores booleanos (regularizados), associados sob a forma de uma expressão algébrica, para representar um sólido. Essa expressão usualmente é representada por uma árvore binária, em que as folhas correspondem às primitivas e nós intermediários aos operadores booleanos.

As primitivas são definidas a partir de formas tridimensionais simples como blocos, esferas, toros, etc. Essas primitivas podem ser compostas com base em semi-espacos definidos por funções tri-variadas. Cada um desses semi-espacos é denominado de *função característica da primitiva*.

A representação das primitivas através de sua função característica pode ser útil com respeito à avaliação das operações booleanas (não regularizadas). A função característica que representa a união de duas primitivas pode ser obtida através da aplicação do operador máximo (*max*) sobre as funções características dos operandos. De modo análogo a função que representa a interseção de duas primitivas pode ser conseguida através do operador mínimo (*min*) aplicado às primitivas associadas ao operador interseção [SALI91]. Ou seja:

$$f = f_1 \cup f_2 = \max(f_1, f_2)$$

$$f = f_1 \cap f_2 = \min(f_1, f_2)$$

Dessa forma a função característica  $f$ , avaliada através da aplicação dos operadores *min/max*, define um particionamento do espaço onde uma das funções  $f_1$  ou  $f_2$  se igualam a  $f$ . A figura 2.1 ilustra essa fato. A função  $f$  resultante será uma função contínua (não diferenciável) por partes [PERS94].

Os operadores *min/max* não fornecem como resultado um sólido regularizado, já que são equivalentes aos operadores booleanos não regularizados. Porém são capazes de nos fornecer uma aproximação para o interior do sólido. Com base nesses operadores apenas podemos afirmar que :

$$\{ p \in \mathbb{R}^3 \mid f(p) < 0 \} \subset \text{int}(S)$$

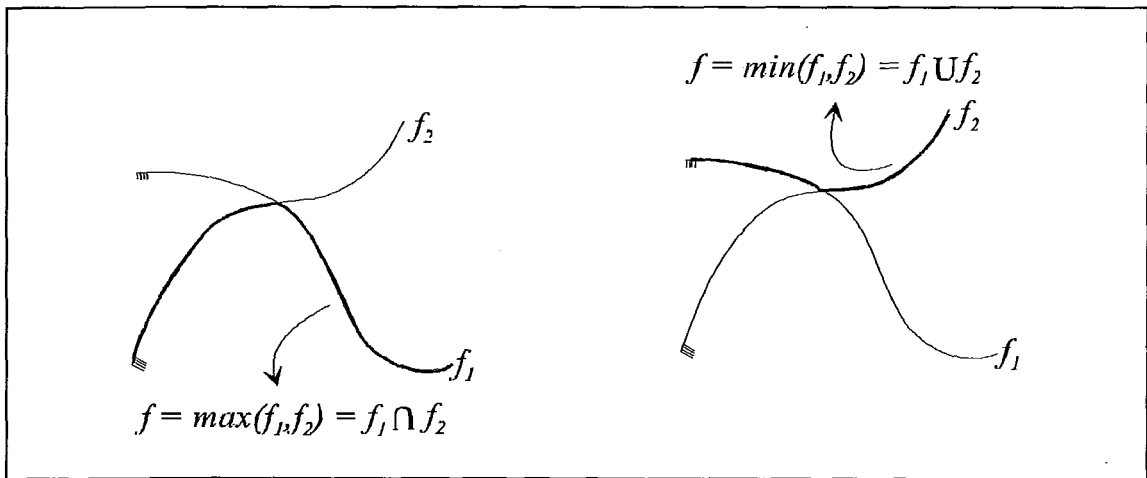


Figura 2.1 Representação da função característica resultante da aplicação dos operadores *min/max* sobre duas funções  $f_1$  e  $f_2$  em substituição aos operadores  $\cup$  e  $\cap$ .

A partir da expressão acima podemos obter a expressão regularizada do sólido pela relação [PERS94] :

$$\text{reg}(S) = \text{fecho}(\text{int}(S)) = \text{fecho}(\{p \in \mathbb{R}^3 \mid f(p) < 0\})$$

A função característica possui, em sua definição, uma propriedade particularmente interessante para nosso propósito, que é o particionamento do espaço em três regiões distintas : uma positiva (associada aos pontos localizados na região externa à primitiva), outra negativa (pontos localizados na região interna) e ainda uma região nula (pontos sobre a sua superfície). Sendo assim, dado um ponto qualquer no espaço, podemos determinar sua posição relativa à primitiva, avaliando o valor da função característica naquele ponto. Como nosso objetivo é aproximar a superfície do sólido, estamos interessados apenas naqueles pontos que anulam a função característica. Colocado nesse novo enfoque, nosso problema consiste em determinar o conjunto de raízes de uma função diferenciável por partes que define a superfície do sólido.

### 2.2.1 Nova Abordagem para o problema

A superfície de fronteira do sólido, e que queremos aproximar, está contida no conjunto de zeros da função característica do sólido, representada através dos operadores *min/max*. No entanto nem todos os zeros serão de nosso interesse, já que buscamos a representação regularizada do sólido.

Na verdade, apenas os seus zeros regulares serão de nosso interesse. Dizemos que um ponto  $p$ , qualquer, é um zero regular se em qualquer vizinhança de  $p$  existem pontos cujos valores da função característica são negativos e positivos.

Portanto nosso problema, de forma genérica, pode ser visto como a busca dos zeros regulares de uma função diferenciável por partes.

Para a solução desse problema um conjunto de métodos comumente empregado são os denominados *métodos simpliciais* [ALLG85], [ALLG87], [GOME89], [SALI91]. Eles são baseados na decomposição do espaço de interesse seguido do cálculo aproximado do conjunto de zeros da função em cada uma das regiões definidas pelo processo de decomposição.

## 2.3 Subdivisão Espacial

O objetivo básico a ser alcançado quando lançamos mão de um processo de subdivisão, seja qual for o problema a ser resolvido, é diminuir sua complexidade. Ao tratarmos pequenas porções do domínio do problema, as soluções tendem a ser mais fáceis de determinar do que quando olhamos para o problema em toda sua complexidade.

Após resolvidos todos os “subproblemas” definidos pela partição do domínio original, o próximo passo é unir essas “subsoluções” de modo que elas juntas formem a solução (mesmo que aproximada) do problema.

Nesse ponto reside a principal dificuldade desse paradigma, pois o processo de subdivisão do domínio do problema deve garantir que, quando unidas, as soluções localizadas formarão uma solução contínua para o problema. Além disso, caso a solução seja uma aproximação da solução original, devemos garantir que ela estará dentro de uma dada tolerância estabelecida.

Em nosso caso particular, o objetivo é a determinação dos zeros regulares de uma função diferenciável por partes tri-variada. O paradigma da subdivisão espacial é bem adequado a este problema, visto que essa função pode ter um comportamento global bastante complexo, porém quando restringimos a análise à porções do seu domínio, a função em geral tende a ter uma variação quase linear em sua derivada.

Regiões do domínio da superfície, onde a variação da sua derivada for elevada, serão grosseiramente aproximadas por uma solução linear. No entanto, em geral, quanto mais próximo estivermos da superfície, menos sentiremos essas variações, pois o comportamento da derivada da função será mais próximo de um comportamento linear local. Nesses casos podemos ter uma solução linear aproximada para a superfície satisfatória.



Devemos no entanto estar atentos ao fato de que isso só ocorre em superfícies cuja continuidade seja pelo menos  $C^1$ . No caso específico das funções que iremos analisar, resultado da composição das funções características de cada primitiva combinadas pelos operadores *min/max*, apenas asseguramos continuidade  $C^0$ . Portanto não necessariamente em todo o domínio da função essa característica será assegurada. Em particular nas vizinhanças de arestas e vértices, por exemplo, a proximidade da superfície não resulta em garantia de uma boa aproximação por uma única solução linear, já que nessas regiões não temos satisfeita a condição de diferenciabilidade da superfície.

## 2.4 Decomposição Celular

Nosso objetivo, como já descrito anteriormente, será “recortar” o espaço em “pedaços” que se “encaixem”, de forma conveniente, para que possamos analisá-los e obter uma solução “local” mais simples.

Para tanto apresentaremos agora as definições de como deverá ser esse “recorte” do espaço, como deverão ser os “pedaços” formados, bem como o modo pelo qual eles devem se “encaixar”.

### 2.4.1 Célula

Seja um conjunto  $V = \{ v_0, v_1, v_2, \dots, v_k \}$  de pontos tais que  $v_i \in E^n$  onde  $0 \leq i \leq k$ . Define-se a célula gerada por  $V$  como a combinação convexa afim dos pontos pertencentes ao conjunto. De outra forma :

$$\langle v_0, v_1, \dots, v_k \rangle = \left\{ u \in E^n ; u = \sum_{i=0}^k \lambda_i \cdot v_i, \lambda_i \geq 0 ; \sum_{i=0}^k \lambda_i = 1 \right\}$$

Em função da relação de dependência linear entre os pontos de  $V$  podemos ter a formação de células de diferentes tipos.

Se  $V = \{ v_0, v_1, v_2 \}$ , dependendo da relação entre os vetores  $v_0-v_1$  e  $v_0-v_2$ , podemos ter uma célula triangular (os dois vetores linearmente independentes), um segmento de reta (os dois vetores coincidentes) ou ainda um ponto (os três vértices coincidentes).

Se restringirmos a relação de dependência linear entre os pontos que formam o conjunto  $V$  de modo que todos os vetores  $u_i = v_i - v_0$ ;  $0 < i \leq k$ , sejam linearmente independentes estaremos construindo uma classe restrita de células: os simplexes.

## 2.4.2 Simplexos

Seja a célula  $\sigma = \langle v_0, v_1, \dots, v_k \rangle$ . Se os  $k$  vetores  $u_i = v_i - v_0$  ( $0 < i \leq k$ ) forem linearmente independentes a célula  $\sigma$  é dita um  $k$ -simplexo (ou *simplexo  $k$ -dimensional*) em  $\mathbb{R}^n$ . Os pontos  $v_0, v_1, \dots, v_k$  são ditos vértices do simplexo  $\sigma$ .

Os simplexos caracterizam as formas poliédricas mais simples para uma dada dimensão  $k$ . Dessa forma um  $1$ -simplexo será um segmento de reta. Um  $2$ -simplexo um triângulo, enquanto que um  $3$ -simplexo será um tetraedro.

A partir de um simplexo  $\sigma$ , de dimensão  $k$ , podemos construir  $l$ -simplexos  $\sigma_l = \langle v_0, v_1, \dots, v_l \rangle$  onde  $l < k$ , formados pelo agrupamento de  $l$  vértices de  $\sigma$ . Esses  $l$ -simplexos são denominados *subsimplexos de  $\sigma$* . Em casos particulares, como em  $\mathbb{R}^3$ , subsimplexos possuem denominações específicas. É o caso dos subsimplexos de dimensão 1 denominados arestas de  $\sigma$  e dos subsimplexos de dimensão 2 chamados de faces de  $\sigma$ . De um modo geral um subsimplexo de dimensão  $l$  ( $l < k$ ) é nomeado *l-face* de  $\sigma$ , ou simplesmente *face de  $\sigma$* . No  $\mathbb{R}^n$  os  $(n-l)$ -simplexos são denominados *facetras* de  $\sigma$  (para  $l > 0$ ).

A partir das definições acima podemos agora determinar qual serão as características dos “pedaços” que comporão a nossa subdivisão espacial. Como nossos objetos de estudo (a superfície implícita da forma  $f(x_1, x_2, x_3) = 0$ , onde  $f: \mathbb{R}^3 \rightarrow \mathbb{R}$ ) estão imersos no  $\mathbb{R}^3$  nossos elementos básicos no particionamento do espaço serão os  $3$ -simplexos, ou seja, *tetraedros*.

Falta agora definir de que forma esses  $3$ -simplexos estarão dispostos no espaço, ou seja, como eles devem se “encaixar” de tal forma que as soluções obtidas localmente possam ser agrupadas para construir uma solução contínua (mesmo que aproximada) do problema.

### 2.4.3 Decomposição Celular

Seja um conjunto de células  $C = \{ c_1, c_2, c_3, \dots, c_k \}$  onde  $k \in \mathbb{N}$ . Seja um conjunto  $D \subset E^n$ . Dizemos que  $C$  é uma decomposição celular do conjunto  $D$ , caso as propriedades abaixo sejam satisfeitas.

- $D = \cup c_i$
- $c_i \cap c_j = \emptyset \wedge c_i \cap c_j = \text{face comum a } c_i \text{ e } c_j \quad \forall i, j \quad i \neq j$

Um caso particular de decomposições celulares se dá quando todas as células da decomposição são simplexos. A essa particular decomposição denominamos *triangulação*. O conjunto  $C$  nesse caso recebe o nome de *complexo simplicial*. A figura 2.2 mostra duas decomposições do espaço, uma celular e a outra uma decomposição qualquer.

A segunda condição imposta pela definição de decomposição celular, caracteriza como os tetraedros da nossa partição do domínio irão se encaixar. Se cada simplexo compartilha com o elemento adjacente exatamente uma face, garantimos que os elementos irão se "encaixar" perfeitamente um ao outro.

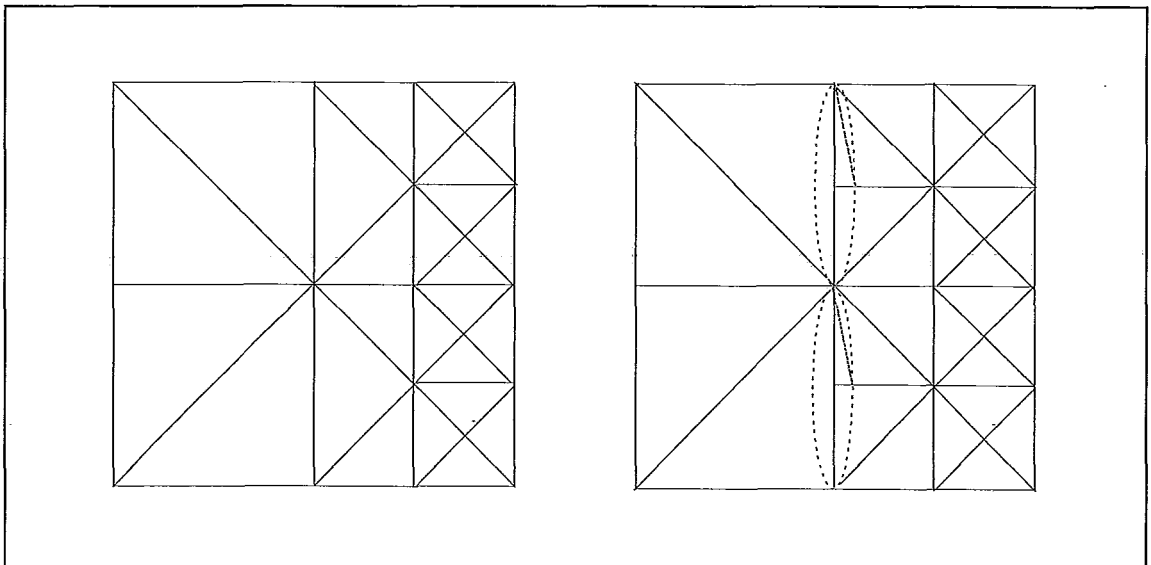


Figura 2.2 Decomposições do espaço bidimensional : a direita uma decomposição qualquer, e a esquerda uma decomposição celular. Em destaque interseções de células que não são faces comuns.

Esse fato será particularmente útil pois irá garantir continuidade quando da composição das soluções parciais geradas em cada elemento da decomposição celular. No nosso caso particular, essa condição nos permite assegurar que a aproximação linear

gerada para uma determinada célula irá se “encaixar” perfeitamente com a aproximação gerada na célula adjacente, já que ambas irão se utilizar de informações comuns à face que compartilham.

## 2.4.4 Triangulações

Uma triangulação pode ser regular ou não de acordo com a distribuição dos simplexos que a formam. Em uma triangulação não regular a forma dos simplexos que a compõem é variável. Dentro dessa classe de triangulações sem dúvida a mais conhecida e utilizada é a triangulação de Delaunay. Esse tipo de triangulação tem sido aplicada em diversas áreas tais como Modelagem de Terrenos [NEVE88], Morphing de Imagens [BRIT94] e de Contornos [OLIV93], entre outras.

As triangulações não regulares são em geral adotadas quando se tem algum tipo de informação a respeito do domínio a ser particionado. Nesses casos podemos guiar a criação dos simplexos com base na distribuição da informação pelo domínio do problema. Dessa forma regiões com alta taxa de informação irão ter uma maior concentração de simplexos que regiões com baixa taxa de informação.

O caráter não uniforme das triangulações não regulares traz como desvantagem a necessidade de se manter alguma estrutura adicional para armazenar as informações de adjacência entre os simplexos. Tal informação é imprescindível para que possamos efetuar operações como percurso na triangulação, inserção e remoção de simplexos, etc.

As triangulações regulares podem ser vistas como um caso particular das triangulações não regulares. Nelas todos os simplexos tem o mesmo formato. Seu uso é adequado quando não temos informação alguma sobre o domínio do problema a priori. Nesses casos temos que de qualquer modo vasculhar todo o espaço, o que se torna mais simples se feito de forma regular. Em triangulações desse tipo não se faz necessário a manutenção de nenhuma estrutura auxiliar, já que sua própria disposição regular nos permite, de forma implícita e simples, obter as relações de adjacência entre os simplexos.

Sua robustez no entanto é mais baixa que a das triangulações não regulares, justamente pelo fato de termos que distribuir seus elementos de forma uniforme pelo domínio. Para obtermos um detalhamento de uma dada região teremos que propagá-lo por todo o domínio em estudo.

Em nosso caso iremos partir de uma triangulação regular do domínio da função a ser aproximada. Essa escolha é fruto da dificuldade de determinarmos onde se localizam os zeros da função em estudo. Nesse caso se torna mais simples avaliar de forma regular todo o seu domínio procurando detectar por onde o conjunto de zeros da função estende-se.

## 2.5 Modelo de Interpolação Linear

Com base em uma triangulação regular, os métodos simpliciais constroem uma aproximação linear para representar o comportamento da função dentro do domínio da triangulação.

Utilizando a estrutura da triangulação podemos montar um conjunto inicial de pontos dos quais conhecemos o valor da função a ser aproximada. Esse conjunto de pontos é formado pelos vértices da triangulação. Seus valores podem ser obtidos através da avaliação da função a ser aproximada em cada vértice. Com base nesses valores podemos, em cada simplexo, construir um interpolador linear de modo a aproximar os valores de cada um dos pontos em seu interior.

Essa tarefa é facilitada por uma característica decorrente da definição dos simplexos : cada um define através de seus vértices uma base geradora do espaço. Podemos portanto descrever qualquer ponto  $p$  do espaço como uma combinação afim de seus vértices  $v_0, v_1, \dots, v_n$ . Ou seja :

$$p = \sum_{i=0}^n \lambda_i v_i \quad \text{onde} \quad \sum_{i=0}^n \lambda_i = 1$$

A partir dessa nova representação para os pontos do espaço podemos definir um novo sistema de coordenadas, baseado nos vértices do simplexo. A  $(n+1)$ -tupla  $(\lambda_0, \lambda_1, \dots, \lambda_n)$  são o que chamamos de *coordenadas baricêntricas do ponto  $p$* .

Portanto, dado um ponto qualquer do domínio, podemos calcular suas coordenadas baricêntricas e, com base nessas, obter o valor da função nesse ponto. Para isso basta que uma combinação afim dos seus valores nos vértices do simplexo seja feita. Com isso somos capazes de construir um interpolador linear para o comportamento da função dentro de um simplexo. Dado um ponto  $p$  qualquer com coordenadas baricêntricas  $(\lambda_0, \lambda_1, \lambda_2, \dots, \lambda_n)$  qualquer podemos determinar o valor do interpolador linear  $L$  da função  $F$  nesse ponto através da relação :

$$L(p) = \sum_{i=0}^n F(v_i) \cdot \lambda_i$$

O conjunto dos interpoladores definidos em cada simplexo irá nos fornecer uma aproximação linear por partes contínua da função em todo o domínio da triangulação. Esse fato pode ser garantido pois partimos da hipótese de que nossa decomposição do espaço é uma triangulação. Dessa forma as informações utilizadas para construção do interpolador na fronteira comum a dois simplexos é igual para ambos, não gerando portanto descontinuidades.

## 2.5.1 Geração da Poligonização

Definido um interpolador linear que nos fornece uma aproximação linear do comportamento da função em todo o domínio, podemos agora buscar quais simplexes da triangulação possuem parte do conjunto de zeros desse interpolador passando em seu interior.

O conjunto de zeros pode ser visto como a fronteira entre duas regiões do espaço: uma definida pelo conjunto de pontos cuja avaliação pelo interpolador resulta em valores positivos e outra cujos valores são negativos. Como base nessa variação, podemos definir um critério simples de aferição que nos permite decidir se um dado simplexo possui em seu interior pontos que pertencem ao conjunto de zeros definidos pelo interpolador.

Partindo desses simplexes, que possuem parte do conjunto de raízes da função em seu interior, podemos, utilizando o interpolador linear já definido, construir um novo conjunto de zeros, aproximação linear do conjunto de zeros original. Com base nessa aproximação linear podemos construir uma nova superfície, linear por partes, que aproxima a função nesse domínio.

Se ao avaliarmos os valores do interpolador nos vértices de um simplexo obtivermos valores positivos e negativos, podemos afirmar que parte do conjunto de raízes do interpolador se encontra dentro desse simplexo. Denominaremos esses simplexes de *interseptantes*, já que eles possuem parte da superfície, definida pelo conjunto de pontos que anulam o interpolador, em seu interior. Fica claro que, caminhando por um segmento de reta que ligue dois desses vértices, em algum ponto nesse segmento iremos cruzar a fronteira entre o lado positivo e o lado negativo. Esse ponto pertence portanto ao conjunto de zeros definido pelo interpolador. Podemos garantir que isso ocorre pois a função que estamos aproximando é suposta contínua, de classe  $C^0$ , em todo o domínio.

Em primeira análise, o caso em que valores desse interpolador nos vértices do simplexo possuem todos o mesmo sinal, pode nos levar a concluir que ele está totalmente imerso em uma das duas regiões, positiva ou negativa, definidas pelo interpolador. Esses simplexes, portanto poderiam ser ignorados, visto que não possuem pontos da superfície definida pelos zeros do interpolador.

Porém, com uma análise mais cuidadosa, podemos constatar que mesmo nesses casos é possível que parte do conjunto solução intercepte o simplexo. A figura 2.3 mostra essa situação analisando o caso bidimensional. No momento, esses casos serão propositadamente ignorados. Sendo assim, nessa situação a poligonização acaba atuando como um filtro, eliminando da aproximação parte do conjunto de zeros do interpolador que não são detectáveis para uma dada resolução da triangulação. Durante

o decorrer do trabalho retornaremos a essa questão estabelecendo um critério um pouco mais preciso para a classificação dos simplexos.

Nesse momento também estamos simplificando o problema presumindo que nenhum dos vértices da triangulação anula a função a ser aproximada. No decorrer do trabalho entraremos em mais detalhes sobre esse caso, explorando o problema e propondo uma solução.

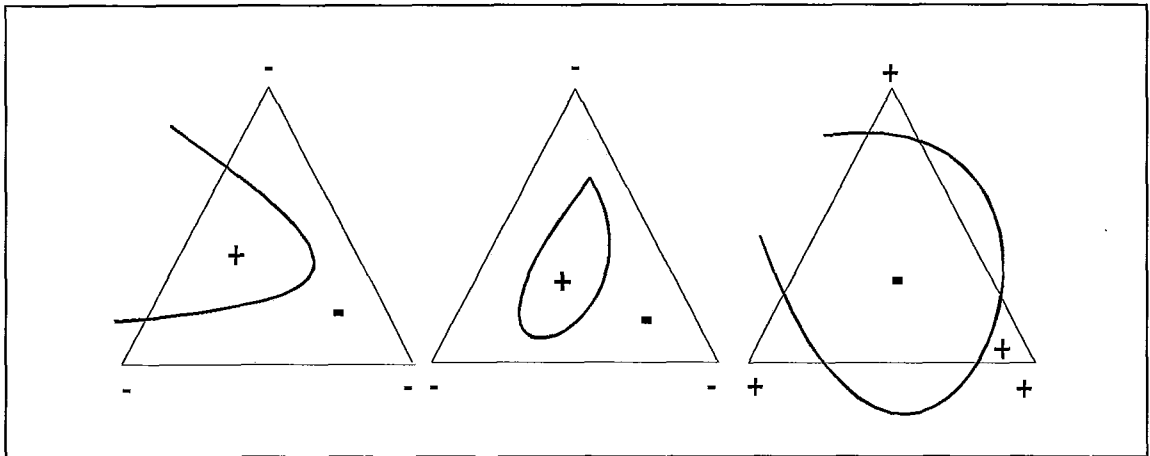


Figura 2.3 Três exemplos de situações em que a análise dos sinais da função nos vértices do simplexo não permitem caracterizar a passagem de parte do conjunto de raízes pelo seu interior.

Portanto, com base na análise dos valores do interpolador nos vértices da triangulação, temos meios de definir em quais simplexos temos parte do conjunto de raízes da aproximação linear. A partir desse conjunto de simplexos iremos construir nossa poligonização.

Analisando os pares de vértices do simplexo que possuem valores associados cujos sinais são opostos, podemos aplicar um interpolador linear e obter um ponto  $p$ , pertencente ao segmento que os une, que se encontra sobre a superfície definida pelos zeros do interpolador linear  $L$ . Tal ponto pode ser obtido pela relação :

$$\alpha = \frac{L(v_i)}{L(v_i) - L(v_j)}$$

$$p = (1 - \alpha) \cdot v_i + \alpha \cdot v_j$$

O valor de  $\alpha$  irá atuar como um indicador de qual vértice está mais próximo da raiz localizada sobre o segmento por eles definido, com base nos valores  $L(v_i)$  e  $L(v_j)$  do interpolador. Determinado esse valor  $\alpha$  aplicamos um interpolador afim sobre as coordenadas cartesianas dos vértices  $v_i$  e  $v_j$ , obtendo assim as coordenadas do ponto

onde se encontra a raiz do interpolador sobre aquele segmento, e que aproxima a raiz da função original.

Para simplificar a construção do polígono aproximador, podemos analisar apenas algumas das combinações, dois a dois, dos vértices cujos valores da função tem sinais opostos. Em particular, podemos tomar como segmentos base para essa análise as arestas que ligam esses pares de vértices. Dessa forma, obteremos pontos sobre as arestas que conectados nos fornecerão um polígono de no mínimo três ou no máximo quatro lados (já que iremos descartar os possíveis casos degenerados, em que o polígono se reduziria a uma aresta ou a um vértice), como mostra a figura 2.4.

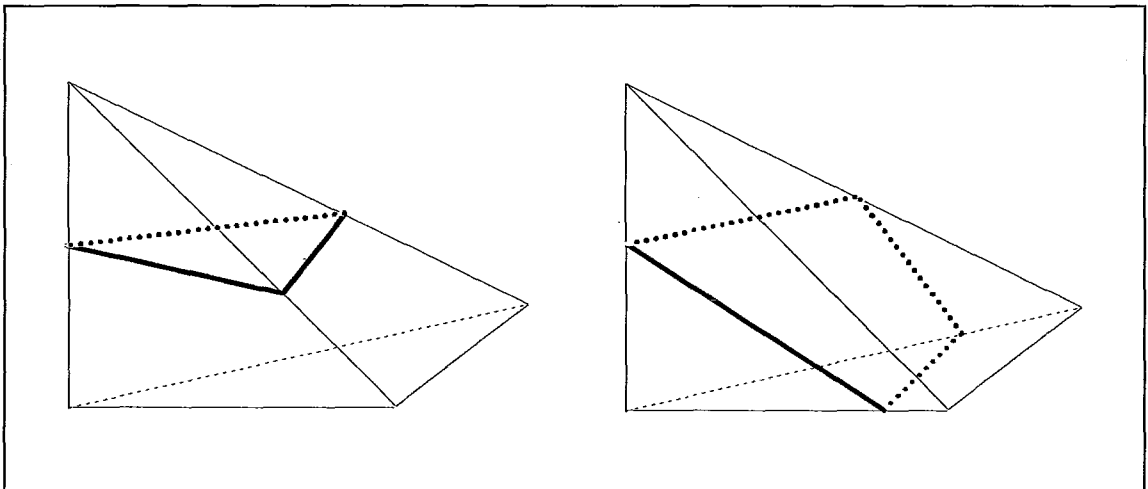


Figura 2.4 Duas situações possíveis na geração da aproximação poligonal do conjunto de raízes da função em um simplexo.

## 2.5.2 Algoritmo

A partir das idéias expostas até aqui podemos elaborar um primeiro algoritmo baseado nos métodos simpliciais. Sua descrição em pseudocódigo é apresentada a seguir.

*ConstroiAproximaçãoLinear(Domínio);*

*Triangulação = CalculaTriangulação(Domínio);*

*para cada simplexo i da Triangulação faça*

*se (ClassificaSimplexo(Simplexo<sub>i</sub>, Função) = Interceptante) então*

*CalculaAproximaçãoLinear(Simplexo<sub>i</sub>, Função);*

*Erro = MAX(CalculaErroAproximação(Simplexo<sub>i</sub>, Função), Erro);*



O algoritmo descrito acima tem início pela construção de uma triangulação regular do domínio da função a ser aproximada. A partir dessa triangulação cada simplexo é classificado de modo que possamos determinar se existe ou não zeros do interpolador, em seu interior.

Será com base nessa classificação que iremos ou não acionar a função de cálculo da aproximação linear naquele simplexo. Essa rotina por sua vez irá determinar o conjunto de arestas que interceptam o conjunto de zeros do interpolador linear definindo sobre a triangulação. Com base na interpolação apresentada na seção anterior, os pontos de interseção da superfície definida pelos zeros do interpolador e as arestas serão calculados, e formarão o polígono aproximador.

A partir da poligonização construída em cada simplexo, podemos estimar o erro que estamos cometendo na aproximação linear. Supondo que temos o valor exato do conjunto de raízes da função original dentro do simplexo, podemos calcular o erro daquela aproximação determinando a diferença entre o conjunto de raízes exatas e o conjunto definido pela aproximação linear. Com o erro global associado à triangulação determinaremos a maior dessas diferenças.

Na hipótese desse erro estar além da tolerância estabelecida, uma nova aplicação do algoritmo se fará necessária, agora com um grau de refinamento maior.

## ***2.6 Nova Proposta***

Analisando com mais atenção o algoritmo apresentado podemos claramente identificar alguns pontos que podem ser aprimorados.

O principal deles está ligado ao uso de uma triangulação regular do domínio da função a ser aproximada. Apesar de tornar o algoritmo mais simples, esse tipo de triangulação pode aumentar não só o trabalho para obtenção de um resultado dentro de nossa tolerância, como também a manutenção de informações topológicas associadas à triangulação.

A utilização de uma triangulação regular descarta dois fatos de vital importância no processo de busca da aproximação linear de uma função. O primeiro se relaciona com o fato de que, em geral, o conjunto de zeros regulares da função a ser aproximada não ocupa totalmente o domínio no qual ela está definida. Utilizando uma triangulação regular teremos um esforço inútil de refinar áreas do domínio que não contêm parte do conjunto de raízes da função até que um dado grau de refinamento seja alcançado. Só mais tarde, durante a análise de cada simplexo da triangulação iremos determinar se o conjunto de zeros regulares da função passa ou não por aquela região do domínio.

Outro fato importante que não é levado em conta dentro do algoritmo é que, na maioria dos casos, o comportamento da função a ser aproximada não é uniforme dentro do domínio. Ou seja em determinadas regiões do domínio, interceptadas pelo conjunto de zeros regulares da função, esta pode, com um nível baixo de refinamento, alcançar uma solução linear que satisfaz nossa condição de precisão. Outras regiões, no entanto, podem requerer um nível maior de refinamento para satisfazer ao mesmo grau de precisão. Nesse sentido, o fato de utilizarmos uma triangulação regular força que o nível de refinamento seja definido pelo pior caso. Ou seja, o nível de refinamento de toda a triangulação será ditado pelo grau de refinamento da região de mais alto erro na substituição da função pela sua aproximação linear.

Visando tornar o processo de construção da poligonização mais eficiente, nossa proposta irá construir um novo algoritmo no qual as características do problema acima descritas são levadas em conta.

A proposta que apresentaremos baseia-se na junção do processo de construção da triangulação com o de avaliação da condição em cada simplexo, tanto no que diz respeito à inclusão do conjunto de zeros regulares da função a ser aproximada no interior de cada simplexo, quanto à qualidade da aproximação linear obtida nesse simplexo. Desse modo durante a construção da triangulação podemos decidir se um dado simplexo possui ou não parte do conjunto de zeros regulares da função a ser aproximada em seu interior. Com isso podemos evitar refinamentos desnecessários nesses elementos.

De modo análogo, durante esse processo também iremos avaliar a condição de erro de cada simplexo decidindo se podemos ou não abortar o processo de refinamento para aquela região.

Fica claro que para tal algoritmo uma triangulação regular do domínio não mais será adequada. Utilizaremos nesse caso uma *triangulação adaptativa* do domínio da função. Essa triangulação irá se caracterizar pelo refinamento não uniforme de seus simplexos.

Nos próximos capítulos discutiremos como construir esse tipo de triangulação. Além disso, apresentaremos os algoritmos que irão incorporar tanto o caráter de seleção dos simplexos que possuem parte dos zeros da função a ser aproximada, quanto um critério para aferição do erro cometido na aproximação dentro de cada simplexo durante o processo de construção da triangulação.

## 3. SUBDIVISÃO ESPACIAL SELETIVA

---

### 3.1 Introdução

Nesse capítulo apresentaremos uma classe de triangulações regulares que forma a base para nosso trabalho : as triangulações CFK. Aproveitando suas propriedades, iremos estender essa classe de modo a construir uma triangulação capaz de suportar graus de refinamento diferentes em sua estrutura.

Discutiremos também um primeiro algoritmo, proposto por [SALI92], que, com base nessa extensão das triangulações CFK, permite introduzir um critério seletivo ao processo de construção da triangulação. Esse critério permite que simplexes identificados como não interceptantes, ou seja, que não contêm parte do conjunto de zeros da função a ser aproximada, sejam eliminados do processo de refinamento.

Esse será o primeiro passo no sentido de criar uma triangulação que seja refinada apenas onde existam raízes da função e apenas o necessário para alcançar a precisão desejada.

### 3.2 Triangulações CFK

#### 3.2.1 Definição

As triangulações CFK se enquadram dentro das chamadas triangulações regulares do espaço. Seu nome tem como origem uma homenagem aos seus criadores, que embora estudando problemas distintos, chegaram à mesma estrutura : Coxeter, Freudenthal e Kuhn.

Nessa classe de triangulações, todos os simplexes são congruentes e podem ser obtidos partindo-se de um único simplexo inicial, replicado pelo espaço de interesse a partir de reflexões ou translações sobre suas faces.

O trabalho de construção de uma triangulação CFK está baseado em primeiro decompor o espaço de forma regular, em células cúbicas. Em seguida, triangulamos uma

de suas células e reproduzimos tal triangulação para as demais células, através de suas relações de adjacências.

Para construir essa decomposição celular regular inicial podemos nos valer de outro tipo de célula, também bastante simples : o cubo (no caso tridimensional). Os cubos são largamente utilizados na tarefa de subdividir o espaço de forma regular (como no caso do esquema de Decomposição Espacial) devido à sua orientação simples, quando trabalhamos com o sistema de coordenadas cartesianas. Portanto, nosso primeiro passo será decompor o espaço de forma também regular, em uma malha de cubos.

Partindo dessa decomposição inicial em cubos, construiremos a triangulação de uma de suas células. Para isso subdividimos o cubo em 6 tetraedros utilizando 3 planos de corte. Cada plano irá cortar um par de faces opostas por suas diagonais, como mostra a figura 3.1 . Note que uma das diagonais do cubo será uma aresta comum a todos os tetraedros gerados.

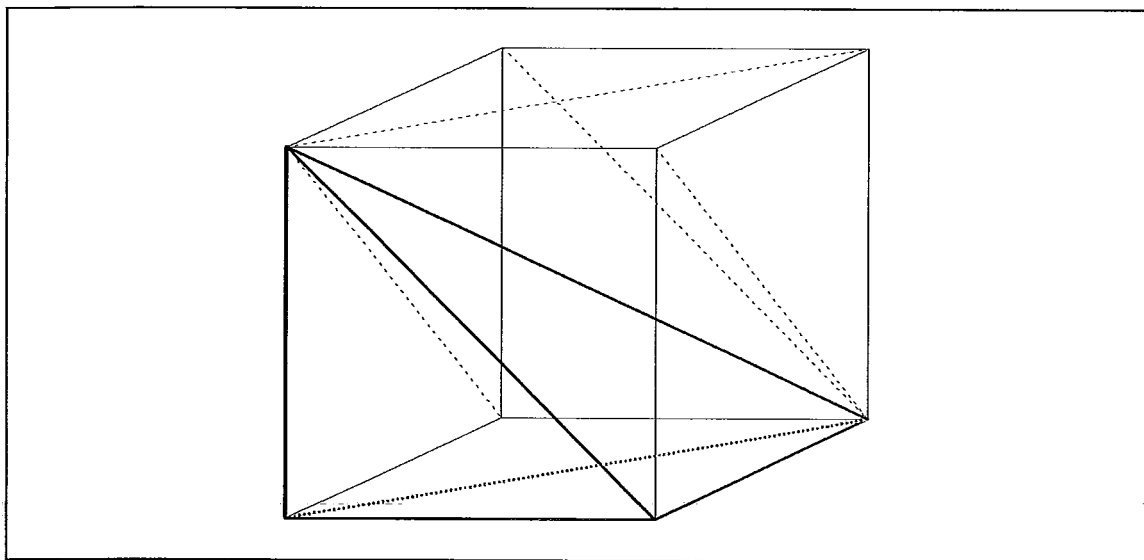


Figura 3.1 Subdivisão da célula cúbica inicial em seis tetraedros.

Com a triangulação do cubo, podemos agora replicá-la pelos cubos adjacentes cobrindo todo o espaço. A tarefa de repassar essa triangulação base pode ser feita de duas formas : através de translações (triangulações K1) ou reflexões (triangulações J1). Exemplos desses dois tipos de triangulação são vistos, no caso bidimensional, na figura 3.2 .

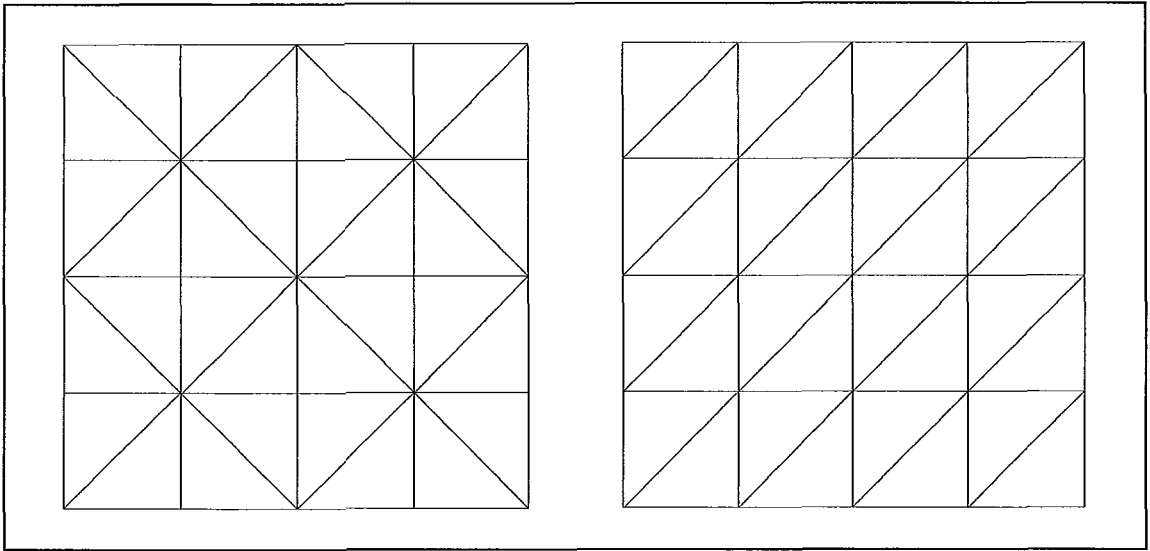


Figura 3.2 Triangulações CFK bidimensionais. À esquerda uma triangulação J1, obtida por reflexões da triangulação da célula inicial. À direita uma triangulação K1, resultado da translação da triangulação da célula inicial.

### 3.2.2 Variações de uma Triangulação CFK

Podemos gerar mais dois tipos de triangulação por meio de *subdivisões binárias* nos simplexos das triangulações CFK, K1 e J1. A *subdivisão binária* de um simplexo promove a criação de dois outros simplexos, semelhantes entre si e não congruentes ao original. Esse dois simplexos são obtidos através de um corte adequado no simplexo origem. Esse corte, no caso tridimensional, é feito por um plano que passa pela aresta oposta a maior aresta do simplexo e o ponto médio desta. A figura 3.3 mostra esse processo.

Em uma triangulação J1 ou K1, todos os simplexos têm como maior aresta a diagonal do cubo que os contém. As demais arestas são definidas ou por uma diagonal da face ou por um lado do cubo.

Se aplicarmos uma subdivisão binária sobre um simplexo da triangulação J1 ou K1, um novo tipo de simplexo será criado. Sua maior aresta será agora a diagonal de uma face do cubo, e a menor a metade da diagonal do cubo. A forma desses simplexos é representada na figura 3.4. A aplicação uniforme da subdivisão binária sobre todos os simplexos da triangulação J1 ou K1 irá resultar em uma outra triangulação que denominaremos de S1.

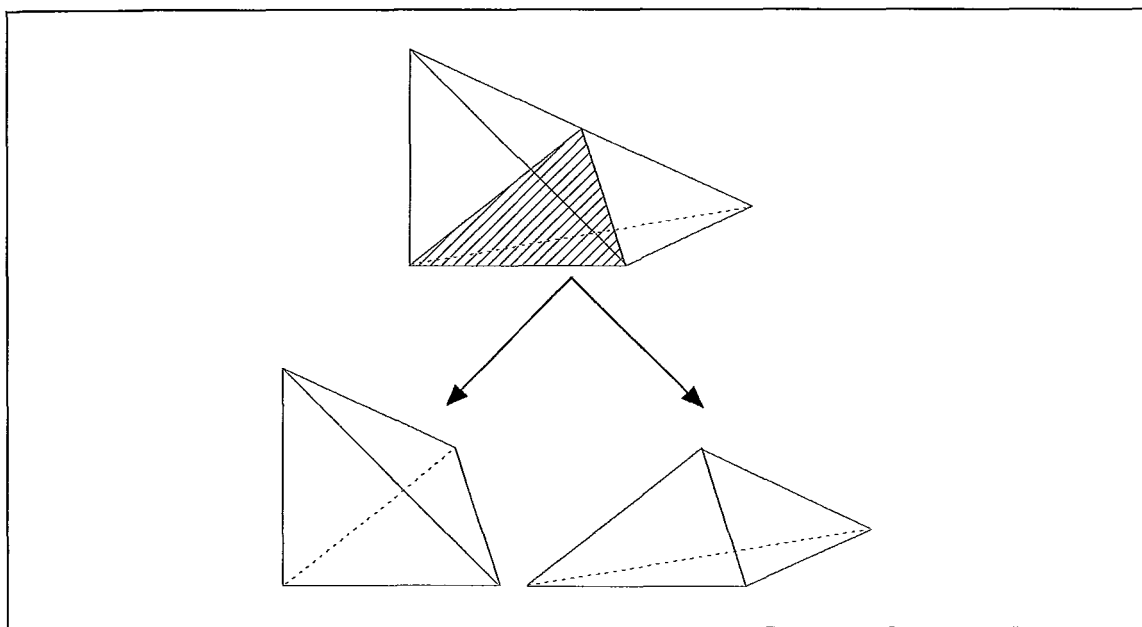


Figura 3.3 Subdivisão binária de um tetraedro.

Promovendo uma nova subdivisão binária sobre os simplexes da triangulação  $S_1$ , um novo tipo é gerado. Um simplexo dessa nova triangulação, denominada  $R_1$ , tem como maior aresta o lado do cubo e menor aresta a metade da diagonal da face. A figura 3.4 mostra um desses simplexes.

Se uma nova subdivisão binária partindo da triangulação  $R_1$  for feita iremos obter simplexes cuja maior aresta é a metade da diagonal do cubo e sua menor aresta é metade do lado do cubo. Analisando atentamente esses novos simplexes veremos que todas as suas arestas tem exatamente a metade do comprimento das arestas dos simplexes da triangulação  $J_1$  ou  $K_1$  inicial. Em outras palavras, os novos simplexes gerados terão dimensões equivalentes à metade dos simplexes originais. Com a continuidade do processo de subdivisão binária os demais simplexes (das triangulações  $S_1$  e  $R_1$ ) também irão se repetir sempre com suas dimensões reduzidas a metade a cada subdivisão.

Podemos definir então três tipos básicos de simplexes possíveis dentro de um processo de subdivisão binária de uma triangulação CFK inicial ( $J_1$  ou  $K_1$ ) :

- tipo A - representam  $1/6$  do volume do cubo inicial. Pertencem a uma triangulação  $J_1$  ou  $K_1$ ;
- tipo B - representam  $1/12$  do volume do cubo inicial. Pertencem a uma triangulação  $S_1$ ;
- tipo C - representam  $1/24$  do volume do cubo inicial. Pertencem a uma triangulação  $R_1$ ;

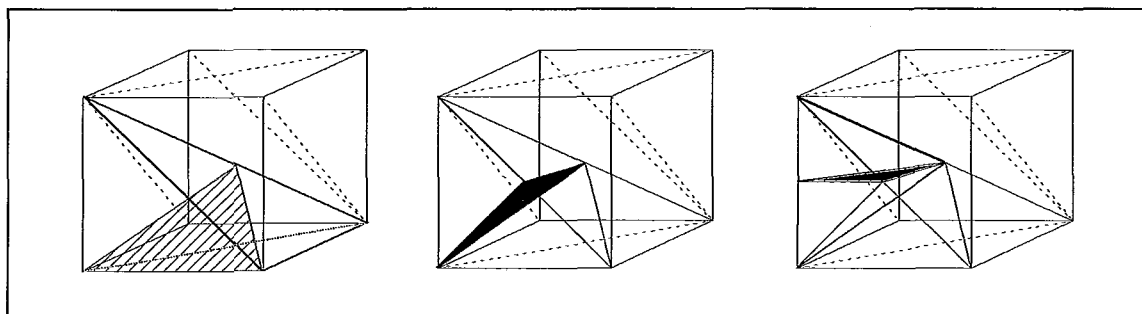


Figura 3.4 Seqüência de subdivisões binárias. À esquerda um simplexo de tipo A (triangulação J1 ou K1) subdividido em dois simplexos do tipo B (triangulação S1). Ao centro a subdivisão de um simplexo de tipo B (triangulação S1) gerando dois simplexos de tipo C (triangulação R1). À direita a subdivisão binária de um simplexo de tipo C gerando dois simplexos de tipo A (triangulação J1).

Um fato que deve ser ressaltado é que a despeito da triangulação inicial (J1 ou K1), após três subdivisões binárias chegamos a simplexos cujas dimensões são metade das dos simplexos originais, porém esses formam sempre uma triangulação J1. Tal fato pode ser observado na figura 3.4. Os simplexos são sempre reflexões de seus vizinhos.

A aplicação do processo de subdivisão binária sobre uma triangulação CFK original pode ser visto como um processo de refinamento uniforme, já que os simplexos sempre diminuem uniformemente de tamanho a cada aplicação.

Uma das características das triangulações geradas pela subdivisão binária, é que nelas não há alterações substanciais da forma dos simplexos originais. Tal fato é importante em várias aplicações em que a forma dos simplexos está intimamente ligada à qualidade da solução que será obtida para o problema, como, por exemplo, em elementos finitos.

Uma medida bastante utilizada para avaliar quão boa é a forma dos simplexos (tetraedros) de uma triangulação é o valor do menor de seus ângulos sólidos internos. Esse valor nos dá uma noção de quanto o simplexo é “bem formado”. Tetraedros degenerados tem o menor ângulo sólido igual a zero, ao passo que tetraedros regulares tem a maior medida possível para o seu menor ângulo sólido.

Simplexos muito “magros” são ruins para diversas aplicações. É o caso de elementos finitos, onde tetraedros muito alongados em uma malha induzem a instabilidade no cálculo da solução do sistema de equações diferenciais associado ao fenômeno em estudo.

Na verdade, a qualidade dos simplexos obtidos pelo processo de subdivisão binária irá depender fortemente das células que compõem a decomposição celular inicial. Caso a decomposição seja formada por células retangulares, dependendo da relação entre suas dimensões, poderemos partir de uma triangulação inicial pouco conveniente. Como o processo de subdivisão binária não altera substancialmente o

formato dos simplexos, a triangulação em qualquer de seus estágios terá simplexos cujos formatos podem não ser interessantes para a aplicação.

### ***3.3 Subdivisão Seletiva do Espaço***

As definições apresentadas até aqui representam o ferramental básico para alcançarmos nosso objetivo : construir uma aproximação linear por partes do conjunto de zeros de uma função definida na forma implícita.

Apresentaremos agora um algoritmo proposto por [SALI91] para a construção dessa aproximação linear baseada na subdivisão seletiva do espaço. Como resultado o algoritmo produz uma aproximação linear, que é uma variedade implícita, construída a partir da triangulação que cobre toda a superfície.

Vale aqui fazer uma ressalva quanto à denominação desse algoritmo. Em seu trabalho, Salim o classifica como um algoritmo adaptativo, na medida em que o processo de subdivisão é feito apenas em regiões onde a função é efetivamente definida. Nosso trabalho também se baseia na elaboração de um algoritmo que subdivide o domínio adaptativamente apenas em regiões que contém raízes da função. Porém nossa idéia é ampliar esse conceito de forma a também levar em consideração a variação da derivada da função a ser aproximada como guia para o processo de adaptatividade.

Portanto, já que os dois algoritmos são adaptativos, em diferentes gradações, para evitar dúvidas quanto a abrangência do termo dentro do trabalho, a partir daqui denominaremos o algoritmo definido por Salim como sendo seletivo, já que a sua adaptatividade reside na seleção de regiões que deverão ou não ser refinadas. Deixaremos o termo adaptativo apenas para os casos em que nos referenciarmos a adaptatividade com relação ao comportamento da superfície, ponto central de nossa nova proposta.

#### **3.3.1 A Triangulação Inicial**

Como já vimos, uma triangulação CFK pode ser construída a partir da decomposição do espaço de interesse em células cúbicas. Tomando uma dessas células e triangulando-as, geramos 6 tetraedros. Replicando-se esses tetraedros pela decomposição obtemos uma triangulação CFK do tipo J1 (por reflexões) ou K1 (por translações).

A decomposição celular inicial, na qual o algoritmo proposto se baseia, é representada pela Caixa Envolvente do domínio no qual a superfície implícita está definida. Portanto, partimos de uma decomposição celular cúbica particular, composta



por uma única célula. Baseado nessa célula inicial iremos gerar os 6 tetraedros que irão compor nossa triangulação inicial.

### 3.3.2 O Processo de Subdivisão Seletivo

Com a triangulação inicial definida, iremos refinar cada um dos seis simplexes iniciais, aplicando sobre eles o processo de subdivisão binária, de forma recursiva. Assim que um simplexo de tamanho adequado à precisão estabelecida para a aproximação seja atingido, o processo de subdivisão é interrompido e a aproximação linear calculada.

Durante o processo de subdivisão, os simplexes de interesse são aqueles que contêm parte dos zeros da função. Os demais, por não possuírem pontos sobre a superfície definida pelo conjunto de raízes da função em seu interior, não serão relevantes dentro de nosso contexto. Portanto, a seletividade do algoritmo reside em eliminar do processo de subdivisão binária os simplexes que não interceptam essa superfície.

### 3.3.3 O Algoritmo

A seguir apresentaremos o algoritmo para promover a subdivisão tal como descrita acima. Os simplexes iniciais, resultantes da triangulação do cubo inicial, são definidos pelos  $\sigma_i$ .

*Aproximação Linear(BoundingBox, Superfície, NívelMax)*

*TriangulaBoundingBox( $\sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5$ );*

*para cada  $\sigma_I$  com  $I=0$  até 5*

*GeraAproximaçãoLinear( $\sigma_I, Superfície, NívelMax$ );*

*GeraAproximaçãoLinear( $\sigma, Superfície, Nível$ )*

*se ( $\sigma \cap Superfície \neq \emptyset$ ) então*

*se ( $Nível = 0$ ) então*

*CalculaAproximaçãoLinear( $\sigma, Superfície$ );*

*senão*

*SubdivideSimplexo( $\sigma, \sigma_{F1}, \sigma_{F2}$ );*

*GeraAproximaçãoLinear( $\sigma_{F1}, Superfície, Nível-1$ );*

*GeraAproximaçãoLinear( $\sigma_{F2}, Superfície, Nível-1$ );*

### 3.3.4 Considerações sobre o Algoritmo

Fazendo uma análise do algoritmo apresentado podemos ressaltar alguns pontos relevantes sobre os quais iremos formular nossa nova proposta. Primeiro, podemos identificar que a seletividade do espaço a ser refinado é obtida através do processo de recursão. Caso um tetraedro  $\sigma$  não contenha parte da superfície a ser aproximada, a recursão é abortada. Nessa classificação uma restrição é imposta : as funções características devem ser polinomiais para que o critério de interferência possa ser aplicado. Assim sendo, a partir desse ponto assumiremos que as funções características são sempre definidas por polinômios tri-variados. Quando da discussão mais aprofundada desse critério de classificação ficará clara a necessidade dessa restrição.

Também observamos que todos os simplexes que irão gerar aproximações lineares estarão sempre no último nível do processo de subdivisão. Isso pelo fato de que a recursão é limitada pelo nível da subdivisão (representado pela variável *Nível* da função *GeraAproximaçãoLinear*). Vale lembrar que o nível da subdivisão está diretamente associado com o tamanho dos tetraedros, já que a cada subdivisão binária de um tetraedro “pai” são gerados dois “filhos” com volume reduzido à metade.

Outro ponto interessante a ser destacado é que o próprio processo de recursão utilizado no algoritmo se encarrega de alocar e desalocar espaço em memória para o armazenamento dos tetraedros. Associado a este fato, nenhuma informação topológica é acrescida à estrutura de dados, já que durante o processo não se faz necessário conhecer as relações entre simplexes vizinhos. Tão logo um simplexo atinge seu nível máximo de refinamento, sua aproximação linear é calculada, de forma independente dos seus “vizinhos”.

A figura 3.5 apresenta qual seria, em duas dimensões, o resultado da aplicação desse algoritmo para a construção da aproximação linear de uma curva qualquer. O nível máximo adotado para subdivisão foi 8.

A partir da análise do exemplo, nos deparamos com uma característica muito importante desse algoritmo. Se considerarmos o conjunto de todos os simplexes gerados durante o processo, esse conjunto não forma uma triangulação. Porém, se considerarmos o subconjunto dos simplexes que contêm a fronteira do sólido, este sim forma uma triangulação. Essa característica em nada impacta a solução, já que apenas os simplexes interceptantes serão processados para a geração da aproximação linear.

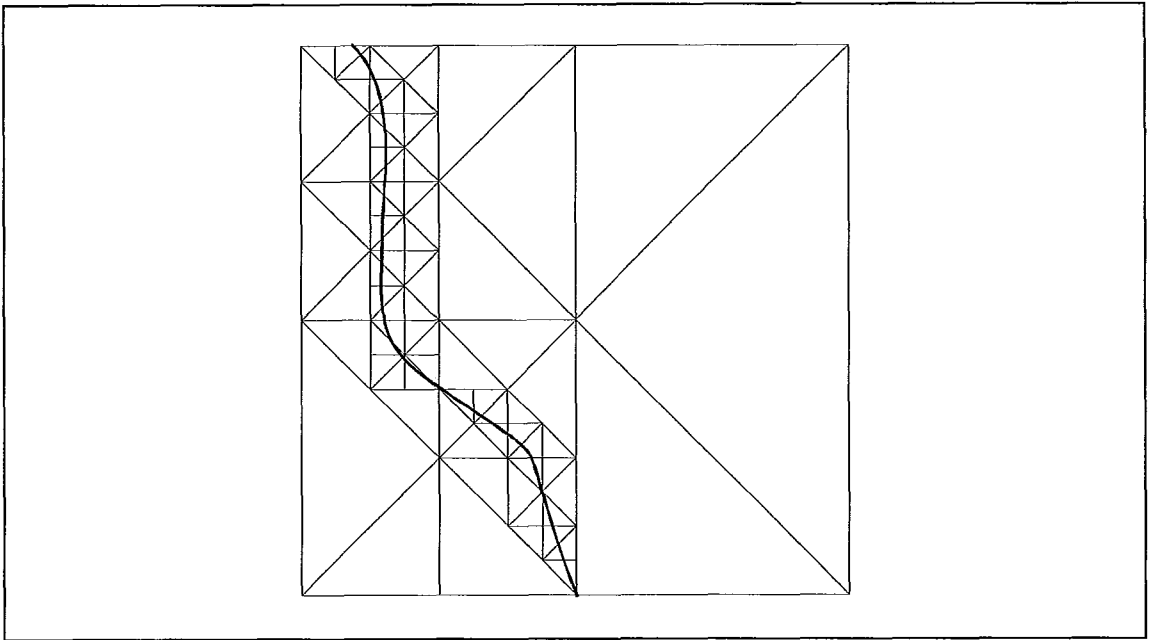


Figura 3.5 Resultado da aplicação do algoritmo seletivo descrito, para a aproximação de uma curva.

### 3.4 Nova Proposta

O objetivo de nossa proposta é estender o grau de adaptatividade do algoritmo analisado de forma a permitir que abortemos o processo de subdivisão não apenas por conta da existência de pontos da superfície no interior de um simplexo. Queremos que tão logo a superfície contida no interior de um simplexo possa ser "bem" aproximada por uma superfície linear, o processo de subdivisão daquele simplexo seja abortado.

Devemos ter em mente que a condição de manter o conjunto de simplexos localizados sobre a variedade como uma triangulação é fundamental na garantia da continuidade da superfície linear por partes que queremos construir.

Para resolver tal problema iremos propor um novo algoritmo para construção da subdivisão do espaço, de forma a garantir que sempre que abortarmos a subdivisão de um simplexo, por conta do critério de precisão para a construção de sua aproximação linear já ter sido atendido, a transição entre esse simplexo e seus vizinhos de níveis diferentes será feita por faces comuns a ambos.

### ***3.5 Conclusões***

Nesse capítulo mostramos que uma determinada classe das triangulações regulares, as CFK, associada com o processo de subdivisão binária, nos fornece meios de promover refinamentos em uma triangulação inicial.

Esse refinamento baseado na subdivisão binária tem a vantagem de manter a forma dos simplexes da triangulação CFK original. Essa sem dúvida é uma propriedade bastante forte no contexto de aplicações que utilizam triangulações em análises numéricas, pois a forma dos simplexes tem impacto direto sobre a instabilidade numérica das soluções.

Analizamos então um algoritmo capaz de construir uma subdivisão seletiva do espaço, de tal sorte que os refinamentos eram direcionados pela existência ou não de parte da superfície definida pelos zeros de uma função polinomial implícita a ser aproximada dentro do simplexo.

No próximo capítulo iremos discutir um novo algoritmo que permita manter o refinamento seletivo do espaço, mas que possa também suportar a adaptatividade quanto à precisão da aproximação linear dentro de uma célula.

## 4. SUBDIVISÃO ESPACIAL ADAPTATIVA

---

### 4.1 Introdução

No capítulo anterior, apresentamos um algoritmo que promove a subdivisão seletiva do domínio de uma superfície, definida a partir do conjunto de zeros de uma função polinomial por partes tri-variada, associada à função característica de um sólido CSG. Destacamos suas principais características e limitações.

Nesse capítulo, daremos continuidade ao nosso trabalho propondo acrescentar àquele algoritmo a adaptatividade com relação a essa superfície que será aproximada por outra, linear por partes. Para isso apresentaremos um novo algoritmo que, mantendo a característica de promover uma subdivisão seletiva do domínio, permite ainda que o processo de subdivisão se adapte à superfície implícita a ser aproximada.

### 4.2 Por Que a Adaptatividade ?

Como vimos no capítulo anterior, a subdivisão seletiva do espaço permite que concentremos nossos esforços nas regiões do domínio que realmente são importantes, ou seja, aquelas que contêm parte da superfície a ser aproximada. Com isso reduzimos o trabalho, localizando-o em regiões bem definidas.

Ainda assim o algoritmo apresentado não é tão eficiente, se levarmos em conta o fato do processo de subdivisão ser uniforme numa vizinhança da superfície. Em outras palavras, a triangulação que cobre a superfície implícita que estamos aproximando possui todos os seus elementos em um mesmo nível de refinamento, e portanto com o mesmo tamanho.

Em geral, as funções que queremos aproximar não se comportam de maneira uniforme em todo o espaço do domínio. É natural que certas regiões sejam “bem comportadas”, ou seja, com uma curvatura baixa, enquanto que em outras sua variação pode ser bastante acentuada.

Levando este fato em consideração, é interessante que a triangulação que envolve a superfície também possa acompanhar essa variação de comportamento através da variação do grau de refinamento de seus elementos. Em regiões cujo comportamento da função se aproxima ao de uma função linear, com um grau de refinamento baixo

podemos garantir que uma boa aproximação será gerada. De forma análoga, as regiões do domínio em que o comportamento da função não é próximo do linear será necessário um refinamento mais rigoroso, de forma que o comportamento da aproximação linear nessa região seja mais próximo da superfície real.

Para alcançar esse grau de adaptatividade o processo de subdivisão deverá ser abortado, não apenas quando simplexos que não interceptam a superfície sejam alcançados, mas também tão logo seja detectado um comportamento da superfície próximo ao de um plano no interior do simplexo.

### ***4.3 Como Garantir a Continuidade ?***

A manutenção da condição descrito acima irá se tornar o principal obstáculo ao tentarmos adaptar o algoritmo apresentado no último capítulo. Ao inserirmos a possibilidade de abortar o processo de subdivisão antes do último nível de refinamento, nos deparamos com o problema de garantir que a aproximação linear resultante seja contínua.

Na figura 3.5 mostramos uma visão da subdivisão seletiva de um domínio. Fica claro que os simplexos localizados no último nível formam uma triangulação. É justamente essa característica que nos permite garantir a continuidade da aproximação gerada a partir desses simplexos.

Os demais simplexos apresentam como elementos de interseção arestas e vértices. Esses últimos tem como característica não serem comuns aos dois simplexos adjacentes pela aresta onde se localizam. Esse tipo de situação irá caracterizar que o conjunto total de simplexos que subdividem o domínio não forma uma triangulação.

No contexto do algoritmo analisado no capítulo anterior, como esse fato não ocorre entre os simplexos que contém parte da superfície ele não é relevante. Porém, ao introduzirmos o novo critério de adaptatividade, tais situações poderão ocorrer também com os simplexos que são cortados pela superfície.

Para assegurar a continuidade da aproximação devemos garantir que, mesmo com tamanhos distintos, o conjunto de simplexos durante todo o processo de subdivisão é uma triangulação. Iremos nos valer das triangulações CFK, mais precisamente das suas variações apresentadas no capítulo anterior, para tornar viável a garantia de tal condição. Como veremos no decorrer desse capítulo, várias de suas propriedades nos serão bastante úteis nesse sentido.

Antes de nos aprofundarmos no algoritmo responsável por essa garantia, vamos introduzir duas definições que nos serão úteis no decorrer dessa seção.

Denomina-se *conjunto de vizinhos de um simplexo*  $\sigma$  em uma triangulação, o conjunto formado por todos os simplexos que compartilham a sua maior aresta. Se todos os simplexos do conjunto de vizinhos de  $\sigma$  tem a aresta comum como maior aresta dizemos que esse conjunto forma um *Núcleo*.

O processo de subdivisão binária se baseia no corte do simplexo  $\sigma$  pela sua maior aresta. Esse corte promove a inserção de um vértice sobre essa aresta, causando uma quebra da condição de triangulação da vizinhança desse vértice/aresta. A idéia por trás do conjunto de vizinhos de  $\sigma$  é justamente localizar em quais simplexos essa inserção gera problemas, para a seguir resolvê-lo.

Portanto precisamos de um algoritmo que, acoplado ao processo de subdivisão binária, garanta que a subdivisão de um simplexo pode ser feita sem que a condição de triangulação do conjunto dos simplexos seja perdida.

## 4.4 Refinamento Básico

O algoritmo que iremos descrever foi proposto por [PERS93] a partir do trabalho de [RIVA92]. Esse algoritmo terá por objetivo não só promover a subdivisão binária dos simplexos, como também garantir a condição de triangulação da subdivisão do domínio. A proposta de Rivara pode ser aplicada a uma triangulação qualquer. No entanto nosso enfoque será na proposta de Persiano, que presume a utilização de uma triangulação baseada nas variações das triangulações CFK, anteriormente apresentadas.

Podemos caracterizar dois casos com os quais iremos nos defrontar durante o processo de subdivisão binária de um simplexo. O primeiro, e mais simples, ocorre quando a subdivisão binária se dá em um simplexo  $\sigma$ , cujo conjunto de vizinhos forma um núcleo. Nesse caso o vértice incluído está localizado sobre a maior aresta de  $\sigma$  e de todos os demais simplexos localizados nesse núcleo. Esse novo vértice no entanto não terá correspondente em nenhum dos simplexos desse núcleo, quebrando portanto a condição que caracteriza a subdivisão como uma triangulação. Para que esse vértice passe a ser comum a todos os simplexos que compartilham aquela aresta, basta que seja aplicada uma subdivisão binária sobre cada um dos simplexos desse núcleo. Dessa forma em cada simplexo do núcleo um novo vértice será introduzido, coincidente com o gerado pela subdivisão binária de  $\sigma$ .

Na figura 4.1, temos um exemplo apresentado a situação descrita acima. Usamos um caso bidimensional para facilitar a representação, porém a idéia pode ser estendida para três dimensões. O simplexo  $\sigma_1$  ao ser subdividido cria o vértice  $V_1$ . Se analisarmos o conjunto de vizinhos de  $\sigma_1$  notamos que o novo vértice inserido não possui correspondente no simplexo  $\sigma_2$ , caracterizando assim que essa subdivisão deixou de ser uma triangulação. O conjunto de vizinhos de  $\sigma_1$  no entanto forma um núcleo, (a aresta

comum a  $\sigma_1$  e  $\sigma_2$  é a maior aresta de ambos). Dessa forma, ao aplicarmos uma subdivisão binária a todos os elementos do núcleo (que nesse caso se resume a  $\sigma_2$ ), voltamos a ter a subdivisão como uma triangulação, já que o novo vértice criado pela subdivisão binária terá um correspondente em cada simplexo do núcleo.

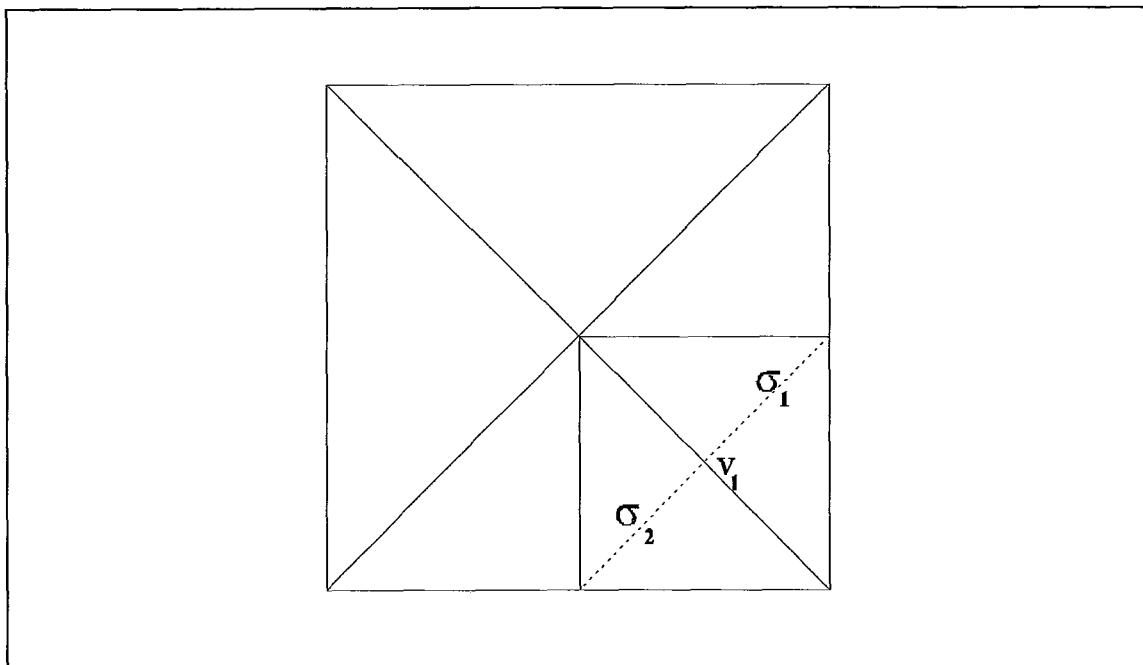


Figura 4.1 Exemplo de subdivisão binária de um simplexo cujo conjunto de vizinhos forma um núcleo.

Caso o conjunto de vizinhos de um simplexo  $\sigma$  não forme um núcleo, o processo para manutenção da condição de triangulação da subdivisão, após a subdivisão binária de  $\sigma$ , será mais complexo. O fato do conjunto de vizinhos de  $\sigma$  não formar um núcleo indica que alguns desses simplexos não tem como maior aresta aquela comum a  $\sigma$ . Portanto a simples aplicação de uma subdivisão binária em todos os simplexos do conjunto de vizinhos de  $\sigma$  não resolve o problema da quebra da condição de triangulação. A subdivisão binária desses simplexos, que não compartilham com  $\sigma$  sua maior aresta, não irá gerar um novo vértice coincidente com aquele, resultado da subdivisão binária de  $\sigma$ . Pelo contrário, irá gerar um novo vértice que afetará um novo conjunto de simplexos.

Nesse caso a subdivisão binária do conjunto de vizinhos de  $\sigma$  irá gerar em todos aqueles simplexos, cuja aresta comum a  $\sigma$  não é a maior, vértices que resultam na quebra da condição de triangulação da subdivisão. Para cada um desses simplexos um novo conjunto de vizinhos fica definido. Novamente, se esse conjunto não formar um núcleo, novos vértices problemáticos serão inseridos. Conforme [RIVA92] prova, esse processo de subdivisão binária e resolução de vértices não comuns é finito, já que a cada novo conjunto de vizinhos com problemas a aresta comum é sempre maior que a do



conjunto de vizinhos anterior. Além disso tão logo um núcleo seja alcançado, o processo se estabiliza sem gerar novos vértices problemáticos.

Na figura 4.2 temos um exemplo da situação acima apresentada. Ao promovermos um refinamento sobre o simplexo  $\sigma_1$  estamos inserindo um novo vértice,  $v_1$ , não comum aos simplexos do seu conjunto de vizinhos ( nesse caso constituído unicamente do simplexo  $\sigma_2$ ). A subsequente subdivisão binária de  $\sigma_2$  não devolve à subdivisão o status de triangulação, pelo contrário, inclui um novo vértice  $v_2$  que quebra a condição de triangulação da subdivisão. Um novo conjunto de vizinhos é então definido a partir de  $\sigma_2$ , que também não forma um núcleo. A subdivisão binária de seus elementos não resolve o problema do vértice  $v_2$ . No entanto, o novo vértice introduzido ao subdividirmos  $\sigma_3$  não altera, localmente, a condição de triangulação para o seu conjunto de vizinhos (que nesse caso é vazio).

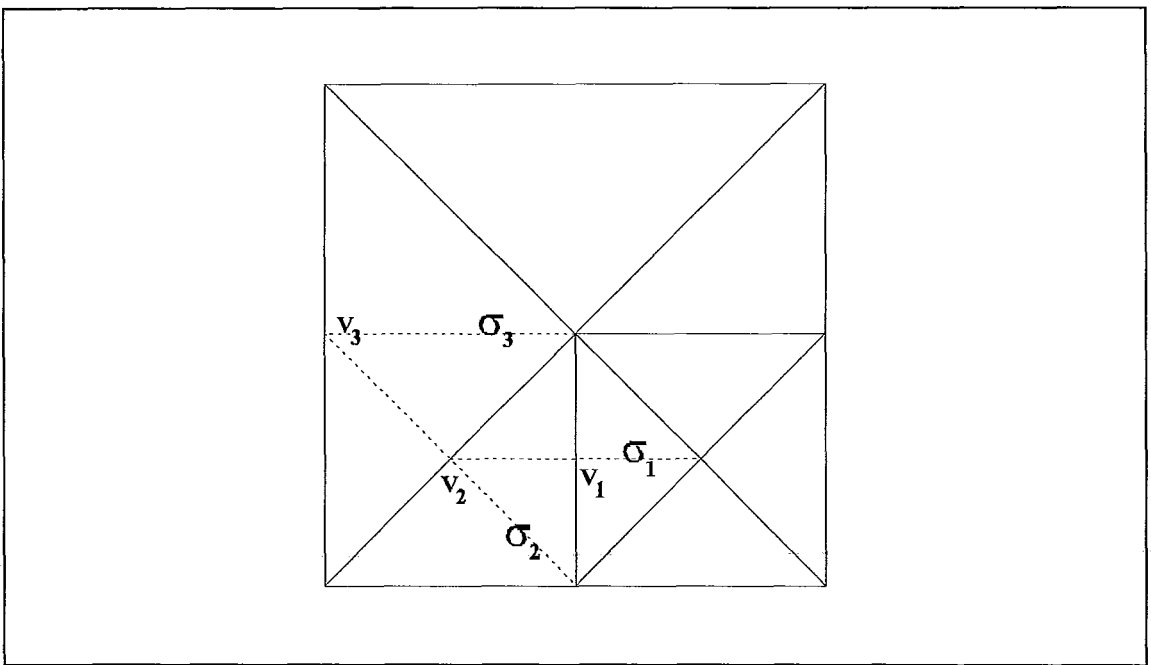


Figura 4.2 Exemplo de subdivisão binária de um simplexo cujo conjunto de vizinhos não forma um núcleo.

Reavaliando o simplexo  $\sigma_2$  podemos observar que um novo conjunto de vizinhos foi definido, já que um dos filhos do simplexo  $\sigma_3$  é agora o vizinho de  $\sigma_2$  por sua maior aresta. Esse novo conjunto de vizinhos forma um núcleo, e portanto com a simples subdivisão de seus elementos resolvemos o problema causado por  $v_2$ . De maneira análoga a reavaliação do conjunto de vizinhos de  $\sigma_1$  agora permite caracterizar um núcleo, e portanto a simples subdivisão binária de seus elementos resolve o problema causado pelo vértice  $v_1$ .

O algoritmo apresentado em [PERS93] propõe que todas as subdivisões sejam feitas como último passo no processo do refinamento. Em outras palavras, primeiro iremos promover a busca por núcleos, onde a subdivisão binária garantidamente não introduz novos vértices que quebram a condição da subdivisão de formar uma triangulação. Ao alcançarmos esses núcleos aplicamos subdivisões binárias em todos os seus elementos. Durante o retorno do processo de busca iremos aplicar as subdivisões binárias sobre todos os conjuntos de vizinhos pelos quais passamos e que agora tornam-se também núcleos.

Podemos fazer tal afirmação com base em duas propriedades das triangulações adaptativas construídas a partir da extensão das triangulações CFK [BARB94]. Dado um tetraedro  $\sigma$  e sua maior aresta, podemos identificar duas de suas faces que compartilham aquela aresta. Associados a essas faces temos dois outros tetraedros, vizinhos de  $\sigma$  por uma delas, onde pelo menos um é de mesmo tipo de  $\sigma$  e o outro de um tipo possivelmente um nível *menos refinado*. Os demais elementos que compartilham a maior aresta de  $\sigma$ , mas que não são adjacentes por face, podem ser menos refinados, e nesse caso diferirem de  $\sigma$  em no máximo dois níveis de refinamento.

A partir dessas propriedades podemos concluir também que o processo de busca por um núcleo é finito, já que a cada vez que um elemento quebra a condição de núcleo de um conjunto de vizinhos ele é maior que o simplexo base do conjunto.

A seguir apresentamos o algoritmo que executa a manutenção da triangulação após uma subdivisão binária.

*Refinamento Básico ( Simplexo, Triangulação )*

*Para cada simplexo  $\sigma$  do conjunto de vizinhos do Simplexo faça*

*Se (  $\text{tipo}(\sigma) \neq \text{tipo}(\text{Simplexo})$  ) então*

*Refinamento Básico ( $\sigma$ , Triangulação)*

*Subdivide Núcleo (Simplexo)*

Na figura 4.3 podemos acompanhar a sequência em que as subdivisões binárias são processadas, através da numeração das novas arestas criadas (pontilhadas) durante o processo de subdivisão do simplexo  $\sigma$ , apresentado na figura 4.2 .

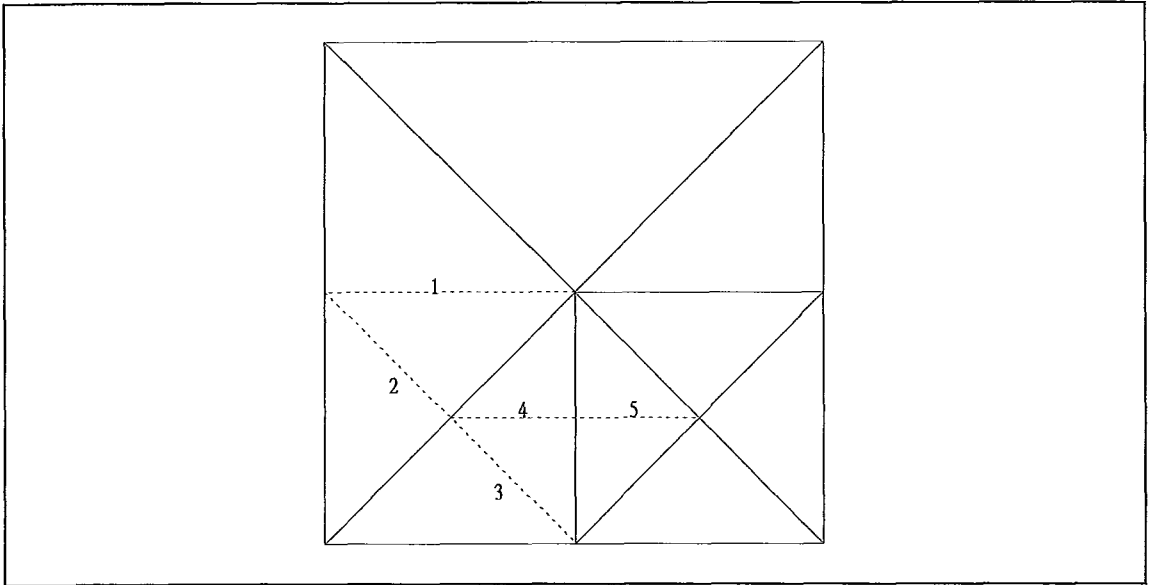


Figura 4.3 Seqüência de subdivisões binárias dentro do processo de refinamento básico.

Uma vantagem desse algoritmo, sobre o originalmente proposto por Rivara, é que em nenhum momento do processo a subdivisão do domínio perde sua condição de triangulação. Todas as subdivisões são processadas no caminho de volta da recursão.

Um fato importante a ser ressaltado nessa etapa é que, ao contrário do algoritmo proposto no capítulo anterior, a utilização do refinamento básico como ferramenta no processo de subdivisão de um simplexo, induz à necessidade de se ter conhecimento das relações de adjacência entre os simplexos. Em outras palavras, a estrutura da triangulação deve estar explicitamente representada. A proposta vista no capítulo anterior podia prescindir dessa representação já que havia a garantia de que todos os simplexos do último nível da subdivisão eram do mesmo tipo, e portanto formavam uma triangulação.

## 4.5 Manutenção da Estrutura da Triangulação

Cabe nesse momento ressaltar alguns detalhes e propriedades das triangulações, construídas através desse algoritmo. Elas servirão de base na manutenção da estrutura de adjacências que deverá ser armazenada para a representação da triangulação.

Em primeiro lugar, precisaremos identificar a maior aresta de um tetraedro. Isso é essencial dentro do algoritmo de subdivisão binária. Utilizaremos um sistema de rotulação dos vértices dos simplexos de forma que a maior aresta seja formada sempre por um par determinado de vértices :  $V_0$  e  $V_3$ .

Esse sistema de rotulação deve garantir que qualquer tetraedro irá manter as rotulações de vértices consistentes com o tamanho das arestas por eles definida. Portanto durante a subdivisão binária a rotulação dos vértices dos tetraedros filhos deve ser feita de maneira consistente.

Desse modo, construímos uma relação capaz de identificar, em qualquer tetraedro, independente de seu tipo, suas arestas pelo seu tamanho. Na figura 4.4 apresentamos a seqüência de subdivisões ilustrando como os vértices são rotulados segundo a proposta acima. Na tabela 4-1 associamos os tamanhos relativos das arestas para cada tipo de tetraedro, bem como a identificação da aresta através da sua própria rotulação e dos vértices que definem.

Com essa rotulação dos vértices podemos também amarrar as relações de adjacência de um tetraedro qualquer. E mais do que isso, construir uma combinatória consistente para o tratamento das vizinhanças dos tetraedros resultantes de subdivisões.

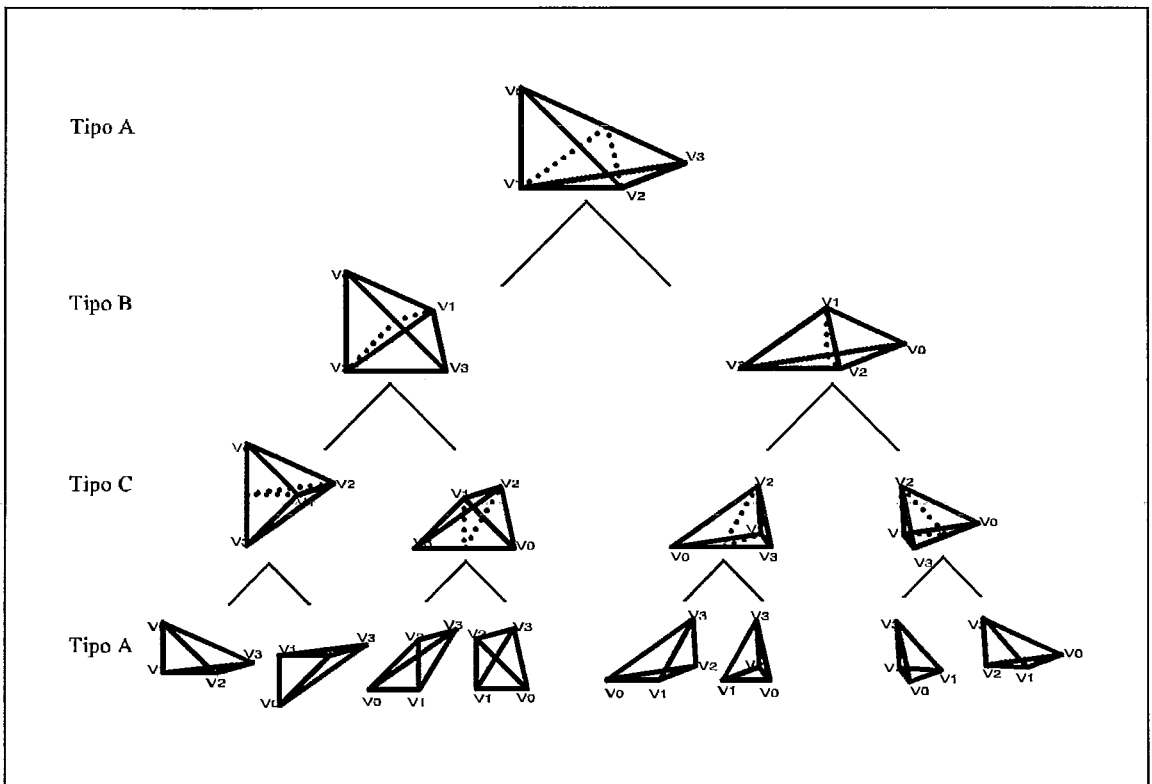


Figura 4.4 Seqüência de subdivisões binárias partindo de um simplexo do tipo A. Na figura as rotulações dos vértices podem ser acompanhadas.

Definiremos, a partir de um tetraedro  $\sigma$ , o vizinho  $i$  de  $\sigma$  como sendo o tetraedro adjacente pela face oposta ao vértice  $i$  de  $\sigma$ .

Quando aplicamos sobre um tetraedro uma subdivisão binária, apenas duas de suas faces (aquelas que definem a maior aresta) são cortadas. As duas restantes são repassadas de forma integral para cada um dos dois tetraedros filhos. Isso pode ser visto na figura 4.4.

Tipo do Simplexo	Rotulação das Arestas					
	0 $V_0-V_3$	1 $V_1-V_3$	2 $V_0-V_2$	3 $V_0-V_1$	4 $V_2-V_3$	5 $V_1-V_2$
A	$\sqrt{3}$	$\sqrt{2}$	$\sqrt{2}$	1	1	1
B	$\sqrt{2}$	1	1	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{3}}{2}$
C	1	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{1}{2}$

Tabela 4-1 Dimensões de cada aresta para cada um dos três tipos de tetraedros.

Podemos dessa forma montar uma nova relação que nos permite identificar, a partir das relações de adjacência do tetraedro pai, quais serão os vizinhos dos tetraedros resultantes de sua subdivisão. Isso é mostrado na tabela 4-2. Por convenção na figura 4.4 os  $filho_1$  e  $filho_2$  resultantes da subdivisão binária de um tetraedro são respectivamente os tetraedros do ramo à direita e à esquerda da árvore de subdivisão.

Esse fato é de grande importância dentro de nossa proposta. Se apenas uma das faces do simplexo pai é repassada para um de seus filhos, isso significa que entre dois tetraedros de nível subsequente só pode haver uma face comum (a face herdada do pai). Todas as demais faces terão dimensões não compatíveis com os tetraedros do nível do simplexo pai. Isso portanto justifica as propriedades apresentadas na seção anterior.

Rótulo vizinho do filho	Pai Tipo A		Pai Tipo B ou C (filho1)		Pai Tipo C (filho2)	
	FILHO <sub>1</sub>	FILHO <sub>2</sub>	FILHO <sub>1</sub>	FILHO <sub>2</sub>	FILHO <sub>1</sub>	FILHO <sub>2</sub>
0	Filho <sub>1</sub>	Filho <sub>2</sub>	Filho <sub>1</sub>	Filho <sub>2</sub>	Filho <sub>1</sub>	Filho <sub>2</sub>
1	T <sub>3</sub>	T <sub>0</sub>	T <sub>3</sub>	T <sub>0</sub>	T <sub>3</sub>	T <sub>0</sub>
2	Filho <sub>1</sub> (T <sub>1</sub> )	Filho <sub>2</sub> (T <sub>2</sub> )	Filho <sub>1</sub> (T <sub>1</sub> )	Filho <sub>2</sub> (T <sub>1</sub> )	Filho <sub>1</sub> (T <sub>1</sub> )	Filho <sub>2</sub> (T <sub>1</sub> )
3	Filho <sub>1</sub> (T <sub>2</sub> )	Filho <sub>2</sub> (T <sub>1</sub> )	Filho <sub>1</sub> (T <sub>2</sub> )	Filho <sub>2</sub> (T <sub>2</sub> )	Filho <sub>1</sub> (T <sub>2</sub> )	Filho <sub>2</sub> (T <sub>2</sub> )

Tabela 4-2 Relações de adjacência para os tetraedros resultantes da subdivisão binária a partir das relações de seu tetraedro pai.

Essa propriedade atua portanto como uma restrição para as possíveis relações de adjacência entre simplexos. Essas restrições simplificam sensivelmente a combinatória dos vizinhos de um tetraedro qualquer. Analisando com cuidado podemos construir

todos os casos viáveis dentro de um conjunto de vizinhos, cujo tetraedro base é de um dos 3 tipos. São exatamente 4 casos com tetraedros base de tipo A ou C e 2 no caso de simplexes base de tipo B [BARB94].

## 4.6 Subdivisão Adaptativa

A partir do algoritmo de refinamento básico podemos propor um novo algoritmo para construir a triangulação do domínio da função de forma seletiva, eliminando elementos que não interceptam a superfície definida pelas suas raízes, e adaptativa, refinando apenas elementos em que a aproximação linear seria grosseira para uma dada precisão.

Nosso algoritmo irá construir uma triangulação inicial, tal como proposto no capítulo anterior. Com base nos 6 tetraedros dessa triangulação, montaremos uma lista que conterá simplexes não avaliados ou, durante a evolução do algoritmo, que ainda não satisfazem nosso critério de precisão. Esses 6 simplexes iniciais serão colocados em uma lista contendo simplexes ainda não avaliados : a *ListaAAvaliar*.

Durante o processo de construção da triangulação teremos que considerar dois critérios básicos para avaliação dos simplexes. O primeiro diz respeito à parte seletiva do processo, que visa determinar se o simplexo contém ou não parte dos zeros da função a ser aproximada em seu interior. O segundo está relacionado a adaptação propriamente dita, avaliando se a aproximação linear à superfície definida pelas raízes da função já está suficientemente próxima da solução real, considerando-se uma dada tolerância.

Enquanto a *ListaAAvaliar* contiver simplexes, retiramos um deles e o avaliamos segundo a precisão estabelecida. Caso ele satisfaça o critério, será colocado em uma outra lista, a de simplexes aprovados. Essa lista ao final do processo irá conter os simplexes a partir dos quais iremos gerar a aproximação linear. Caso seja recusado pelo critério de precisão, um refinamento básico é aplicado gerando, além de seus dois filhos, uma família de novos simplexes por propagação do refinamento básico. Esses são então avaliados contra o critério seletivo e, caso aprovados, incluídos na *ListaAAvaliar*.

A seguir apresentamos, em pseudocódigo, o algoritmo descrito acima.

*TriangulaçãoAdaptativa* (*Domínio, Função, NívelMax*);

*ListaAAvaliar* = *ConstroiTriangulaçãoInicial*(*Domínio*);  
 enquanto (*ListaAAvaliar*  $\neq \emptyset$ ) faça

*RetiraSimplexo*( $\sigma$ , *ListaAAvaliar*);

se (*AtendeCritério*( $\sigma$ , *Função*) ) então

*AprovaSimplexo*(*ListaDeAprovados*,  $\sigma$ );

senão

*Subdivide*( $\sigma$ );

para cada tetraedro  $\sigma_i$  em *ListaDeAprovados* faça

*GeraAproximaçãoLinear*( $\sigma_i$ , *Função*);

O algoritmo inicia montando a *ListaAAvaliar* com os 6 simplexos, resultado da subdivisão do domínio (cubo) onde a função está definida. Um ciclo iterativo é então iniciado de modo que cada simplexo dessa lista é removido e avaliado até que não haja mais simplexos dependentes de avaliação.

A função *AtendeCritério* é a responsável pela avaliação do critério de precisão. Nela um simplexo será avaliado quanto à qualidade da aproximação linear dos zeros da função em seu interior, dada uma certa tolerância. Caso o simplexo analisado seja aprovado o procedimento *AprovaSimplexo* o coloca em *ListaDeAprovados*.

Caso contrário o procedimento *Subdivide* irá aplicar o refinamento básico sobre esse simplexo gerando dois filhos. Outros simplexos podem ser subdivididos em razão da propagação de subdivisões binárias que podem ocorrer quando um refinamento básico é aplicado. O conjunto de simplexos subdivididos é removido da lista. Nesse ponto aplicamos o critério seletivo para determinar se algum dos filhos desses simplexos subdivididos possui parte do conjunto de zeros da função em seu interior. Aqueles ditos *interseptantes* serão colocados novamente na *ListaAAvaliar* para que possam ser novamente avaliados quanto à componente adaptativa do critério.

Ao final do processo de refinamento, quando não houver mais tetraedros na *ListaAAvaliar*, todos os simplexos aprovados pelo critério estarão na *ListaDeAprovados*. Percorremos, então, essa lista construindo a aproximação linear da superfície em cada simplexo.

Vale enfatizar um detalhe importante no que diz respeito a geração da aproximação. Estas não são geradas tão logo um simplexo é aprovado pelo critério de precisão. Nesse novo algoritmo nada impede que um simplexo aprovado, e portanto colocado na *ListaDeAprovados*, seja subdividido novamente. Visto que o processo de refinamento básico pode gerar uma propagação de subdivisões binárias, que

possivelmente alcançam simplexos já aprovados, temos que aguardar até que todos os simplexos a avaliar sejam processados para então construir a aproximação linear. Caso contrário corremos o risco de construir aproximações que não são contínuas.

Cabe também ressaltar que podemos assim proceder pois nosso problema é *espacialmente decomponível*. Essa característica nos permite afirmar que quanto mais refinado nosso domínio, melhor será a solução encontrada. Desse modo, o fato de refinarmos mais um tetraedro já aprovado não irá de forma alguma piorar a sua aproximação. A essa propriedade denomina-se *invariância a inclusão* [PERS93].

Uma outra característica da solução final encontrada por esse algoritmo é que ela é *mínima*, no sentido de construir a menor triangulação de sua classe que satisfaz às condições de precisão estabelecidas [PERS93].

## 4.7 Conclusão

Apresentamos nesse capítulo os principais obstáculos na extrapolação da idéia primária de adaptatividade proposta no capítulo anterior.

Para suplantá-los definimos uma nova maneira de subdividir o espaço, agora baseada no fato de que a camada de simplexos que envolve o conjunto de zeros da função a ser aproximada, não mais será uniforme como no algoritmo anterior. Isso porque estamos interessados em estender o conceito de adaptatividade de modo que a subdivisão possa acompanhar também a variação de curvatura da superfície implícita a ser aproximada.

Tal fato nos levou também à necessidade de representar a estrutura da triangulação do espaço durante todo o processo. Discutimos também outras propriedades referentes as triangulações, definidas adaptativas e baseadas em extensões das triangulações CFK, que reiteram a sua utilidade dentro do novo algoritmo.

No próximo capítulo nos aprofundaremos mais em alguns dos detalhes relativos aos critérios que guiam o algoritmo proposto. Por exemplo estaremos interessados em responder perguntas como : De que maneira podemos aferir se um simplexo possui parte da superfície em seu interior ? Ou ainda como definir se uma aproximação já é próxima o suficiente da superfície, dada uma certa precisão ?



# 5. CRITÉRIO DE INTERFERÊNCIA

---

## 5.1 Introdução

No capítulo anterior, apresentamos um algoritmo capaz de construir, a partir de uma triangulação CFK inicial do domínio de uma função implícita, uma outra triangulação (adaptativa), refinamento da original, guiada pela busca do conjunto das suas raízes.

A garantia de que durante todo o processo de refinamento temos nossa subdivisão do domínio formando uma triangulação é assegurada pelo procedimento definido como *refinamento básico*.

A busca pelas raízes da função é guiada por dois critérios : o seletivo, que determina se um dado simplexo intercepta ou não a superfície definida pelo conjunto, e o de precisão, que é responsável pela avaliação da qualidade da aproximação linear da superfície dentro do simplexo.

Nosso objetivo nesse capítulo será descrever, de forma mais cuidadosa, o funcionamento do primeiro critério, que dentro do escopo desse capítulo denominaremos de *critério de interferência*.

Nesse ponto, retomaremos o escopo original de nosso trabalho, quando procurávamos uma forma de construir uma aproximação linear para a fronteira de um sólido representado no esquema CSG. Estaremos preocupados portanto em definir como um simplexo pode ser classificado contra a árvore CSG que representa o sólido. Mais ainda, como podemos estabelecer um critério seguro para determinar se uma dada primitiva é interceptada pelo simplexo, e, de posse dessa informação, utilizá-la na composição de uma classificação geral para toda a árvore CSG.

Iniciaremos discutindo questões relativas à representação das primitivas, especificando que representação é mais adequada para a avaliação do critério de interferência. Em seguida, abordaremos uma solução para compor as classificações das diversas primitivas segundo a árvore CSG do sólido.

## 5.2 Representação das Primitivas

O algoritmo proposto no capítulo anterior tem como característica a possibilidade de aplicação a um conjunto bastante amplo de problemas. A única condição é que possamos descrevê-los através de uma função implícita, em um domínio limitado. A partir daí, somos capazes de construir uma triangulação adaptativa, base para o cálculo de uma solução linear por partes, aproximadora da solução original, e que atende a um certo grau de precisão preestabelecido.

No entanto, nossa motivação original é mais específica. Utilizaremos tal algoritmo para a construção de uma poligonização da superfície definida implicitamente pela fronteira de um sólido, representado no esquema CSG.

Esse problema se apresenta como um caso particular dentro do contexto até agora abordado, já que a superfície de um sólido CSG pode ser definida implicitamente por uma função, composição das funções características de cada primitiva, combinadas através dos operadores *min/max*. O que caracteriza uma restrição ao escopo anterior, é o fato das primitivas utilizadas na construção dos sólidos serem, em geral, definidas por funções polinomiais.

Em geral, a representação das superfícies polinomiais que definem as primitivas é feita na base algébrica. A utilização dessa base pode, no nosso caso, trazer algumas dificuldades, em particular no que diz respeito ao tratamento adequado das informações necessárias aos critérios que guiam nosso algoritmo.

A primeira é como determinar se uma primitiva intercepta ou não um simplexo. Como já vimos, esse problema se resume a uma busca pelo conjunto das raízes da função polinomial que descreve a primitiva dentro do simplexo. Essa tarefa pode se tornar computacionalmente cara para polinômios de graus maiores que 2 ou 3.

Outro detalhe importante é garantir que teremos meios de conhecer o comportamento dessa função no interior de um simplexo. Nesse sentido a análise dos coeficientes algébricos da função característica associada à primitiva não nos fornece informação adequada para que uma avaliação simples possa ser feita.

Uma base alternativa, amplamente utilizada na área de modelagem de superfícies, é a base de Bernstein. Para polinômios univariados, de grau  $n$ , a base de Bernstein é definida por polinômios da forma :

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}$$

São várias as propriedades dessa base que a tornam interessante para a utilização na área de modelagem de superfícies [FARI86] [FARI90]. Destacaremos aqui aquelas que serão úteis para nossos propósitos.

⇒ Os polinômios de Bernstein satisfazem a relação recursiva :

$$B_i^n(t) = (1-t)B_i^{n-1}(t) + t B_{i-1}^{n-1}(t)$$

Essa propriedade é particularmente útil para algoritmos de construção de curvas ou superfícies representadas nessa base.

⇒ Os polinômios de Bernstein são estritamente não-negativos no intervalo  $[0,1]$ .

⇒ O somatório dos  $n$  polinômios que formam a base de Bernstein para polinômios de grau  $n$  é igual a 1. Dizemos, portanto que os polinômios de Bernstein formam uma *partição da unidade* :

$$\sum_{i=0}^n B_i^n(t) \equiv 1$$

⇒ Nos extremos do intervalo  $[0,1]$  apenas um dos polinômios de Bernstein tem valor não nulo.

$$B_0^n(0) = 1$$

$$B_n^n(1) = 1$$

⇒ Cada polinômio de Bernstein,  $B_i^n$  de grau  $n$  possui um único máximo, associado ao valor  $t = i/n$ .

⇒ Os polinômios de Bernstein possuem simetria, com respeito aos valores  $t$  e  $1-t$ .

$$B_i^n(t) = B_{n-i}^n(1-t)$$

Na figura 5.1, apresentamos um gráfico que descreve o comportamento dos polinômios de Bernstein univariados de grau 3 dentro do intervalo  $[0,1]$ . Nesse exemplo podemos constatar várias das propriedades acima citadas.

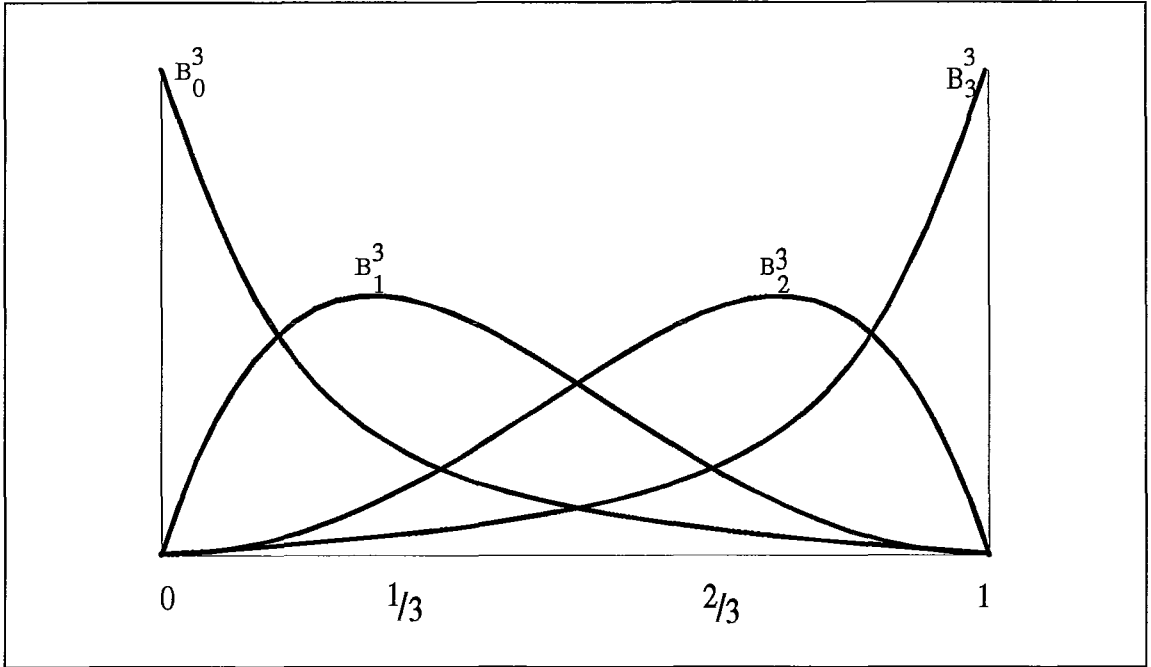


Figura 5.1 Polinômios de Bernstein uni-variados de grau 3.

Podemos notar, por exemplo, que os pontos  $1/3$  e  $2/3$  correspondem ao máximo dos polinômios  $B_1^3$  e  $B_2^3$  respectivamente. Também é simples percebermos que nos extremos do intervalo  $[0,1]$  apenas os polinômios  $B_0^3$  e  $B_3^3$ , respectivamente, são não nulos. E nesse caso apresentam valor máximo 1.

Nenhum dos polinômios apresenta valor negativo em todo o intervalo  $[0,1]$ . Além disso podemos notar a relação de simetria entre os polinômios  $B_0^3$  e  $B_3^3$  e também entre  $B_1^3$  e  $B_2^3$ .

No nosso caso particular, estaremos interessados em representar funções polinomiais de grau  $n$  tri-variadas, que irão representar as primitivas associadas à representação CSG do sólido. Nesse caso a representação dos polinômios de Bernstein é feita na forma :

$$B_{i,j,k,l}^n(u,v,w,t) = \frac{n!}{i!j!k!l!} u^i v^j w^k t^l \quad \text{onde} \quad i+j+k+l=n \quad u+v+w+t=1$$

A partir dessa definição dos polinômios que compõem nossa nova base, todo polinômio deste espaço pode ser expresso na forma :

$$P(u,v,w,t) = \sum_{i+j+k+l=n} C_{ijkl} B_{ijkl}^n(u,v,w,t)$$

Os termos  $C_{ijkl}$  são ditos *coeficientes de Bézier* do polinômio  $P$ .

A quádrupla  $(u,v,w,t)$  representa as coordenadas baricêntricas de um ponto qualquer do domínio de  $P$  ( $E^3$ ), definidas a partir de um simplexo.

Com base na propriedade dos simplexos de formarem uma base geradora do espaço (os vetores que os definem são linearmente independentes) qualquer ponto  $p$  desse espaço pode ser escrito com uma combinação afim dos vértices desse simplexo. Ou seja :

$$p = \sum_{i=0}^n \lambda_i v_i \quad \text{onde} \quad \sum_{i=0}^n \lambda_i = 1$$

Os termos  $v_i$  são associados aos vértices do simplexo (em coordenadas cartesianas) enquanto que a  $(n+1)$ -tupla  $(\lambda_0, \lambda_1, \lambda_2, \dots, \lambda_n)$  representa as *coordenadas baricêntricas* do ponto  $p$  com relação ao simplexo.

Essa representação se mostra bastante interessante pelas suas propriedades. Qualquer ponto localizado no interior do simplexo terá coordenadas baricêntricas estritamente não-negativas. Pontos exteriores ao simplexo terão algumas de suas coordenadas baricêntricas negativas, e nos vértices do simplexo, apenas uma coordenada será não nula e igual a 1.

A partir desse fato, e aliado às propriedades dos polinômios de Bernstein, podemos constatar que os coeficientes associados aos vértices do simplexo possuirão valor idêntico ao do polinômio  $P$  avaliado nesse ponto.

Para cada máximo local de um dos polinômios da base de Bernstein, definimos um coeficiente de Bézier dentro de um simplexo. A figura 5.2 mostra a disposição desses coeficientes nos casos de polinômios uni, bi e tri-variados de grau 3. Eles são importantes pois representam a *influência máxima* daquele polinômio da base dentro do simplexo. No entanto, esses coeficientes não interpolam a superfície, como os coeficientes nos vértices.

Essa relação dos coeficientes de Bézier com o comportamento do polinômio por eles definido será de vital importância na definição do critério de interferência que apresentaremos mais adiante.

Cabe nesse momento ressaltar que a maioria dos modeladores trabalham durante o processo de criação de modelos com as primitivas definidas na base algébrica. Portanto, para que nosso critério possa ser aplicado a esses modeladores, uma conversão base algébrica  $\rightarrow$  base Bernstein se faz necessária. Essa conversão só é necessária uma única vez quando da construção da triangulação inicial.

Essa conversão pode ser feita utilizando a *função de Blossom*. Maiores detalhes sobre o processo podem ser encontrados em [SEID89].

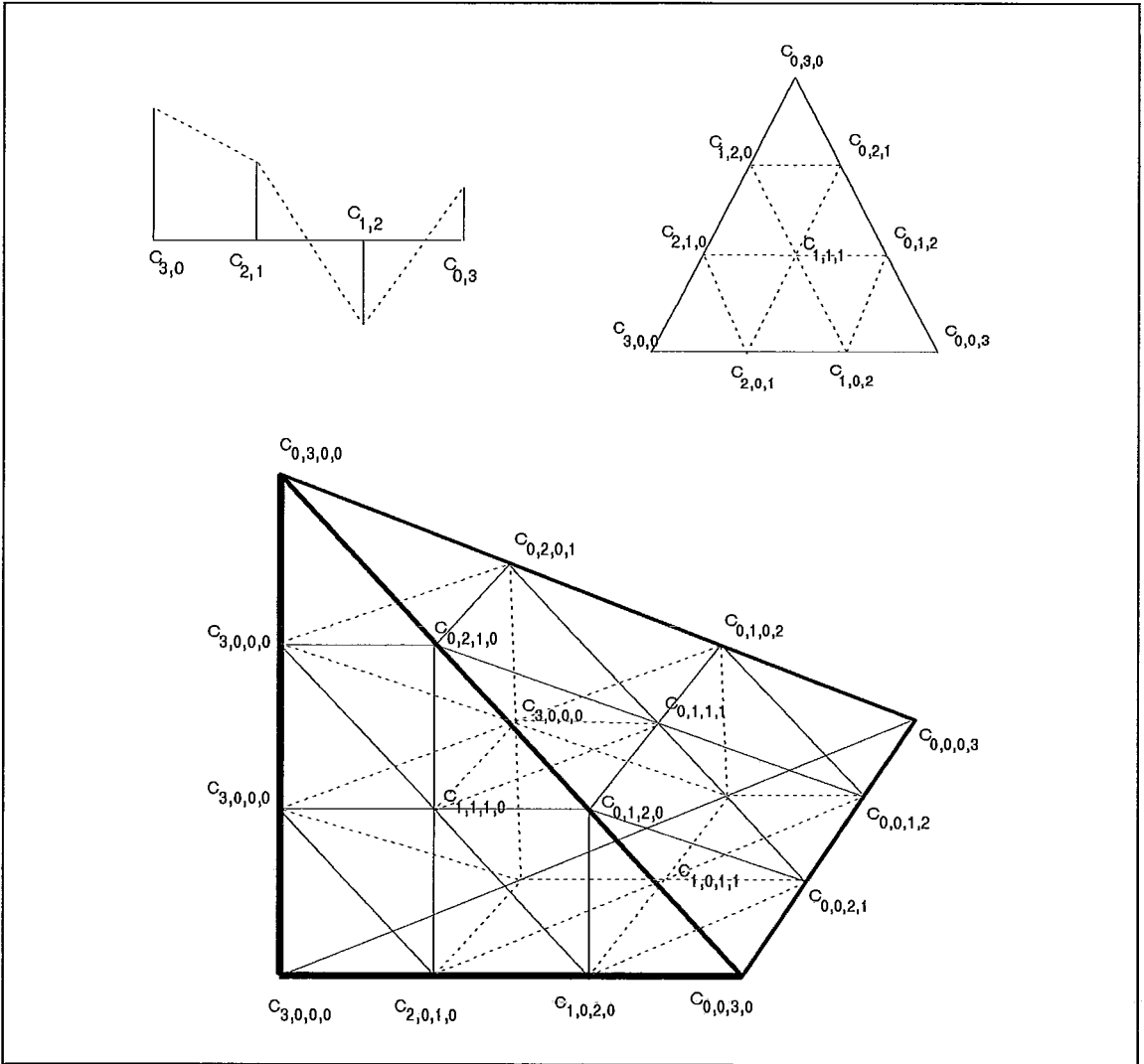


Figura 5.2 Distribuição dos coeficientes de Bézier para polinômios uni, bi e tri-variados.

### 5.3 Subdivisão Binária dos Coeficientes Bézier de uma primitiva

A conversão das bases algébrica  $\rightarrow$  Bernstein além de computacionalmente custosa é também numericamente instável. Logo, o ideal é que apliquemos essa conversão o mínimo possível, ou seja, apenas uma vez, para cada primitiva, em cada um dos 6 simplexes iniciais.

No entanto, durante o processo de refinamento adaptativo da triangulação, temos a necessidade de gerar novos coeficientes de Bézier a cada subdivisão binária aplicada em um simplexo. Várias conversões das representações algébrica  $\rightarrow$  Bernstein, pelos fatos já mencionados, tornaria o algoritmo inviável.

Felizmente, a representação das primitivas na base Bernstein nos fornece uma técnica simples e numericamente estável para obtenção dos novos coeficientes dos simplexos, resultantes de uma subdivisão binária. Essa técnica nada mais é que uma extensão do método de de Casteljau para avaliação de um ponto em uma curva de Bézier.

O método de de Casteljau baseia-se na relação recursiva apresentada na seção anterior, que permite descrever um ponto qualquer  $t$  ( $\in [0,1]$ ) do espaço de parâmetros, de uma curva de grau  $n$ , através da combinação afim dos polinômios de grau  $n-1$  no mesmo ponto  $t$ . Aplicando-se essa relação recursivamente, obtemos um polinômio  $B_i^0(t)$  cujo valor corresponde ao ponto sobre a curva.

A partir do método descrito, podemos extrair uma propriedade bastante interessante e particularmente útil para nosso objetivo : a aplicação desse método divide a curva em dois ramos, um relativo ao intervalo  $[0, 1-t]$  do espaço de parâmetros e outro ao intervalo  $[1-t, 1]$ . Tomando-se de maneira adequada alguns dos coeficientes gerados durante o processo de avaliação da curva no ponto  $t$ , podemos construir dois conjuntos de valores que representam os coeficientes desses dois ramos.

Na figura 5.3 mostramos um exemplo da aplicação do algoritmo de de Casteljau para a avaliação de uma curva de grau  $n=3$ , definida pelos coeficientes  $C_0, C_1, C_2$  e  $C_3$ . O conjunto  $B_0^0, B_1^0, B_2^0$ , e  $B_3^0$  representa (e define) o ramo associado a metade esquerda da curva enquanto que o conjunto  $B_3^0, B_2^1, B_1^2$  e  $B_0^3$ , representa o ramo da direita da curva dividida pelo ponto  $t$ .

Podemos assim gerar toda a malha de coeficientes dos simplexos resultantes de uma subdivisão binária através da aplicação do método de Casteljau (estendido para o espaço tridimensional) sobre o ponto médio da maior aresta. Duas malhas de coeficientes serão geradas, equivalentes aos coeficientes de cada um dos simplexos filhos.

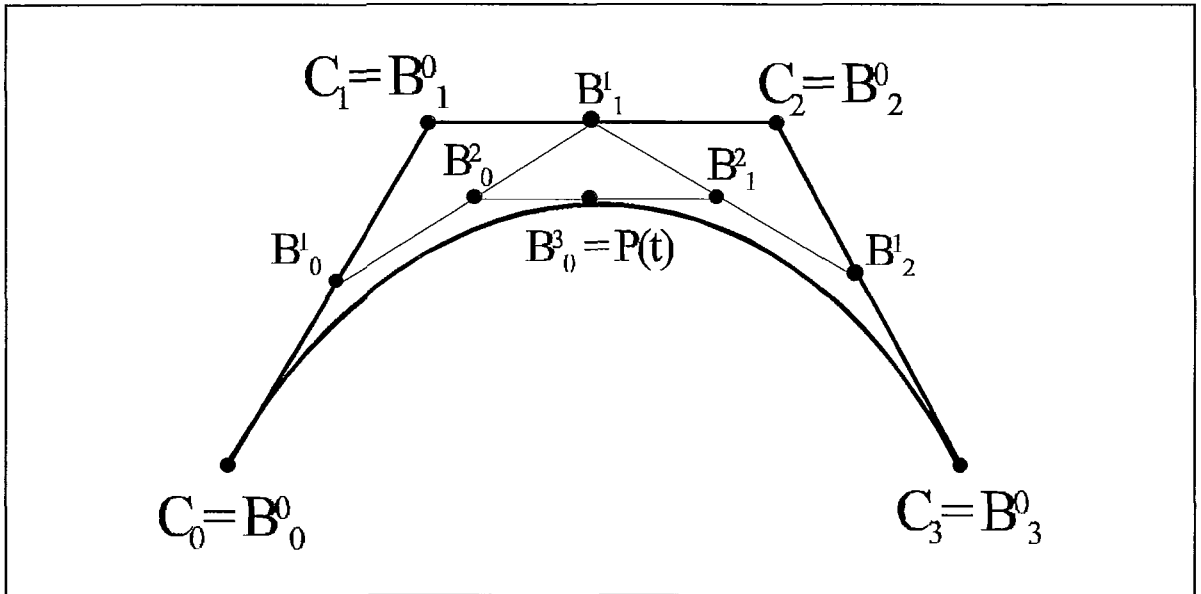


Figura 5.3 Esquema da aplicação do algoritmo de de Casteljau para avaliação de uma cúbica em  $t=0.5$ .

### 5.4 Construção do Critério de Interferência

Uma das propriedades da representação do polinômio, associado a uma primitiva na Base de Bernstein, diz respeito a possibilidade de se obter, através de seus coeficientes, uma avaliação da existência ou não de parte do conjunto de raízes da função a ser aproximada.

Os coeficientes de Bézier de um polinômio  $P$  são considerados como valores limitantes da sua variação dentro do simplexo  $S$ . Ou seja :

$$\text{Min}(C_{ijkl}) \leq \text{Min}(P(x)) \leq \text{Max}(P(x)) \leq \text{Max}(C_{ijkl}) \quad \forall x \in S$$

Esse fato nos permite determinar em que circunstâncias garantidamente não há raízes do polinômio dentro do simplexo. Em outras palavras, baseados nos valores associados ao maior e menor dos coeficientes de Bézier de um polinômio, podemos garantir que uma dada primitiva não intercepta o simplexo.

Essa situação se caracteriza quando o valor máximo e mínimo dos coeficientes de Bézier do polinômio são ambos positivos ou ambos negativos. Vale lembrar que o polinômio descrito na forma Bézier, em nosso contexto, corresponde à representação da função característica da primitiva, que por sua vez define uma partição do espaço em



três regiões distintas : uma positiva (exterior da primitiva), uma negativa (interior da primitiva) e finalmente uma região nula (superfície da primitiva).

A partir dessa interpretação podemos garantir que se ambos os coeficientes máximo e mínimo da representação do polinômio (função característica da primitiva) na base de Bernstein são positivos, o simplexo se encontra na região positiva do domínio. Essa região pode ser interpretada como sendo o exterior da primitiva. De modo análogo quando ambos os coeficientes máximo e mínimo são negativos podemos afirmar que o simplexo se encontra na região interior à primitiva. Na figura 5.4 temos essas duas situações ilustradas nos esquemas a) e b).

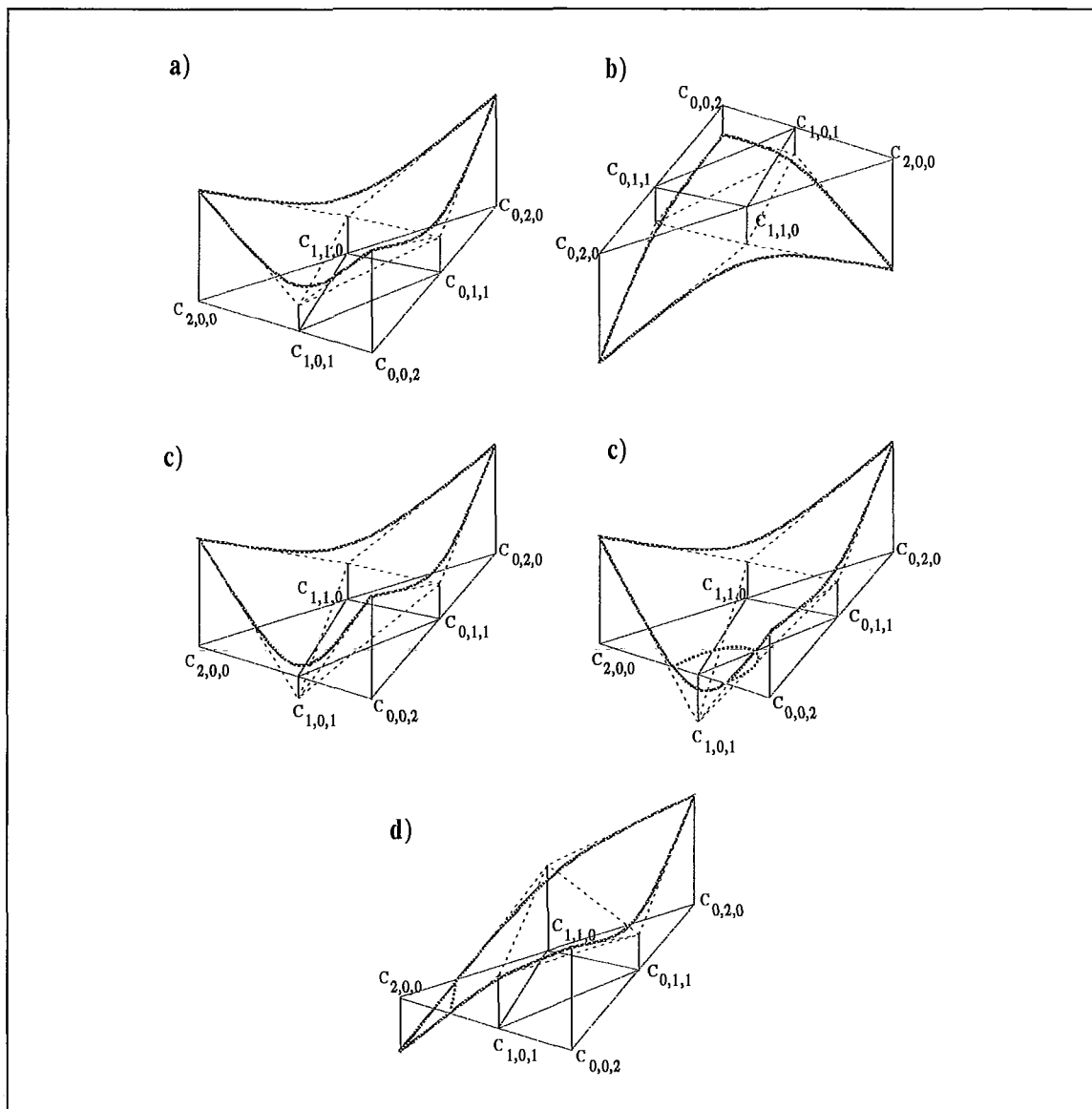


Figura 5.4 Relação entre os valores dos coeficientes de Bézier de um polinômio de grau 2, bi-variada, e seu comportamento dentro do simplexo.

Nada podemos afirmar no entanto quando os sinais dos coeficientes extremos são distintos. Nesses casos pode ou não haver interseção da superfície definida pelos zeros do polinômio com o simplexo (figura 5.4 esquemas *c*)).

Em um caso particular podemos afirmar com segurança que a função intercepta o simplexo : quando os coeficientes dos vértices do simplexo possuem sinais distintos. Nesse caso, garantidamente, parte do conjunto de raízes da função se encontra no interior do simplexo, já que os coeficientes nos vértices interpolam o polinômio (figura 5.4 esquema *d*)).

Podemos portanto classificar um simplexo, quanto à sua posição relativa a uma primitiva de quatro maneiras diferentes. Dizemos que o simplexo :

- a) localiza-se no interior da primitiva - quando os coeficientes extremos da primitiva são ambos negativos. Classificamos o simplexo nesse caso como *interior*.
- b) localiza-se no exterior da primitiva - quando os coeficientes extremos da primitiva são ambos positivos. Classificamos o simplexo nesse caso como *exterior*.
- c) *pode* cruzar a superfície da primitiva - quando os coeficientes extremos tem sinais opostos. Classificamos o simplexo nesse caso como *interceptante*.
- d) cruza a superfície da primitiva - quando os coeficientes nos vértices do simplexo tem sinais opostos. Classificamos o simplexo nesse caso como *P-interceptante*.

Dentro de nosso algoritmo, a classificação de um simplexo como *interior* ou *exterior* implica na sua eliminação do processo de refinamento, já que garantidamente ele não conterà parte da superfície definida pelos zeros da função característica. Na prática, isso corresponde à sua remoção da *ListaAAvaliar*.

Simplexos classificados como *interseptantes* não nos fornecem informação suficiente para sua eliminação do processo de refinamento, já que essa classificação deixa um margem de dúvida quanto à real posição do simplexo com relação à superfície. Nesse caso, o simplexo é refinado e seus filhos mantidos no processo de refinamento (na *ListaAAvaliar*). Como nosso critério é invariante a inclusões, é razoável esperar que durante o processo de refinamento os seus descendentes possam ser classificados com mais segurança.

O último caso, *P-Interceptante*, baseai-se na análise do sinais dos coeficientes nos vértices do simplexo. Essa classificação só deve ser utilizada ao final do processo de refinamento, durante o cálculo da aproximação linear. Se aplicarmos essa avaliação durante o processo de refinamento estaremos correndo o risco de eliminar do processo simplexos em que não é possível, dado seu nível de refinamento, caracterizar a interseção da superfície em seu interior (como na figura 2.3).

Em resumo, simplexos serão eliminados do processo de refinamento, pelo critério de interferência, toda vez que os coeficientes máximo e mínimo, que representam a função característica da primitiva, tiverem sinais idênticos. Do contrário o simplexo é refinado e seus filhos colocados na *ListaAAvaliar*. Quando o processo de refinamento chegar ao seu final, todos os simplexos da *ListaAprovados* são submetidos a nova avaliação pela análise dos coeficientes da primitiva nos vértices dos simplexos. Para aqueles classificados como *P-Interseptantes*, a aproximação linear é calculada.

Com essa nova base para representação das primitivas que compõem o sólido CSG, e com o critério de interferência acima apresentado, somos capazes de guiar o processo de construção da triangulação adaptativa do seu domínio em sua componente seletiva. No entanto, resta-nos ainda a tarefa de compor os resultados das classificações individuais de cada primitiva, em um resultado único, associado ao sólido como um todo.

### 5.4.1 Classificação da Árvore CSG

Na seção anterior definimos um método de classificação dos simplexos quanto ao seu interior conter ou não parte dos zeros da função polinomial que define uma primitiva. No entanto, os sólidos em geral são compostos por várias primitivas, associadas entre si por operadores booleanos regularizados. Nesse contexto temos que, de alguma forma, poder compor os resultados obtidos na classificação de cada primitiva de modo a obter uma classificação única para aquele simplexo, com relação ao sólido.

Um algoritmo para classificação de um simplexo contra a árvore CSG do sólido deve basicamente percorrê-la. Em cada folha, onde estarão representadas as primitivas, o método anteriormente descrito deve ser aplicado. Nos nós internos da árvore, onde os operadores booleanos se encontram, teremos que compor os resultados da classificação de suas subárvores direita e esquerda, obtendo a classificação do sólido intermediário definido naquele nó. A classificação obtida para a raiz da árvore CSG será portanto o resultado da classificação do simplexo com relação ao sólido.

A tabela 5-1 resolve a classificação para todas as possíveis combinações de resultados em nós intermediários, com base nos resultados associados a suas subárvores. As classificações *int*, *out* e *inter* correspondem, respectivamente, a *interior*, *exterior* e *interceptante*.

Operação	$\cap$			$\cup$		
	Int	Ext	Inter	Int	Ext	Inter
Int	<i>Int</i>	<i>Ext</i>	<i>Inter</i>	<i>Int</i>	<i>Int</i>	<i>Int</i>
Ext	<i>Ext</i>	<i>Ext</i>	<i>Ext</i>	<i>Int</i>	<i>Ext</i>	<i>Inter</i>
Inter	<i>Inter</i>	<i>Ext</i>	<i>Inter</i>	<i>Int</i>	<i>Inter</i>	<i>Inter</i>

*Tabela 5-1 Classificação de um nó intermediário de uma árvore CSG baseada nos resultados das classificações de suas subárvores.*

Em modelos reais, o número de primitivas que tomam parte da representação do sólido pode ser bastante elevado. Nesses casos, a utilização, durante todo o processo de refinamento, da árvore CSG completa pode ser bastante custosa. Esse problema afeta a etapa de classificação da árvore, já que para cada simplexo teremos que ter uma classificação da primitiva perante a árvore completa. Além disso, e mais crítico, é o fato de que para cada primitiva uma aplicação do método de Casteljaou se fará necessário para a geração dos coeficientes dos simplexos filhos.

A tarefa de classificação da árvore pode ser simplificada se levarmos em conta o fato de que, à medida que o algoritmo de subdivisão adaptativa avança, menores são os simplexos, e portanto menor é a porção da superfície do sólido neles contida. Em geral, as primitivas exercem influência sobre porções localizadas da superfície do sólido; portanto quanto menor a porção da sua superfície dentro do simplexo, menor deve ser a quantidade de primitivas que exercem influência direta naquela região.

Levando-se em conta esses fatos podemos pensar em simplificar a árvore que representa o sólido, eliminando as primitivas que não exercem qualquer influência na porção da superfície do sólido, contida em um dado simplexo, durante o processo de refinamento.

## 5.4.2 Localização de um Sólido

A idéia de localização de um sólido tem como objetivo eliminar da sua árvore CSG as primitivas que não exercem influência na superfície do sólido, para uma dada região associada a um simplexo. Em [TILO84] essa técnica é empregada no contexto da detecção de objetos nulos em esquemas CSG. A localização de um árvore CSG pode ser feita de forma simples aplicando-se as seguintes relações, baseadas na álgebra booleana (onde  $W$  representa o conjunto universo) :

$$1) A \cap \emptyset = \emptyset$$

$$2) A \cap W = A$$

$$3) A \cup \emptyset = A$$

$$4) A \cup W = W$$

Os conjuntos  $\emptyset$  e  $W$  podem ser relacionados com a classificação *interior* e *exterior*, respectivamente, de uma primitiva.

Quando um nó interno da árvore, que contém uma operação booleana, é encontrado, aplicamos as relações acima descritas para substituir toda a subárvore da qual ele é raiz pela primitiva  $A$  (nos casos 2 e 3) ou simplesmente removê-la (nos casos 1 e 4). Essa última hipótese pode ser assim aplicada pois nos dois casos a composição das primitivas resultam em uma classificação *interior* ou *exterior*, e que, portanto, não são de nosso interesse.

### 5.4.3 Algoritmo de Classificação

Aplicando-se a tabela 5-1 e as relações apresentadas para localização em uma árvore CSG, podemos construir um algoritmo que durante a etapa de classificação promova a localização da árvore CSG do simplexo. A seguir apresentamos tal algoritmo [SALI92].

*ClassificaSimplificaCSG (ArvoreCSG, Simplexo);*

*Se (ArvoreCSG = Folha) então*

*Retorna (ClassificaçãoPrimitiva (ArvoreCSG));*

*Senão*

*ClasseEsq = ClassificaSimplificaCSG (FilhoEsquerdo(ArvoreCSG, Simplexo));*

*Se ( ( ClasseEsq = EXT e Operador (ArvoreCSG) =  $\cap$  ) ou*

*( ClasseEsq = INT e Operador (ArvoreCSG) =  $\cup$  ) ) então*

*Elimina(ArvoreCSG);*

*Retorna(ClasseEsq);*

*senão*

*Se ( ClasseEsq = EXT ou ClasseEsq = INT ) então*

*EliminaFilhoEsquerdo(ArvoreCSG);*

*ArvoreCSG = FilhoDireito(ArvoreCSG);*

*Retorna(ClassificaSimplificaCSG(ArvoreCSG, Simplexo));*

*ClasseDir = ClassificaSimplificaCSG (FilhoDireito(ArvoreCSG, Simplexo));*

Se ( ( ClasseEsq = EXT e Operador (ArvoreCSG) =  $\cap$  ) ou  
 ( ClasseEsq = INT e Operador (ArvoreCSG) =  $\cup$  ) ) então  
     Elimina(ArvoreCSG);  
     Retorna(ClasseDir);  
 senão  
     Se ( ClasseDir = EXT ou ClasseDir = INT ) então  
         EliminaFilhoDireito(ArvoreCSG);  
         ArvoreCSG = FilhoEsquerdo(ArvoreCSG);  
         Retorna(ClasseEsq);  
 Retorna (Inter);

Esse algoritmo é recursivo, visto que reflete o percurso de uma árvore binária.

O primeiro teste se refere à condição de parada do processo recursivo. Assim que uma primitiva (folha) é encontrada, sua classificação é obtida de acordo com o critério já definido. Essa classificação é retornada para o nível imediatamente superior no processo recursivo.

Os nós intermediários são classificados e então testados na hipótese de uma simplificação da subárvore por eles definida. Duas situações podem nesse momento ocorrer : ou a sua classificação pode induzir à eliminação de toda subárvore da qual esse nó é raiz (  $A \cap \emptyset = \emptyset$  ou  $A \cup W = W$  ); ou um ramo dessa subárvore pode ser eliminado (  $A \cap W = A$  ou  $A \cup \emptyset = A$  ). Nesse último caso, um ramo é eliminado e uma nova raiz é adotada para aquela subárvore.

Caso os dois ramos de um nó intermediário não possam ser simplificados esse nó é automaticamente classificado como *interceptante*. Vale ressaltar que nesse caso o simplexo não necessariamente intercepta a superfície do sólido; porém, na dúvida, serão mantidos na lista de simplexos a serem avaliados.

## 5.5 Conclusão

Nesse capítulo retomamos o contexto de nossa proposta original : a construção de uma aproximação linear da fronteira de um sólido CSG. Caracterizamos esse problema como um caso particular da aproximação linear de uma superfície definida pelo conjunto de raízes de uma função implícita.

Discutimos os detalhes para implementação de um dos critérios que guiam o algoritmo proposto no capítulo anterior para a construção de uma triangulação adaptativa de um domínio.

Como forma de simplificar a avaliação de nosso critério seletivo, propusemos uma mudança na base para construção das funções características associadas às primitivas CSG. Ao invés de trabalharmos com a base algébrica iremos definir nossas funções características na base de Bernstein. Com base nos coeficientes das primitivas representadas nessa nova base, construímos um critério de classificação dos simplexes quanto à existência ou não de parte do conjunto de raízes da polinomial dentro de um simplexo.

Para simplificar a tarefa de classificação da árvore CSG associada a um simplexo, introduzimos a necessidade de simplificação dessa árvore durante o processo de refinamento da triangulação.

Mostramos também que a mudança da representação das primitivas da base algébrica para a base de Bernstein simplifica também a obtenção dos coeficientes associados aos simplexes filhos após uma subdivisão binária ser aplicada.

No próximo capítulo apresentaremos o segundo critério, guia do processo de refinamento : o *critério adaptativo* ou *de poligonização*. Ele será o responsável pela aferição da qualidade da aproximação linear dentro de um dado simplexo de referência.

# 6. CRITÉRIO PARA POLIGONIZAÇÃO

---

## 6.1 Introdução

No capítulo anterior definimos um dos critérios utilizados por nosso algoritmo para a construção de uma triangulação adaptativa do domínio de uma função polinomial implícita, que representa a superfície de um sólido CSG. Esse critério nos fornecia subsídios para a classificação de um simplexo quanto a sua interferência com a superfície.

Porém para que nossa idéia de adaptatividade seja implementada em sua totalidade precisamos de meios de aferir se, em um dado momento do processo de refinamento, um simplexo produz uma aproximação linear que satisfaz a nossa tolerância. Em caso afirmativo, o processo de refinamento desse simplexo pode ser abortado. Com isso evitamos ter o trabalho de descer desnecessariamente até o último nível de refinamento em todos aqueles simplexos que possuem parte da fronteira do sólido em seu interior, como [SALI92].

Nesse capítulo definiremos como essa avaliação pode ser feita, bem como quais suas restrições de aplicação. Voltaremos também a um ponto bastante importante, que até aqui foi assumido inexistente : vértices que se localizam sobre a superfície do sólido, e que por conseguinte possuem coeficientes de Bézier nulos. Aprofundaremos a discussão desse caso propondo duas soluções para o problema.

Discutiremos ainda como, ao final do processo de construção da triangulação adaptativa, a poligonização será gerada. Também abordaremos, ainda que de maneira superficial nesse momento, algumas características dessa poligonização propondo meios de otimizá-la.

## 6.2 Critério Adaptativo

Conjugado ao critério seletivo, precisamos definir uma forma de avaliar qual o erro que estaremos cometendo ao substituímos a porção real da superfície que intercepta um dado simplexo, em um momento qualquer do processo de refinamento da triangulação, por uma outra superfície linear, fruto de nossa aproximação. Será com base na avaliação desse erro que o nosso algoritmo poderá abortar o processo de



refinamento daqueles simplexos que estejam dentro da margem de tolerância preestabelecida.

No contexto de superfícies definidas implicitamente, podemos construir tal critério baseando nossa análise no valor máximo da derivada de segunda ordem (curvatura). Esse valor pode nos dar uma boa estimativa do erro que estaremos cometendo, se levarmos a cabo uma aproximação linear em determinado momento do processo de refinamento. No entanto, esse critério de avaliação tem como desvantagem seu custo computacional.

Como nossa representação para as funções características, que representam as primitivas, é construída a partir da formulação Bézier-Bernstein, podemos imaginar um critério mais simples, e portanto de menor custo computacional, que seja baseado na análise do conjunto de pontos de controle que essa representação fornece.

Nossa proposta é comparar, de forma absoluta, as diferenças entre os coeficientes de Bézier da polinomial original e os mesmos coeficientes para uma primitiva linear, porém definida no mesmo grau do polinômio original. Tomando-se a diferença máxima alcançada, temos um limitante superior de erro obtido pela aproximação. Esse valor pode ser utilizado então como estimativa do erro cometido.

A representação linear da função pode ser facilmente construída da seguinte forma:

$$L(t, u, v, w) = t \cdot p_{n,0,0,0} + u \cdot p_{0,n,0,0} + v \cdot p_{0,0,n,0} + w \cdot p_{0,0,0,n}$$

onde  $n$  representa o grau da primitiva.

A nossa estimativa de erro pode ser avaliada pela formula :

$$\xi(S, p) \leq \mu(S, p) = \frac{\max\{ \|p_{i,j,k,l} - L_{i,j,k,l}\| \mid i+j+k+l = n \}}{\|\nabla L\|} \tag{6.1}$$

$$\text{onde } L_{ijkl} = \frac{i}{n} p_{n000} + \frac{j}{n} p_{0n00} + \frac{k}{n} p_{00n0} + \frac{l}{n} p_{000n}$$

A medida de erro absoluto  $\xi(S,p)$  representa a distância máxima entre os zeros da polinomial  $p$ , de grau  $n$  definida no simplexo  $S$ , e os zeros da aproximação linear  $L$ .

Nossa estimativa  $\mu(S,p)$  avalia o quão não linear é o conjunto de zeros da polinomial  $p$  dentro de  $S$ .

Seja  $P = \sum_{ijkl} p_{ijkl} B_{ijkl}^n$  um polinômio definido na base de Bernstein. Podemos definir seu interpolador linear através da relação :

$$\begin{aligned} L(u, v, w, t) &= u.P(1,0,0,0) + v.P(0,1,0,0) + w.P(0,0,1,0) + z.P(0,0,0,1) \\ &= u.p_{n000} + v.p_{0n00} + w.p_{00n0} + z.p_{000n} \end{aligned}$$

**Teorema** : O erro entre o polinômio  $P$  e seu interpolador linear dentro do simplexo em que  $P$  está representado é :

$$|\xi(u, v, w, t)| \leq \max_{ijkl} \left\{ \left| p_{ijkl} - L\left(\frac{i}{n}, \frac{j}{n}, \frac{k}{n}, \frac{l}{n}\right) \right| \right\}$$

**Prova** :

O interpolador linear  $L$  pode ser expresso na base Bernstein por meio de [FARI90]:

$$L = \sum_{ijkl} L\left(\frac{i}{n}, \frac{j}{n}, \frac{k}{n}, \frac{l}{n}\right) \cdot B_{ijkl}^n$$

Logo, a diferença entre  $P$  e  $L$  será o polinômio :

$$\xi(u, v, w, t) = \sum_{ijkl} (p_{ijkl} - L\left(\frac{i}{n}, \frac{j}{n}, \frac{k}{n}, \frac{l}{n}\right)) \cdot B_{ijkl}^n$$

Pelas propriedades da base Bernstein, para  $u, v, w, t \geq 0$ , temos que a relação abaixo é correta :

$$\min_{ijkl} (p_{ijkl} - L\left(\frac{i}{n}, \frac{j}{n}, \frac{k}{n}, \frac{l}{n}\right)) \leq \xi(u, v, w, t) \leq \max_{ijkl} (p_{ijkl} - L\left(\frac{i}{n}, \frac{j}{n}, \frac{k}{n}, \frac{l}{n}\right))$$

Daí decorrendo a tese proposta.

A partir do teorema apresentado acima podemos chegar à equação (6.1) que estima o erro cometido na aproximação linear de uma polinomial dentro de um simplexo base. Seja  $S$  um simplexo cujo erro absoluto vale  $\bar{\xi}$ . Nesse caso vale a relação:

$$L - \bar{\xi} \leq P \leq L + \bar{\xi}$$

Para  $Z(P)$ , o conjunto de zeros da polinomial  $P$  dentro do simplexo  $S$ , podemos afirmar que :

$$-\bar{\xi} \leq L \leq \bar{\xi}$$

Seja  $p$  um ponto pertencente ao conjunto  $Z(P)$ . Então :

$$-\bar{\xi} \leq L(p) \leq \bar{\xi}$$

Seja  $q$  um ponto pertencente ao conjunto  $Z(L)$ , conjunto de raízes do interpolador linear dentro de um simplexo  $S$ , resultado da projeção ortogonal de  $p$  em  $Z(L)$ . Então :

$$L(p) = L(q) + \langle p - q, \nabla L \rangle = \langle p - q, \nabla L \rangle$$

Mas :

$$\langle p - q, L \rangle = \pm \|p - q\| \cdot \|\nabla L\|$$

Logo :

$$\bar{\xi} \geq |L(p)| = \|p - q\| \cdot \|\nabla L\|$$

Daí podemos concluir que :

$$\|p - q\| \leq \frac{\bar{\xi}}{\|\nabla L\|}$$

A figura 6.1 mostra a interpretação geométrica para essa expressão para uma polinomial uni-variada de grau 3.

O critério acima definido nos fornece uma boa estimativa do erro na substituição da função polinomial original por seu interpolador linear, quando apenas uma primitiva está definida dentro do simplexo.

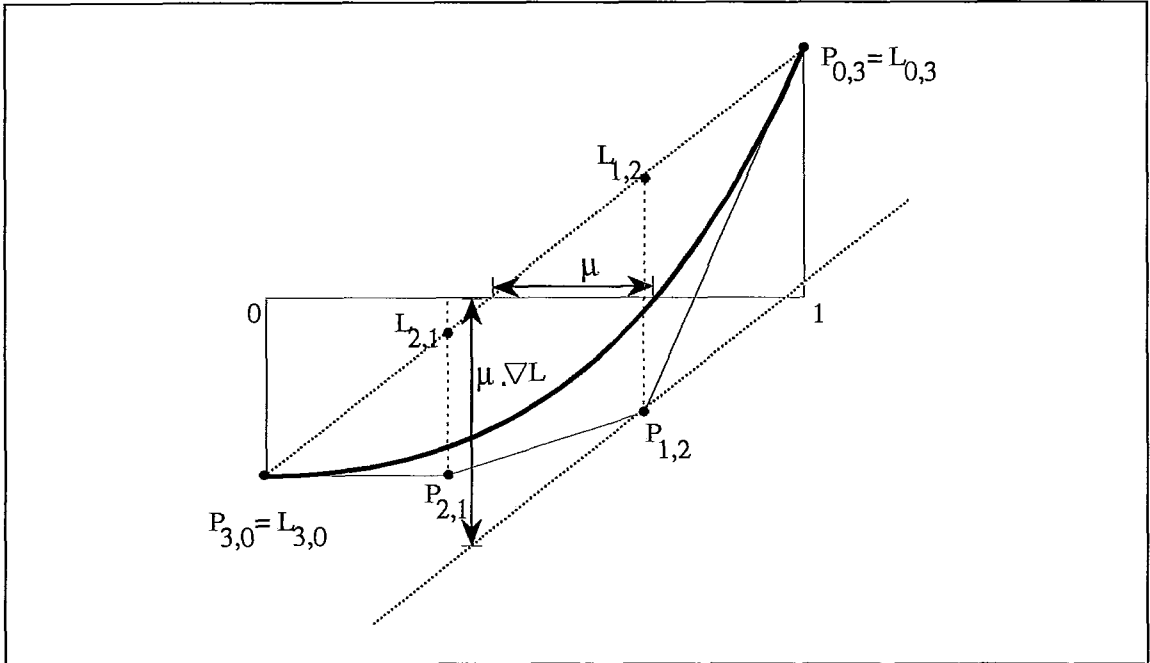


Figura 6.1 Interpretação geométrica para a avaliação do critério de precisão para a poligonização.

Nos casos em que a árvore CSG associada ao simplexo possui mais de uma primitiva, a polinomial que aproxima o comportamento da função dentro do simplexo será obtida pela aplicação dos operadores *min/max*. Nesse caso torna-se difícil obtermos uma boa aproximação do erro cometido na linearização já que a função característica pode nem mesmo ser diferenciável [PERS94].

Para essas situações, tomaremos como estimativa do erro cometido na aproximação linear o diâmetro do simplexo, ou seja, o comprimento de sua maior aresta, que certamente superestima o erro cometido. De posse do tamanho dos simplexos da triangulação inicial podemos calcular a partir de qual nível de refinamento  $n$  o diâmetro do simplexo será inferior a precisão estipulada.

Para isso basta levarmos em consideração que, a cada três níveis no processo de refinamento as dimensões da maior aresta do simplexo inicial se reduz à metade. Considerando  $l$  o valor dessa maior aresta,  $n$  o nível máximo de refinamento e  $\xi$  o valor máximo de erro aceitável, podemos montar a seguinte relação :

$$\frac{l}{2^{(n/3)}} \geq \xi$$

Podemos assim determinar o nível ideal de refinamento, com base no tamanho da maior aresta do simplexo original e na precisão estabelecida, de modo que a tolerância especificada seja atendida.

$$n \leq 3 \log(1/\xi)$$

Tipicamente simplexos contendo arestas e vértices do modelo serão refinados até esse nível máximo, já que suas árvores, mesmo que simplificadas, irão conter mais de uma primitiva.

Nossa triangulação final da fronteira de um sólido terá como característica ser muito refinada (nível máximo) nas arestas e vértices. Em regiões cuja curvatura é quase nula teremos regiões pouco refinadas. Entre esses extremos serão criadas regiões de transição cujos elementos terão níveis variados de refinamento.

Isso se deve a dois fatores básicos. O primeiro decorrente do próprio processo de refinamento adaptativo, que garante a manutenção permanente da condição da subdivisão formar uma triangulação (via refinamento básico). Junto a isso temos a propriedade das triangulações adaptativas, baseadas nas extensões das triangulações CFK, que nos garante que entre simplexos vizinhos há no máximo diferença de uma unidade no nível de refinamento.

Com isso podemos concluir que o grau de refinamento de uma dada região é influenciado não só pelo grau de curvatura da região (informação básica para o critério de precisão) mas também pela proximidade de uma aresta, vértice ou mesmo região de alta curvatura.

Uma proposta de critério semelhante ao apresentado acima, pode ser encontrada em [WARR90]. Em sua proposta, dois planos são criados, translações de  $+\xi$   $-\xi$  (sendo  $\xi$  a tolerância) do plano definido pela aproximação linear  $L$ . Caso a superfície esteja totalmente contida na região definida pelos dois planos o simplexo é dado por aprovado.

### ***6.3 Cálculo da Aproximação***

Definido o último critério que guiará nosso algoritmo para construção da triangulação adaptativa do domínio da função, que define a fronteira do sólido CSG a ser aproximado, podemos detalhar melhor o processo de construção da poligonização. Essa poligonização tentará representar o comportamento do interpolador linear definido no capítulo 2, naqueles simplexos interceptados pela superfície do sólido.

Antes, porém, devemos voltar a uma discussão deixada em aberto no mesmo capítulo. Naquele instante presumimos que essa situação não iria ocorrer por simplificação. No entanto, na prática, ela ocorre, e até com uma frequência razoável.

Estamos nos referindo   possibilidade de um v rtice da triangula o se localizar sobre a superf cie do s lido a ser aproximado.

### 6.3.1 V rtices que Anulam a Fun o Caracter stica

Uma importante propriedade apresentada em [ALLG80], no contexto dos m todos simpliciais, nos permite afirmar que, se em uma triangula o de um dom nio, nenhum v rtice est  localizado sobre a fun o em an lise, garantimos que a sua aproxima o linear   uma variedade. Em outras palavras se tomarmos um ponto qualquer pertencente   aproxima o e analisarmos sua vizinhan a, esta ser  homeomorfa a um disco. Essa garantia   particularmente  til no nosso caso, j  que a superf cie que desejamos aproximar equivale a superf cie de um s lido regularizado, ou seja, uma variedade tridimensional.

Essa propriedade nos permite garantir tamb m que, mesmo que tenhamos um s lido (fun o) original que n o   uma variedade, sua aproxima o o ser , caso nenhum v rtice da triangula o esteja sobre a superf cie.

Analisaremos com mais cuidado os problemas que um v rtice localizado sobre a superf cie traz ao nosso esquema de constru o de uma poligoniza o. Para isso tomemos como exemplo o caso representado na figura 6.2 . Novamente apresentamos um exemplo bidimensional para facilidade de representa o.

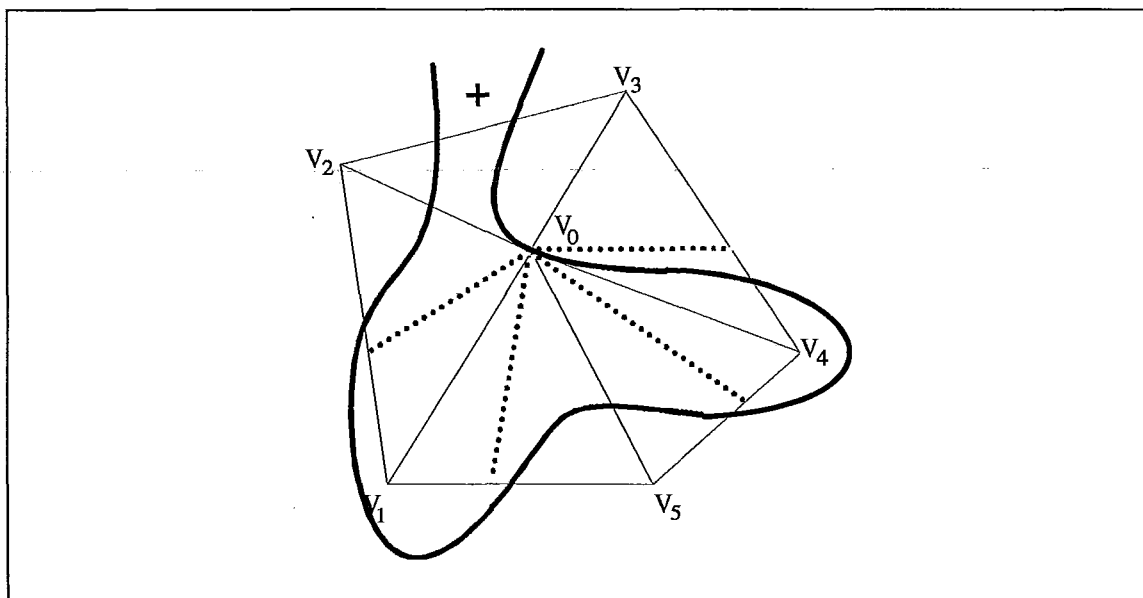


Figura 6.2 Exemplo de triangula o em que um de seus v rtices anula a fun o caracter stica de uma primitiva.

Na figura 6.2 o v rtice  $V_0$  est  localizado sobre a superf cie. Ao analisarmos o simplexo definido pelos v rtices  $V_0, V_1$  e  $V_2$ , do ponto de vista do crit rio seletivo, a aresta  $V_1V_2$  ser  garantidamente interceptante. No entanto, a classifica o das duas outras arestas, que t m como um de seus v rtices  $V_0$ , deve ser feita com cautela. Caso a classifica o das arestas seja *interceptante*, iremos aplicar o interpolador linear j  definido para o c lculo da solu o aproximada nessa aresta. Como resultado teremos, em ambas as arestas, o pr prio v rtice  $V_0$ . A mesma situa o ocorrer  com todos os simplexos que compartilham o v rtice  $V_0$  e possuam uma outra aresta, da qual esse v rtice n o faz parte, classificada como *interceptante*. Na figura 6.2 vemos claramente que segundo essa hip tese quatro arestas pertencentes   poligoniza o final compartilham o v rtice  $V_0$ , caracterizando portanto uma n o variedade.

Uma proposta de solu o para esse tipo de problema   apresentada por [CAST90], e pode ser resumida   aplica o de uma perturba o na geometria do v rtice e o correspondente recalculo do valor da fun o caracter stica nesse novo ponto. Essa perturba o deve ser feita de modo que seja id ntica em todos os simplexos que compartilham esse v rtice. A figura 6.3 (direita) apresenta como essa perturba o pode ser aplicada quando um v rtice nulo   detectado, usando como base o exemplo apresentado na figura 6.2 .

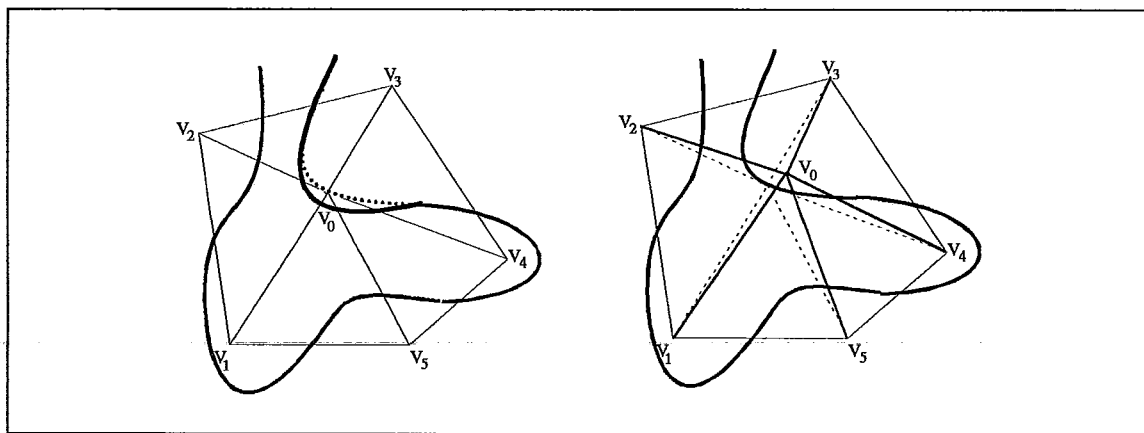


Figura 6.3 Solu es para o problema de v rtices que anulam a fun o caracter stica de uma primitiva :   esquerda perturba o do coeficiente associado   primitiva naquele v rtice;   direita a perturba o na geometria do v rtice.

Outra solu o pass vel de ser aplicada para esse problema   tamb m baseada em uma perturba o, por m feita no coeficiente daquele v rtice que anula a fun o caracter stica da primitiva. Com base na toler ncia  $\xi$  estabelecida, podemos determinar um limite m ximo para essa perturba o. Analisando os coeficientes vizinhos ao v rtice que anula a fun o, e tomando seu m ximo, podemos definir um valor para essa perturba o segundo a rela o :

$$C_{0n00} = \frac{\xi \cdot \max \{C_{1n-100}, C_{0n-110}, C_{0n-101}\} \cdot n}{\text{tamanho do Simplexo}}$$

Na verdade, ao perturbarmos esse coeficiente estamos fazendo com que a função sofra um pequeno “desvio” nas proximidades daquele vértice. A figura 6.3 (esquerda) ilustra essa outra solução no contexto do exemplo apresentado na figura 6.2 .

Essa solução, assim como a perturbação geométrica, deve garantir que a alteração do coeficiente seja feita de forma coerente, ou seja, todos os simplexos que compartilham aquele vértice devem sofrer a mesma perturbação.

Em nossa implementação, optamos por aplicar a perturbação ao coeficiente da primitiva associada a um determinado vértice da triangulação. Durante o processo de subdivisão binária (refinamento básico) temos a garantia de que apenas um novo vértice é inserido na triangulação, quando do corte de um núcleo. Portanto tão logo esse vértice seja gerado podemos testar se alguma das primitivas associadas aos simplexos que o compartilham (simplexos do núcleo) se anula nesse novo vértice. Em caso afirmativo aplicamos uma perturbação e a propagamos por todos os simplexos do núcleo.

Esse procedimento tem a vantagem de garantir que em nenhum instante do processo de construção da triangulação adaptativa teremos vértices anulando a função característica de uma primitiva.

### 6.3.2 Algoritmo para Poligonização

Ao final do processo de construção da triangulação adaptativa, teremos uma lista de simplexos avaliados e aprovados. Alguns desses serão aprovados por atenderem ao critério de precisão estabelecido, enquanto que outros o serão por terem alcançado o nível máximo de refinamento associado à tolerância do modelo.

Será a partir dessa lista que iremos proceder à construção de nossa poligonização, analisando o interpolador linear já definido. Essa tarefa só pode ser feita no momento em que não haja mais simplexos a avaliar. Isso pelo fato de que nosso procedimento de refinamento básico poder eventualmente propagar subdivisões binárias por elementos já aprovados, por conta da manutenção da condição de triangulação da subdivisão.

O algoritmo, em pseudocódigo para a construção da poligonização é apresentado a seguir.



*GeraPoligonização(ListaAprovados)*

*enquanto (ListaAprovados  $\neq$  vazia) faça*

*Se ( Classificação(ListaAprovados->Simplexo) = P-Interceptante ) então*

*CalculaAproximaçãoPoligonal (Simplexo);*

*RemoveLista(ListaAprovados->Simplexo);*

Partindo da *ListaAprovados* iremos reavaliar cada um dos simplexos através da função *Classificação*. Como mencionamos no capítulo anterior, as classificações dos simplexos durante o processo de refinamento da triangulação são feitas baseadas nos maiores e menores coeficientes de Bézier na representação das primitivas na base de Bernstein. No entanto, no momento de gerar a poligonização, utilizaremos os valores dos coeficientes nos vértices, que nos fornecem uma classificação mais precisa quanto à existência ou não de parte do conjunto de raízes de nosso interpolador linear dentro do simplexo, numa dada precisão.

Caso essa reavaliação classifique o simplexo em questão como *P-Interceptante* ativamos a função *CalculaAproximaçãoPoligonal* que constrói a face poligonal aproximadora da função dentro do simplexo. Esse cálculo é feito baseado na geração dos pontos de interseção do interpolador linear em cada aresta. O conjunto de pontos assim obtidos são agrupados de maneira adequada a formar uma face (polígono) planar. Vale observar que a organização desse conjunto de pontos é importante na medida em que servirá de aproximação para uma superfície que possui uma orientação que deve ser mantida.

Cada simplexo após ser avaliado é eliminado de *ListaAprovados*. O processo é retomado até que o último simplexo dessa lista seja avaliado.

Nossa poligonização será descrita no final do processo a partir de uma lista indexada de vértices, na qual especificaremos suas coordenadas. Outras informações, tais como cor do vértice, valor da normal, etc, podem ser acrescentadas de acordo com a necessidade da aplicação. Logo após essa lista de vértices, temos uma outra lista descrevendo as faces da poligonização, através da relação de vértices que as formam. Nessa relação, os vértices são descritos pelos seus respectivos índices.

Através da estrutura da triangulação podemos recuperar as informações de vizinhança dos simplexos e com isso evitar que a lista de vértices gerada pelo nosso algoritmo possua elementos repetidos. Dado esse fato somos capazes de, a partir da poligonização, gerada no formato acima, montar uma representação B-Rep de nosso sólido de forma bastante simples[HOTT92].

Tal fato não é possível no contexto da proposta de subdivisão seletiva, já que informações topológicas não são mantidas durante o processo de refinamento de subdivisão. O resultado final, por ele obtido, se restringe a um conjunto de faces

planares, sem qualquer relação entre elas. Para conseguirmos montar um B-Rep através da sua aproximação teríamos que, por meio de testes geométricos, identificar vértices comuns e a partir daí recuperar as relações topológicas entre as faces.

## 6.4 Qualidade das Faces da Aproximação

O formato das faces geradas pelo nosso algoritmo está relacionado com o posicionamento da superfície dentro do domínio e a localização do simplexo com respeito a essa função. Dessa forma, não podemos garantir que não existirão faces da poligonização cujo formato seja ruim, ou seja, possuam ângulo interno mínimo muito pequeno ou tenham área muito próxima de zero. Dois casos são ilustrados na figura 6.4 .

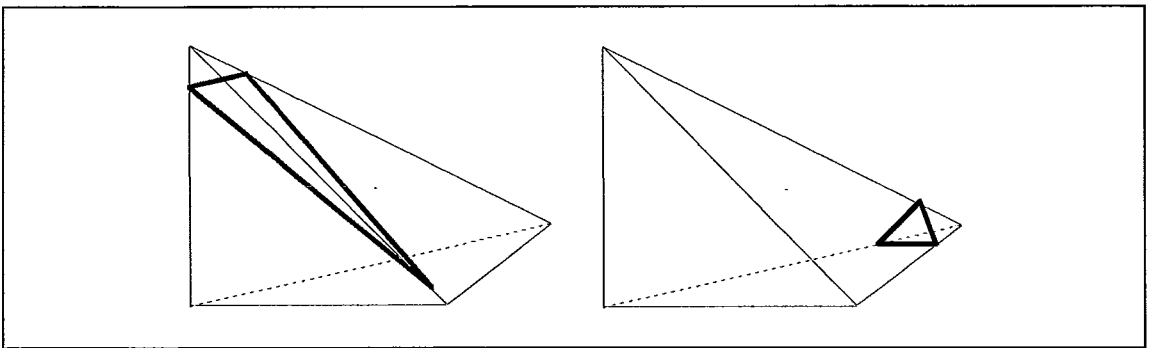


Figura 6.4 Exemplo de faces cuja área é próxima de zero (à direita) e em que uma das arestas é muito pequena frente às dimensões das demais (à esquerda), geradas pelo algoritmo de poligonização.

Para algumas aplicações tais faces podem se constituir em um problema, já que, dependendo do grau de precisão adotado, essas faces podem ser encaradas como faces que se degeneraram em arestas (quando dois dos vértices da aproximação são muito próximos - figura à esquerda) ou mesmo em um único ponto (figura à direita, onde todos os três pontos podem ser considerados coincidentes).

Apresentaremos a seguir uma idéia para minimizar esse problema, removendo essas faces da poligonização de forma controlada.

### 6.4.1 Colapso de Vértices

O ponto central do problema reside no fato de dois vértices que geometricamente seriam coincidentes, se considerarmos uma certa precisão, estarem representados na triangulação como vértices distintos. Para eliminar o problema podemos então fazer com que a informação geométrica predomine. Ou seja, devemos fazer com que geométrica e topologicamente os dois vértices coincidam.

Essa idéia implica em reduzir as faces geradas na aproximação a arestas ou a pontos. Dessa forma podemos retirar da triangulação essas faces sem que “buracos” sejam gerados. Nessas situações diremos que um dos vértices considerados coincidentes, dentro da precisão utilizada, será *colapsado*, tornando-se idêntico ao outro.

Para que o colapso de um vértice possa ser feito de maneira coerente, iremos utilizar uma função de mapeamento para os vértices. Baseado na lista de vértices gerada como representação da poligonização, podemos montar uma função que associe cada vértice ao seu índice correspondente. Em outras palavras, a função de mapeamento terá por objetivo vincular o índice de um vértice com a posição na lista onde sua informação geométrica está localizada.

Para que dois vértices, próximos segundo alguma precisão, sejam colapsados basta que a função de mapeamento gere, para ambos, o mesmo índice referente à lista de vértices. Dessa forma, ao montarmos uma face, com base nos índices providos por essa função, os vértices que forem colapsados deverão apontar para mesma posição na lista de vértices.

Se olharmos no contexto global da aproximação linear, o colapso de um vértice tem o efeito de “arrastar” todas as arestas que compartilham esse vértice. Por conseqüência as faces que contêm tais arestas serão “alargadas” ou “afinadas” de acordo com o sentido da movimentação. A figura 6.5 ilustra esse efeito.

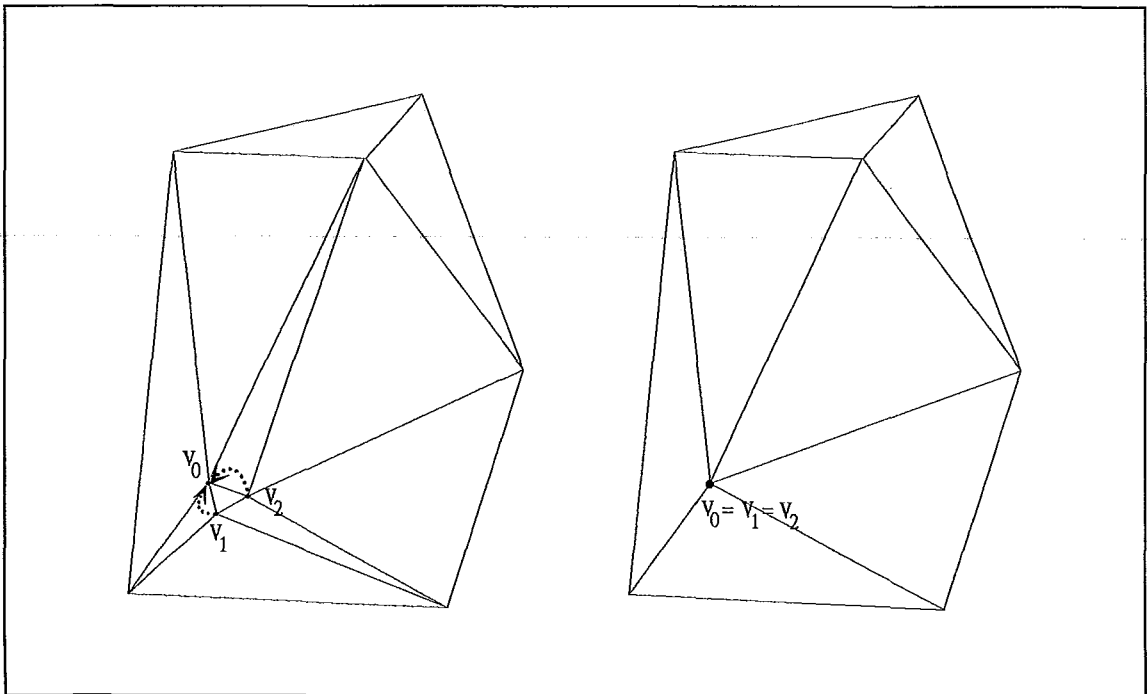


Figura 6.5 Exemplo do efeito causado pelo colapso de dois vértices ( $V_1$  e  $V_2$ ) sobre a poligonização.

Nesse caso supomos que os vértices  $V_0$ ,  $V_1$  e  $V_2$  foram detectados como vértices candidatos à colapso devido à sua proximidade. Nesse caso ao colapsarmos esses vértices estaremos eliminando a face por eles formada em um único ponto, no caso  $V_0$ . Ao mesmo tempo estamos também determinando o colapso de todas as suas faces vizinhas, que no caso também seriam colapsadas, já que uma de suas arestas teria vértices muito próximos.

As faces que têm seus vértices colapsados não devem ser geradas, já que elas agora formam faces degeneradas em arestas ou mesmo vértices.

A construção da função de mapeamento será feita durante a geração da lista de vértices. Para cada face gerada, processamos uma verificação quanto à distância entre seus vértices, associados dois a dois. Caso essa distância esteja aquém da precisão estabelecida, promovemos o colapso dos vértices adequados através da alteração na função de mapeamento.

Quando da geração da lista de faces, consultamos a função de mapeamento para que os vértices dessa lista sejam gerados, já levando em conta os possíveis colapsos de vértices. Para aqueles simplexes que tiveram algum de seus vértices colapsados, não geramos nenhuma entrada nessa lista. Dessa forma eliminamos por completo as faces de área pequena ou cujo ângulo interno mínimo seja muito pequeno.

Na literatura, outras estratégias para eliminação de faces de um modelo representado por um conjunto de faces planares são apresentadas. Uma solução é a apresentada em [WARR90]. Sua proposta é mover vértices dos simplexes que estejam muito “perto” da superfície a ser aproximada. “Perto” nesse caso significa a relação entre a distância do ponto a superfície e o tamanho do simplexo. Esses vértices serão posicionados sobre a superfície e nesse caso tratados de maneira adequada. Outra possibilidade seria a aplicação, em uma fase de pós-processamento, de técnicas para otimização de malhas triangulares como em [HOPP93].

## ***6.5 Conclusão***

Nesse capítulo definimos o segundo e último critério utilizado para guiar o processo de construção da triangulação adaptativa : o critério de precisão, responsável pela avaliação do erro cometido ao se aproximar o comportamento da função original por uma linear, em um determinado ponto do processo. Esse critério está baseado na representação Bézier-Bernstein das primitivas dentro dos simplexes e nos fornece um limitante superior para esse erro. Sua utilização, no entanto, está restrita à avaliação de erro em simplexes que possuem apenas uma primitiva passando por seu interior.

De posse da triangulação adaptativa final, discutimos um algoritmo para a construção da poligonização da superfície definida pela função característica do sólido.

Nesse momento retomamos a discussão do caso em que um vértice se localizava sobre a superfície, discutindo uma solução apontada por [CAST90] de perturbar a geometria desse vértice. Abordamos também uma solução alternativa que perturba não o vértice mas o coeficiente a ele associado.

Iniciamos também a discussão a respeito da qualidade da poligonização gerada pelo nosso algoritmo. Abordamos apenas o problema das faces muito pequenas ou alongadas, onde um par ou trio de vértices estão próximos demais uns dos outros, com respeito a uma determinada precisão, propondo como solução o colapso desses vértices. Dessa forma tais faces são removidas da poligonização sem qualquer quebra na topologia da poligonização.

No próximo capítulo apresentaremos e discutiremos com mais critério os resultados obtidos com nossa nova proposta, fazendo uma análise comparativa dos resultados. Além disso discutiremos algumas propostas para trabalhos futuros.

# 7. AVALIAÇÃO E CONSIDERAÇÕES FINAIS

---

## 7.1 Introdução

O propósito desse trabalho foi construir uma aproximação linear por partes (poligonização) da superfície definida pela fronteira de um sólido descrito no esquema de representação CSG. De posse dessa poligonização, podemos aplicar algoritmos para visualização, cálculo de propriedades inerentes à sua superfície ou mesmo proceder uma conversão para o esquema de representação por fronteira (B-Rep).

O processo de obtenção dessa aproximação baseia-se em uma subdivisão do espaço onde o sólido está imerso. Essa subdivisão leva em consideração dois importantes fatores para que o trabalho seja minimizado. Primeiro elimina da análise, tão cedo quanto possível, regiões do espaço que garantidamente não contêm parte da fronteira a ser aproximada. Baseado no que denominamos *critério seletivo*, classificamos essas regiões do espaço como *interiores*, *exteriores* ou *interseptantes* com relação a superfície a ser aproximada. Com isso, temos condições de decidir que regiões devem ou não ser abandonadas durante o processo de refinamento.

Também consideramos na análise de cada região do espaço, definida durante o processo de refinamento, se a aproximação linear da parte da fronteira que cruza seu interior já é aceitável dentro de uma dada precisão estabelecida. A esse critério de avaliação da qualidade da aproximação linear em um dado ponto do processo de refinamento denominamos *critério adaptativo* ou *de poligonização*, já que uma região que preencha os requisitos desse critério não necessita de mais refinamentos.

Combinados esses dois critérios, temos uma forma de guiar os esforços computacionais envolvidos no processo de subdivisão espacial, direcionando-os a regiões relevantes do espaço e estabelecendo um nível mínimo de refinamento localizado.

A seguir analisaremos alguns dos resultados obtidos em uma implementação computacional das idéias apresentadas até essa etapa do trabalho. Apresentaremos também que pontos ainda podem ser atacados no sentido de melhorar ainda mais os resultados obtidos, bem como sugestões para futuros trabalhos nessa mesma linha.

## 7.2 Avaliação do Esquema Proposto

Nessa seção apresentaremos uma análise do método proposto para construção da poligonização da fronteira de um sólido. Em primeiro lugar, analisaremos a complexidade do algoritmo proposto, nos baseando no número de simplexes gerados durante o processo de refinamento.

A seguir discutiremos, através de quatro sólidos de exemplo (figura 7.1), a capacidade de adaptação do algoritmo fazendo uma análise comparativa com o esquema de subdivisão seletiva apresentado em [SALI92], e sobre o qual desenvolvemos o critério de adaptação.

### 7.2.1 Análise da Complexidade do Algoritmo

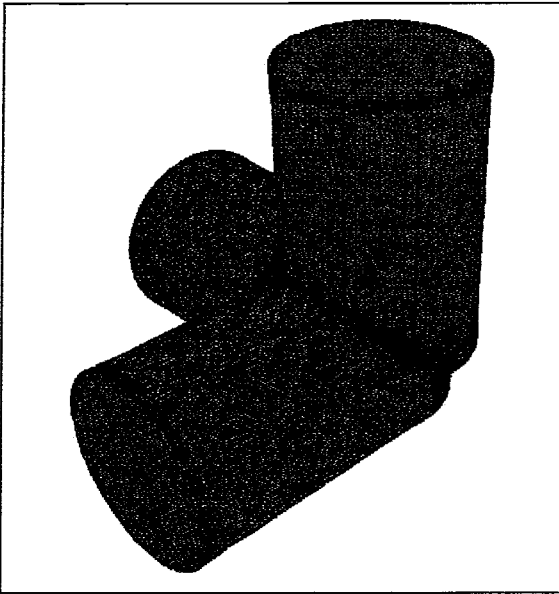
Nosso algoritmo foi desenvolvido partindo da idéia apresentada por Salim para construção de uma subdivisão simplicial do espaço de forma seletiva, guiando o processo de refinamento de modo a atuar apenas em regiões interceptadas por parte da fronteira do sólido. Nossa proposta veio introduzir um grau maior de adaptatividade no sentido de eliminar a necessidade de refinamento também em regiões com parte da fronteira em seu interior. A condição para essa parada é que a aproximação linear da fronteira dentro da região satisfaça a uma certa tolerância estabelecida.

Nesse sentido, podemos fazer um paralelo entre a complexidade dos dois algoritmos, no que diz respeito à quantidade de avaliações de primitivas durante o processo de construção da aproximação linear. Em seu trabalho, Salim compara seu método com mais outros três, teóricos : Método de Avaliação Completo (MAC), Método de Avaliação Ideal (MAI) e Método de Avaliação Ótimo (MAO). Para nossa análise apenas o MAO será relevante. Nesse método, presumimos o conhecimento prévio não só de um simplexo inicial para cada componente conexa, como também quais primitivas devem ser avaliadas em cada um dos simplexes mínimos que contém parte da fronteira do sólido.

A complexidade do MAO é analisada em [SALI92] que apresenta a ordem do custo de avaliação da árvore completa do sólido em um simplexo  $S$ , estimada pela seguinte relação :

$$O(\text{Avaliar } S) = \frac{1}{l^2} \sum_{i=1}^{N_{Prim}} n_i^4 \cdot A_i$$

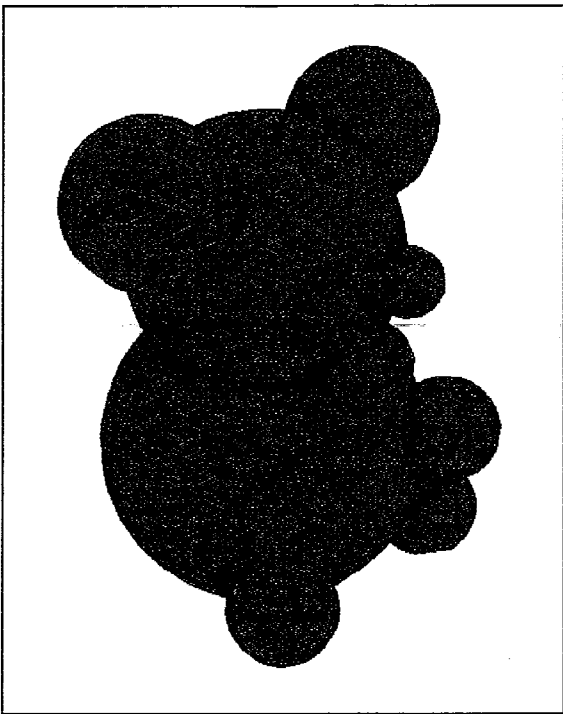
onde  $l$  representa o tamanho da maior aresta do simplexo mínimo  $S$  da triangulação,  $n_i$  representa o grau da primitiva  $P_i$  e  $A_i$  a área da fronteira do sólido para qual a primitiva  $P_i$  contribui.



```

3Cilindros {
  cilindro1 { ESCALA[2,2,6]CILINDRO
  }
  cilindro2 { ROTACAO[90,0,0]cilindro1
  }
  cilindro3 { ROTACAO[0,90,0]cilindro1
  }

  cilindro1 + cilindro2 + cilindro3
}
    
```

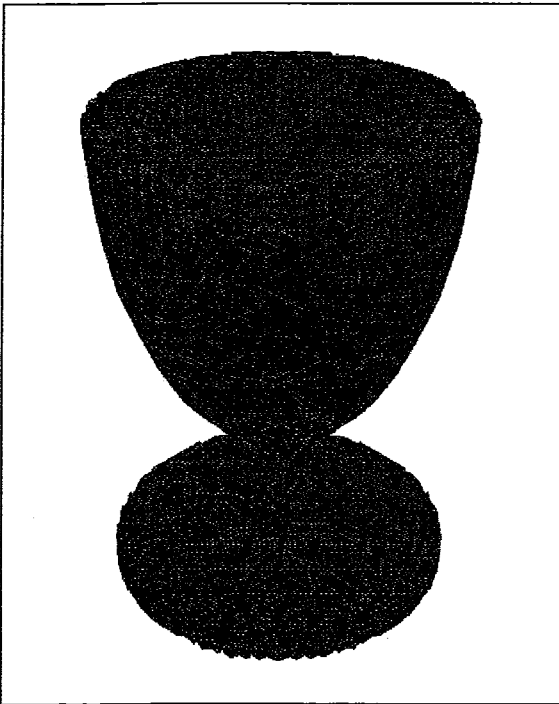


```

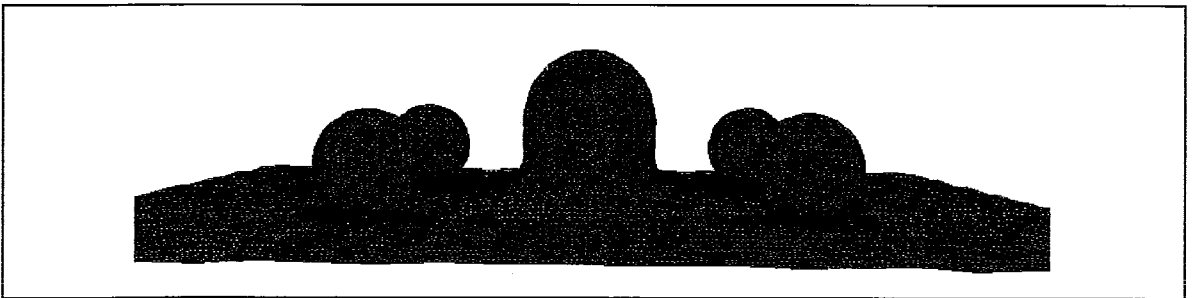
Urso {
  Barriga { MATERIAL[1] ESFERA }
  BracoEsq { MATERIAL[5] ESCALA[0.32,0.32,0.32]
  TRANSL[0.65,0.2,-0.9] ESFERA }
  BracoDir { MATERIAL[5] ESCALA[0.32,0.32,0.32]
  TRANSL[-0.65,0.2,-0.9] ESFERA }
  PeEsq { MATERIAL[5] ESCALA[0.32,0.32,0.32]
  TRANSL[0.65,-0.45,-0.8] ESFERA }
  PeDir { MATERIAL[5] ESCALA[0.32,0.32,0.32]
  TRANSL[-0.65,-0.45,-0.8] ESFERA }
  Cabeca { MATERIAL[1] ESCALA[0.8,0.8,0.8]
  TRANSL[0,1.3,0] ESFERA }
  OrelhaEsq { MATERIAL[5] ESCALA[0.45,0.45,0.45]
  TRANSL[-0.8,1.85,0] ESFERA }
  OrelhaDir { MATERIAL[5] ESCALA[0.45,0.45,0.45]
  TRANSL[0.8,1.85,0] ESFERA }
  Nariz { MATERIAL[5] ESCALA[0.25,0.25,0.25]
  TRANSL[0,0.9,-0.8] ESFERA }
  OlhoEsq { MATERIAL[5] ESCALA[0.2,0.2,0.2]
  TRANSL[-0.3,1.3,-0.8] ESFERA }
  OlhoDir { MATERIAL[5] ESCALA[0.2,0.2,0.2]
  TRANSL[0.3,1.3,-0.8] ESFERA }
  Corpo { ROTACAO[90, 0, 0] (
  escala[2, 2, 2] (
  Barriga + BracoEsq + BracoDir + PeEsq +
  PeDir + Cabeca + OrelhaEsq + OrelhaDir +
  Nariz + OlhoEsq + OlhoDir)
  )
  }
  Corpo
}
    
```

Figura 7-1 Imagem e definição dos modelos CSG : 3 Cilindros e Urso.





```
calice {
  haste { rotacao [-90, 0, 0] escala [0.1, 1, 0.1] cilindro
        }
  copo { (transl [0.0, 2.0, 0.0] ( escala [1, 2, 1] esfera -
        escala [0.6, 1.7, 0.6] esfera)) *
        escala [2, 2, 2] transl [-1, 0, -1] bloco
        }
  base { rotacao [-90, 0, 0] escala [1, 0.2, 1] cilindro
        }
  rotacao [90, 0, 0] (base +haste +transl [0, 0.9, 0] copo)
}
```



```
Palhaço {
  Cubica { MATERIAL[1] POLINOMIAL[3,
    1,0,0,2,1,0,0,1,0,-1,0,0,2,0,0,0,2,0,-4] }
  Esf1 { MATERIAL[2] TRANSL[-1.0,-4.0,-4.0]
    ESFERA }
  Esf2 { MATERIAL[3] TRANSL[-1.0,-4.0,4.0]
    ESFERA }
  Esf3 { MATERIAL[4] TRANSL[-1.0,4.0,-4.0]
    ESFERA }
  Esf4 { MATERIAL[5] TRANSL[-1.0,4.0,4.0]
    ESFERA }
  Cubica + (Esf1 + Esf2 + Esf3 + Esf4)
}
```

Figura 7-1 (cont.) Imagem e definição dos modelos CSG : Cálice e Palhaço.

Baseado em sua definição, o MAO nos fornece o menor número de simplexes,  $N_s$ , a serem avaliados para a construção da aproximação linear. Salim nos mostra que esse número é o mesmo gerado por seu método, já que ele gera no máximo  $2xN_s$  simplexes durante todo o processo de subdivisão.

Podemos dizer que o método proposto dentro desse trabalho também é ótimo, já que no pior caso sua complexidade, no que tange ao número de avaliações de primitivas durante o processo de refinamento, é igual ao método seletivo.

Suponhamos a representação de uma esfera oca, cuja espessura  $e$  de sua parede seja menor que a tolerância estabelecida. Esse sólido portanto possui em sua definição CSG duas primitivas esféricas concêntricas e com raios diferindo de  $e$ . Aplicando-se o método adaptativo proposto, com um posicionamento adequado, podemos garantir que em nenhum momento do processo de refinamento conseguiremos isolar apenas uma das primitivas em um simplexo. Nosso critério adaptativo só pode ser avaliado naqueles simplexes interceptados por apenas uma primitiva. Portanto, nesse caso, seremos obrigados a descer até o último nível de refinamento em todos simplexes interceptados pela superfície, gerando o mesmo número de avaliações que o método seletivo.

No entanto, em casos mais comuns, seremos capazes de reduzir de forma significativa o número de avaliações de primitivas com a aplicação do método adaptativo. A tabela 7-1 mostra os resultados obtidos na construção da poligonização de cinco sólidos exemplo pelos métodos seletivo e adaptativo. Na primeira coluna temos o nome do sólido. Na segunda coluna especificamos o nível de refinamento máximo estipulado para construção da aproximação. No caso do método adaptativo esse valor é obtido em função da tolerância estabelecida para a aproximação (especificada na terceira coluna da tabela) tal como definimos no capítulo 6. Na quarta e quinta colunas apresentamos o número de simplexes classificados durante todo o processo de refinamento. Nas colunas sete e oito apresentamos o número de simplexes que após o processo de refinamento, efetivamente irão produzir alguma aproximação linear em seu interior. As colunas seis e nove apresentam o percentual de simplexes gerados pelo método adaptativo com relação ao seletivo. Podemos notar que em todos os casos apresentados houve uma redução do número de simplexes analisados. Dependendo das características dos modelos essa redução foi mais ou menos acentuada.

Modelo	Nível	Toler.	Simplexos					
			Classificados			Interseptantes		
			Seletivo	Adaptativo	%	Seletivo	Adaptativo	%
3 Cilindros	18	0.5	51552	21356	41,43%	40054	6102	15,23%
	21	0.3	194648	47444	24,37%	154934	14320	9,24%
	25	0.1	1192476	138226	11,59%	949158	42409	4,47%
Palhaço	18	0.5	103604	12804	12,36%	81370	4128	5,07%
	21	0.3	416036	27758	6,67%	326866	8541	2,61%
	25	0.1	2626756	73056	2,78%	2100610	23646	1,13%
Urso	18	0.5	26224	14240	54,30%	20556	4354	21,18%
	21	0.3	103388	33162	32,08%	81754	9865	12,07%
	25	0.1	650548	107106	16,46%	517544	31262	6,04%
Cátice	18	0.5	6508	5038	77,41%	5030	1456	28,95%
	21	0.3	22864	14674	64,18%	18142	4544	25,05%
	25	0.1	125584	46450	36,99%	100080	14024	14,01%
<b>Média</b>	-	-	-	-	<b>31,72%</b>	-	-	<b>12,09%</b>

Tabela 7-1 Distribuição dos simplexos avaliados durante o processo de refinamento seletivo e adaptativo.

Na seção a seguir discutiremos com mais detalhamento cada um dos modelos escolhidos como exemplo bem como as características específicas do método adaptativo

## 7.2.2 Exemplos e Resultados

Dentro dessa seção apresentaremos os resultados obtidos pelo algoritmo de poligonização adaptativa proposto. Na tabela 7-2 apresentamos a distribuição dos simplexos para cada modelo de exemplo em cada nível de refinamento, detalhando melhor aspectos inerentes ao método proposto.

Nas colunas dois e três dão conta, respectivamente, do nível de refinamento máximo e da tolerância estabelecida para o modelo. Como na tabela anterior o nível máximo é sempre obtido em função da tolerância estabelecida. Nas duas colunas subsequentes apresentamos o número de simplexos classificados durante o processo e quantos desses simplexos pertencem a triangulação final, como simplexos onde a superfície do sólido pode passar. Na coluna seis correlacionamos esses dois valores apresentando o percentual de simplexos que pertencem a triangulação final perante o total de simplexos classificados durante o processo. Como nem todos os simplexos classificados como *interseptantes* produzem aproximações lineares, segundo o interpolador definido, temos a sétima coluna apresentando quantos simplexos realmente irão produzir facetas para nossa poligonização. Esse valor é importante pois reflete o quão preciso é o critério de classificação dos simplexos. Podemos notar que na maioria dos casos esse percentual varia em torno de 90%.

Modelo	Nível	Toler.	Simplexos										
			Classif.	Triang. Final	%	Com Aprox.	%	Aprovados				Subdiv. Triang.	%
								Adapt.	%	Últ. Nível	%		
3 Cilindros	18	0.5	21356	6102	28,57%	5774	94,62%	3196	52,38%	2777	45,51%	292	1,37%
	21	0.3	47444	14320	30,18%	13776	96,20%	8020	56,01%	6135	42,84%	290	0,61%
	25	0.1	138226	42409	30,68%	41179	97,10%	27785	65,52%	12575	29,65%	2905	2,10%
Palhaço	18	0.5	12804	4128	32,24%	3808	92,25%	2944	71,32%	848	20,54%	471	3,68%
	21	0.3	27758	8541	30,77%	7805	91,38%	5954	69,71%	1532	17,94%	1644	5,92%
	25	0.1	73056	23646	32,37%	21798	92,18%	16707	70,65%	4383	18,54%	3869	5,30%
Urso	18	0.5	14240	4354	30,58%	3929	90,24%	2270	52,14%	1844	42,35%	450	3,16%
	21	0.3	33162	9865	29,75%	9065	91,89%	5956	60,38%	3339	33,85%	1077	3,25%
	25	0.1	107106	31262	29,19%	29700	95,00%	20483	65,52%	8124	25,99%	4553	4,25%
Cálice	18	0.5	5038	1456	28,90%	1104	75,82%	239	16,41%	1162	79,81%	65	1,29%
	21	0.3	14674	4544	30,97%	4162	91,59%	1571	34,57%	2875	63,27%	191	1,30%
	25	0.1	46450	14024	30,19%	13450	95,91%	8236	58,73%	5278	37,64%	788	1,70%
Media	-	-	-	-	30,37%	-	92,02%	-	56,11%	-	38,16%	-	2,83%

Tabela 7-2 Distribuição dos simplexos gerados a partir do método adaptativo.

As quatro colunas seguintes tratam especificamente dos resultados decorrentes da aplicação do critério adaptativo. As colunas nove e dez indicam quantos simplexos foram aprovados a partir do critério e quanto esse número representa em termos do total de simplexos na triangulação final, respectivamente. Podemos constatar que, em média, quase a metade desses simplexos tiveram abortado seu processo de refinamento devido ao critério adaptativo. Na coluna onze temos o número de simplexos em que ou a avaliação do critério adaptativo não permitiu que o processo de refinamento parasse antes do último nível de refinamento ou devido a impossibilidade de aplicação do critério (tipicamente os simplexos localizados em arestas ou vértices). Na coluna seguinte temos o percentual que esses simplexos representam com relação ao total de simplexos na triangulação final.

Nas duas últimas colunas apresentamos a quantidade absoluta e relativa de simplexos que são subdivididos por conta da manutenção da condição da subdivisão ser uma triangulação. Esse total leva em conta apenas as subdivisões de elementos já aprovados, ou seja, aqueles localizados na *ListaDeAprovados*. Esse número como podemos ver na tabela em média 2,83% dos simplexos são subdivididos mais de uma vez por conta da manutenção da triangulação.

A seguir discutiremos detalhes da distribuição dos simplexos pelos diversos níveis de refinamento, para cada um dos modelos utilizados como exemplo. Os resultados estão agrupados nas tabelas 7-3 a 7-6. Para uma melhor visualização apresentamos também os resultados obtidos nas figuras 7.1 a 7.4. Nessas figuras utilizamos uma graduação de cores para distinguir as facetos geradas em cada nível de refinamento, onde os tons magenta dizem respeito a níveis altos de refinamento, enquanto que polígonos azuis relacionam-se com sempre com o último nível de refinamento do modelo.

A partir da análise mais detalhada dos resultados em cada modelo, podemos levantar várias características importantes do método proposto. A mais forte, no entanto, é que o método é dependente, em certo nível, do modelo a ser aproximado. Modelos

complexos, com muitas arestas e vértices não são capazes de se valer do método adaptativo em sua plenitude. Já que o critério adaptativo pode ser aplicado apenas em simplexos com uma única primitiva, o nível máximo de refinamento tenderá sempre a ser levado ao limite por conta de arestas e vértice. Esse fato é bastante evidente nos modelos 3 Cilindro e Urso. Uma análise das tabelas 7-3 e 7-4, que mostram a distribuição dos simplexos por nível de refinamento para esses dois modelos, deixa claro qual a faixa onde os maiores percentuais se localizam (entre 50% e 30%) estão nos últimos níveis. Analisando as figuras 7.1 e 7.2 podemos notar que os níveis de maior refinamento se localizam nas arestas dos modelos.

O modelo *Palhaço*, ao contrário, consegue se valer de modo bastante eficiente do critério adaptativo. Nesse caso a superfície sobre a qual as quatro esferas estão colocadas é uma cúbica. No entanto até que o “monte” em seu centro comece a se formar temos um comportamento bastante linear, que é levado em conta pelo critério adaptativo. Nesse caso temos em torno de 20% dos simplexos localizados no último nível além de uma concentração entre 10% a 20% em níveis mais baixos, conforme a tabela 7-5 mostra. Na figura 7.3 podemos notar a distribuição dos simplexos no modelo pela variação de suas cores, onde o vermelho representa o menor nível de refinamento e azul o maior.

No modelo *Cálice* podemos notar que a distribuição dos simplexos por nível é bastante elevada para os dois valores iniciais da tolerância. Porém para seu último valor há um retorno para os mesmos níveis dos modelos *Urso* e *3 Cilindros*, como podemos evidenciar na tabela 7-6. Nas regiões da base e haste do cálice um nível de refinamento mínimo é requerido devido à sua pequena espessura. A partir de certo valor de tolerância os simplexos localizados nessas áreas tende a permanecer estacionados, em termos do nível de refinamento, como podemos constatar na figura 7.4. Podemos notar também que os simplexos que geram aproximação em seu último nível de refinamento tendem a se concentrar em todas as regiões de bordo do modelo.

<i>Nível</i>	<i>3 Cilindros</i>		
	<i>0,50</i>	<i>0,30</i>	<i>0,10</i>
25	-	-	30,61%
24	-	-	15,30%
23	-	-	11,67%
22	-	-	5,33%
21	-	43,94%	6,95%
20	-	18,09%	4,91%
19	-	8,41%	6,25%
18	47,46%	8,45%	9,10%
17	18,26%	5,43%	5,93%
16	7,83%	3,30%	2,72%
15	9,28%	5,59%	1,17%
14	8,82%	4,72%	0,06%
13	6,69%	1,37%	-
12	1,57%	0,67%	-
11	0,10%	0,04%	-

*Tabela 7-3 Distribuição dos simplexos pelos diversos níveis de refinamento para o modelo 3 Cilindros.*

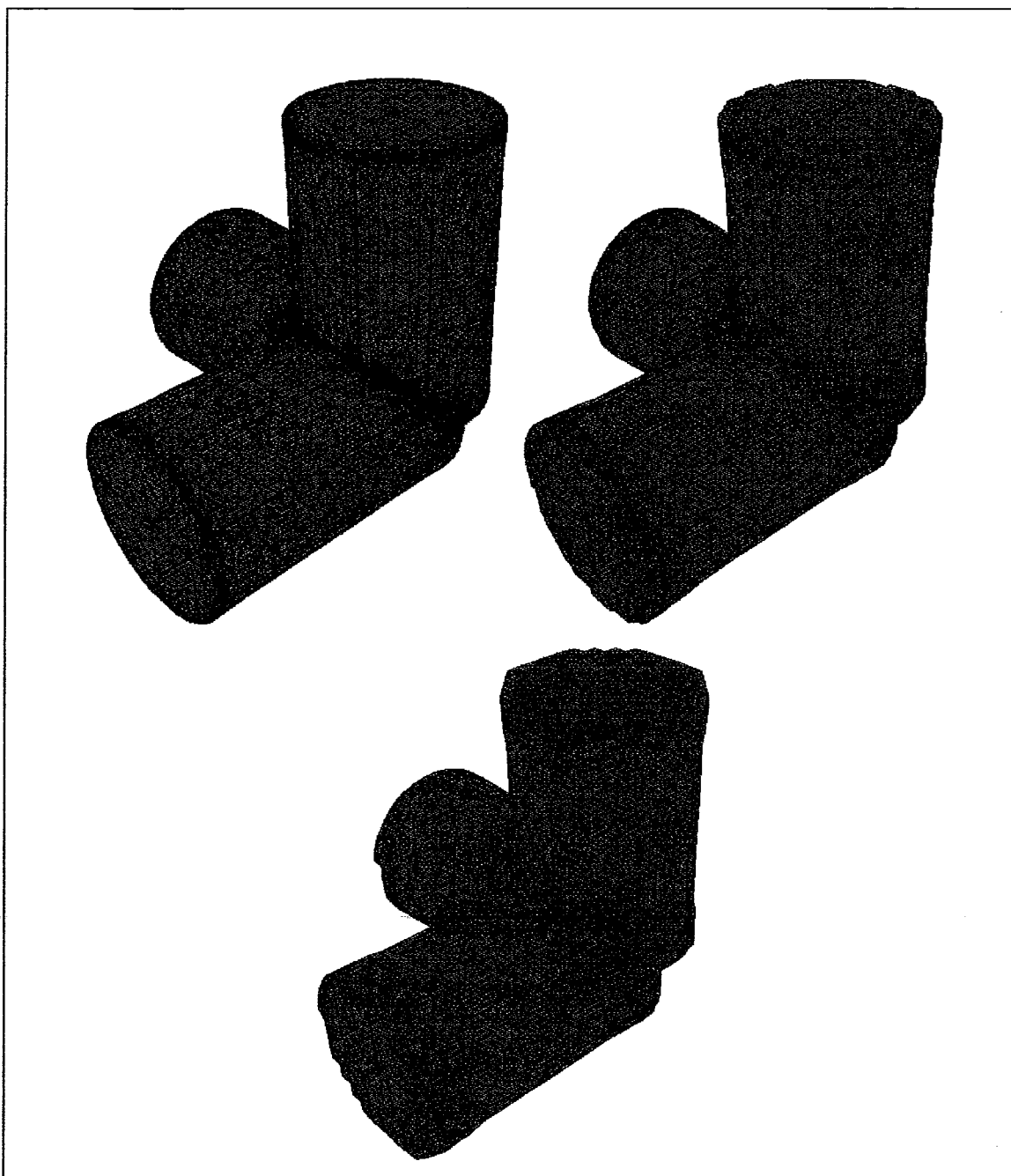


Figura 7.1 Representação da poligonização do modelo 3 Cilindros com tolerância 0,1 (acima à esquerda), 0,3 (acima à direita) e 0,5 (abaixo).

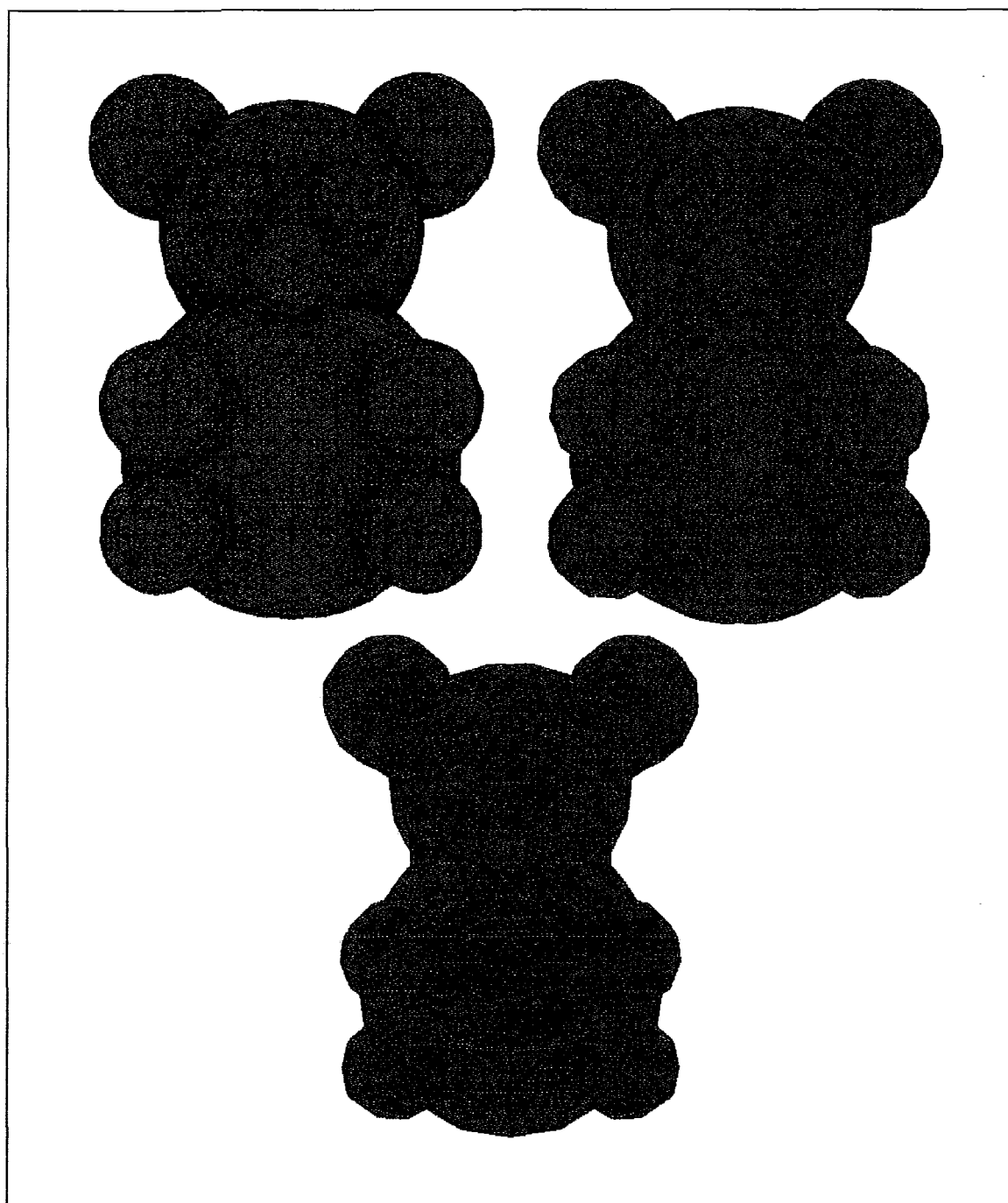
Nível	Urso		
	0,50	0,30	0,10
25	-	-	29,48%
24	-	-	17,00%
23	-	-	9,87%
22	-	-	8,06%
21	-	35,68%	11,32%
20	-	17,25%	9,96%
19	-	13,61%	7,35%
18	47,36%	18,07%	4,88%
17	26,78%	7,65%	1,78%
16	11,39%	4,52%	0,27%
15	9,88%	2,23%	0,04%
14	4,13%	0,85%	-
13	0,46%	0,12%	-

Tabela 7-4 Distribuição dos simplejos pelos diversos níveis de refinamento para o modelo Urso.

Nível	Palhaço		
	0,50	0,30	0,10
25	-	-	19,35%
24	-	-	6,80%
23	-	-	4,70%
22	-	-	2,77%
21	-	19,48%	5,62%
20	-	6,46%	11,60%
19	-	6,73%	14,44%
18	20,93%	14,79%	12,06%
17	12,21%	19,18%	4,45%
16	18,31%	7,97%	7,33%
15	16,67%	6,49%	2,65%
14	5,52%	4,50%	3,82%
13	11,43%	7,59%	4,40%
12	5,62%	3,56%	-
11	4,26%	2,62%	-
10	3,68%	0,63%	-
9	0,78%	-	-
8	0,58%	-	-

Tabela 7-5 Distribuição dos simplejos pelos diversos níveis de refinamento para o modelo Palhaço





*Figura 7.2 Representação da poligonização do Modelo Urso com tolerância 0,1 (acima à esquerda), 0,3 (acima à direita) e 0,5 (abaixo).*

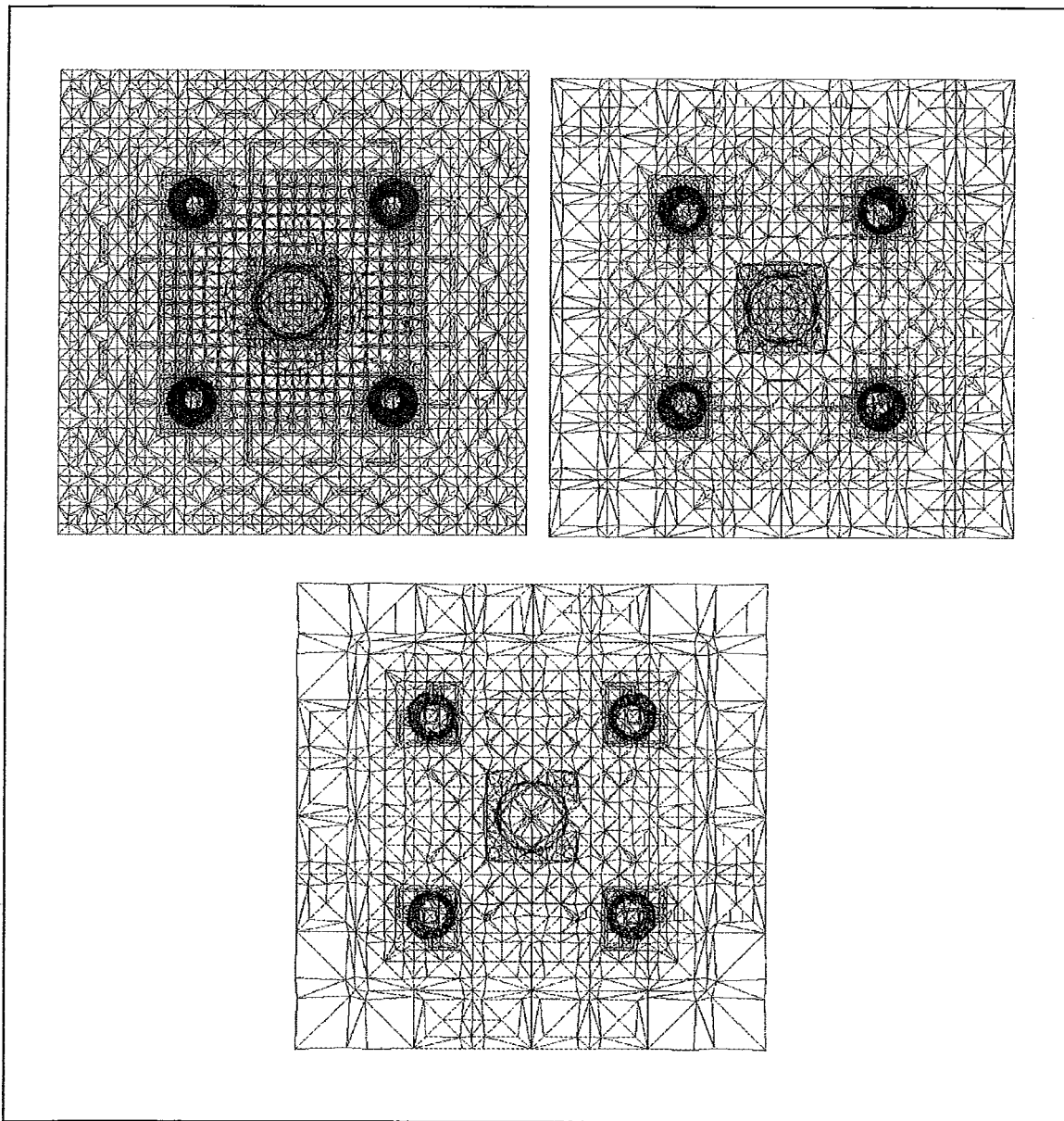
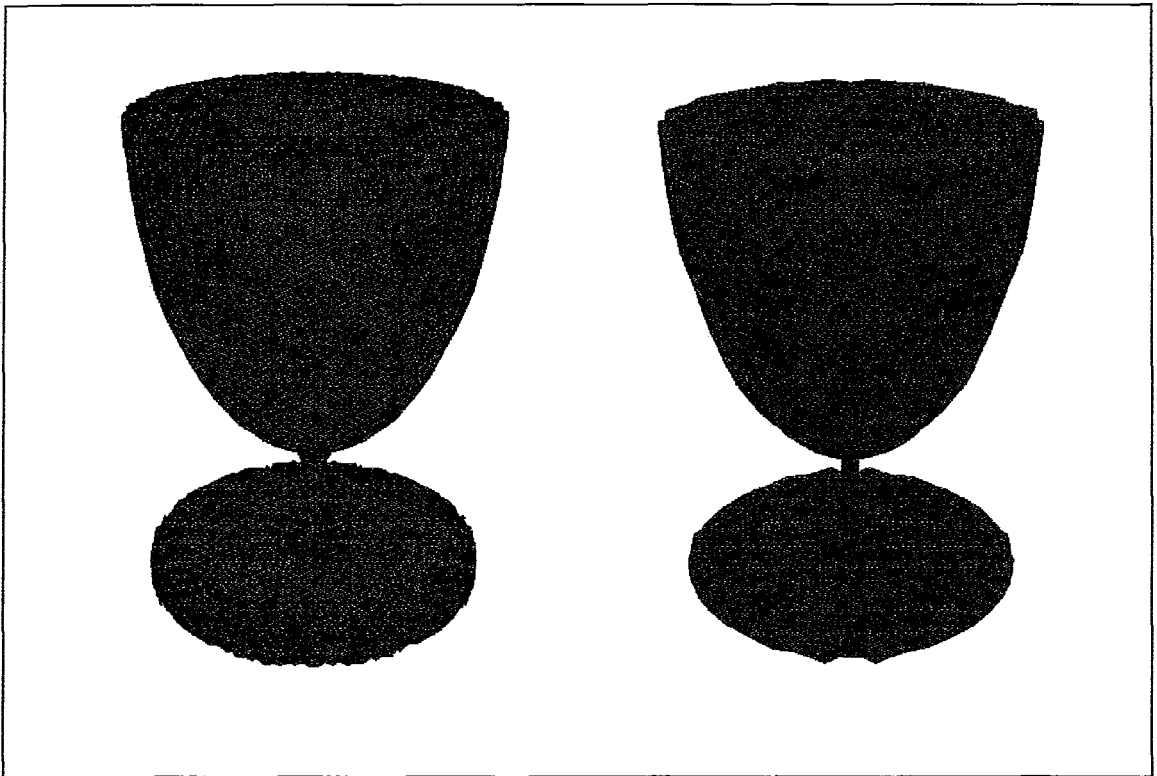


Figura 7.3 Representação da poligonização do modelo Palhaço com tolerância 0,1 (acima à esquerda), 0,3 (acima à direita) e 0,5 (abaixo)

<i>Nível</i>	<i>Calice</i>		
	<i>0,50</i>	<i>0,30</i>	<i>0,10</i>
25	-	-	38,25%
24	-	-	21,76%
23	-	-	12,96%
22	-	-	6,35%
21	-	65,40%	10,27%
20	-	14,39%	6,90%
19	-	11,49%	2,68%
18	86,81%	7,83%	0,83%
17	9,89%	0,88%	-
16	3,30%	-	-

*Tabela 7-6 Distribuição dos simplexos pelos diversos níveis de refinamento para o modelo Cálice.*



*Figura 7.4 Representação da poligonização do modelo Cálice com tolerância 0,1 (esquerda) e 0,3 (direita).*

Outro aspecto importante a ser considerado é o desempenho do algoritmo em sua implementação computacional. A tabela 7-6 mostra os tempos (em segundos) obtidos na geração das poligonizações dos modelos utilizados nos exemplos. Os modelos foram gerados em uma estação de trabalho *Sun Sparcstation 5* com 32 Megabytes de memória. A primeira marca indica quanto tempo de CPU foi dispensado ao programa. Outro tempo marcado é o de utilização de recursos do sistema (duas últimas colunas).

No que diz respeito ao tempo de CPU, os três primeiros modelos tiveram sempre tempos inferiores para a critério adaptativo em comparação com o critério seletivo. O modelo *Cálice* mostra os dois primeiros tempos maiores para o método adaptativo que para o seletivo. No último nível de tolerância, no entanto, essa situação se inverte. De acordo com o que já foi discutido anteriormente esses resultados são justificáveis.

Um resultado que deve ser analisado com cuidado é o tempo médio gasto por simplexo. Computamos esses valores para cada modelo dividindo o tempo de CPU pelo número de simplexos gerados (tabela 7-1). Nesse ponto vemos que o tempo médio para o processamento de cada simplexo no método adaptativo é da ordem do dobro do mesmo tempo gasto no processo seletivo. Esse resultado não é inesperado, já que adicionamos ao método seletivo todo o peso de uma estrutura de dados para suportar a triangulação, assim como um critério de avaliação de erro na aproximação linear. O que podemos concluir, no entanto, é que esse tempo mesmo sendo maior, pode, no final do processo, garantir um menor tempo de geração do modelo, já que nos exemplos conseguimos, em média, uma taxa de redução de avaliações de cerca de 70%.

Ao analisarmos os gastos de tempo por função nas duas implementações (com o utilitário *profile* do Unix), observamos que em ambos os casos os gastos com operações de I/O e alocação dinâmica de memória são os mais pesados. A seguir o cálculo dos coeficientes dos simplexos resultantes de uma subdivisão binária. Destaca-se no método adaptativo o gasto de tempo no processo de subdivisão dos núcleos. É nessa fase que, além da geração dos novos coeficientes, as informações topológicas dos simplexos resultantes são atualizadas. A aplicação do critério de poligonização se mostrou, nesse contexto, relativamente econômica.

Fica claro que o fator crítico de todo o processo se localiza na sua incapacidade de evitar que arestas e vértices do modelo levem a um refinamento excessivo (último nível), o que gera, inevitavelmente, uma queda na diferença entre o número de simplexos gerados em cada método.

Modelo	Nível	Toler.	Tempo de CPU		Tempo CPU p/ Simp.		Tempo do Sistema	
			Seletivo	Adaptativo	Seletivo	Adaptativo	Seletivo	Adaptativo
3 Cilindros	18	0.5	14,1	11,6	0,000274	0,000543	1,8	5,9
	21	0.3	48,7	24,9	0,000250	0,000525	7,6	13,4
	25	0.1	272,3	78,0	0,000228	0,000564	49,9	67,5
Palhaço	18	0.5	24,4	6,4	0,000236	0,000500	3,8	4,3
	21	0.3	95,5	16,7	0,000230	0,000602	15,4	9,3
	25	0.1	632,1	37,0	0,000241	0,000506	110,3	24,4
Urso	18	0.5	6,8	7,4	0,000259	0,000520	0,9	4,0
	21	0.3	23,5	16,5	0,000227	0,000498	3,8	9,1
	25	0.1	140,8	55,0	0,000216	0,000514	25,3	40,1
Cálice	18	0.5	3,4	3,9	0,000522	0,000774	0,2	1,5
	21	0.3	8,0	9,2	0,000350	0,000627	0,9	4,2
	25	0.1	33,2	26,9	0,000264	0,000579	5,0	13,6
Média	-	-	-	-	0,000275	0,000563	-	-

Tabela 7-6 Tempos de execução (em segundos) para geração da aproximação linear utilizando os algoritmos seletivo e adaptativo.

## 7.3 Conclusão

Pelos resultados obtidos, podemos avaliar que o algoritmo aqui apresentado fornece uma solução mais concisa que a obtida pela simples aplicação do critério seletivo.

Nossa proposta também se mostra mais apropriada para executar a tarefa de conversão de um sólido no esquema CSG para o esquema B-Rep. Devido à manutenção da informação topológica da triangulação, somos capazes de construir uma poligonização a partir da qual podemos recuperá-la facilmente quando da montagem de uma representação B-Rep.

## 7.4 Sugestões Para Trabalhos Futuros

Sem dúvida nenhuma, a grande restrição desse trabalho reside no fato de que o critério adaptativo só pode ser aplicado em simplexes que garantidamente possuem apenas uma primitiva em seu interior. Essa restrição se torna muito forte se levarmos em conta que sólidos complexos podem ter um número bastante grande de primitivas em sua composição, o que pode levar o algoritmo a ter que se portar tal qual o método seletivo.

Uma solução para esse problema deve levar em consideração a possibilidade do cálculo efetivo das operações booleanas das primitivas dentro de um simplexo. Nesse caso podemos aplicar o critério adaptativo a cada primitiva dentro do simplexo e, caso

todas possam ser “bem” aproximadas por uma superfície plana, no contexto do simplexo, calcular de modo efetivo sua composição. Atuando dessa forma podemos nos livrar da incômoda e custosa tarefa de levar o refinamento até seu nível limite em regiões com arestas e vértices.

O cálculo dessas operações booleanas dentro dos simplexos não é trivial e fica como a principal sugestão para um trabalho futuro, dentro dessa linha de subdivisão adaptativa aplicada a construção de aproximações poligonais da fronteira de um sólido CSG.

# REFERÊNCIAS

---

- [ALLG80] E. Allgower e K. Georg. "Simplicial and Continuation Methods for Approximating Fixed Points and Solutions to Systems of Equations". SIAM Review - vol. 22 - nº. 1 - 1980.
- [ALLG85] E. Allgower e P. Schmidt. "An Algorithm for Piecewise-Linear Approximation of an Implicitly Defined Manifold". SIAM Journal of Numerical Analysis - vol. 22 - nº. 2 - 1985.
- [ALLG87] E. Allgower e S. Gnutzmann. "An Algorithm for Piecewise-Linear Approximation of Implicitly Defined Two-Dimensional Surfaces". SIAM Journal of Numerical Analysis - vol. 24 - nº. 2 - 1987.
- [BARB94] V. Barbalho. "Geração Adaptativa de Malhas de Elementos Finitos de Sólidos CSG". Dissertação de Mestrado (COPPE/UFRJ) - 1994.
- [BLOO88] J. Bloomenthal. "Polygonization of Implicit Surfaces". Computer Aided Geometric Design - vol. 5 - 1988.
- [BRIT94] J. Brito, S. Nascimento e S. Meerbaum. "Morphing de Imagens". Anais do SIBGRAP'94 - VII Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens - 1994.
- [BRUN90] P. Brunet e I. Navazo. "Solid Representation and Operation Using Extended Octrees". ACM Trans. Graph. - vol. 9 - nº. 2 - 1990.
- [CAST90] A. Castelo, S. Freitas e G. Tavares. "PL Approximation to Manifolds and Its Application to Implicit ODEs". Lect. Appl. Math. (Amer. Math. Soc.) - vol. 26 - 1990.
- [ROMA92] P. Roma. "Criação e Manutenção de Subdivisões do Espaço". Dissertação de Doutorado (PUC / RJ) - 1992.



- [ESPE90] C. Esperança. "Técnicas para Visualização Direta de Sólidos CSG". Dissertação de Mestrado (COPPE/UFRJ) - 1990.
- [FARI86] G. Farin. "Triangular Bernstein-Bézier Patches". CAGD - vol. 3 - nº. 2 - 1986.
- [FARI90] G. Farin. "Curves and Surfaces for Computer Aided Geometric Design". Academic Press - 1988.
- [HUI94] K. Hui. "Solid Modeling with Sweep-CSG representation". Proceedings of CSG 94 Conference - Set-Theoretic Solid Modelling : Techniques and Applications.
- [GUIB85] L. Guibas e J. Stolfi. "Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams". ACM Trans. Graph. - vol. 4 - nº. 2 - 1985.
- [HOFF89] C. Hoffman. "Geometric and Solid Modelling - An Introduction". Morgan Kauffman Publishers, Inc. - 1989.
- [HOPP93] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald e W. Stuezle. "Mesh Optimization". Computer Graphics Proceedings - 1993.
- [HOTT92] V. Hottum. "Uma Estratégia para Construção de Representações Brep de Sólidos CSG". Dissertação de Mestrado (COPPE/UFRJ) - 1992.
- [JACK80] C. Jackins e S. Tanimoto. "Octrees and their Use in Representing Three-dimensional Objects". Computer Graphics and Image Processing - vol. 14 - nº 3 - 1980.
- [MÄNT90] M. Mäntylä. "An Introduction to Solid Modeling". Computer Science Press - 1990.
- [GOME89] J. Gomes e G. Tavares. "Métodos Simpliciais em Computação Gráfica". 17º Colóquio Brasileiro de Matemática - 1989.

- [NEVE88] J. Neves. "Sistema Interativo para Mapeamento Baseado na Triangulação de Pontos do Plano". Dissertação de Mestrado (COPPE/UFRJ) - 1988.
- [OLIV93] A. Oliveira e A. Cruz. "Transformação de Curvas Poligonais : Um Algoritmo Linear a partir de uma Triangulação". Anais do SIBGRAPI'93 - VI Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens - 1993.
- [PERS93] R. Persiano, J. Comba e V. Barbalho. "An Adaptative Triangulation Refinement Scheme and Construction". Anais do SIBGRAPI'93 - VI Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens - 1993.
- [PERS94] R. Persiano e A. Apolinário. "Boundary Evaluation of CSG Models by Adaptative Triangulation". Proceedings of CSG 94 Conference - Set-Theoretic Solid Modelling : Techniques and Applications.
- [PREP85] F. Preparata e M. Shamos. "Computational Geometry". Spring Verlag - 1985.
- [REQU80] A. Requicha. "Representation for Rigid Solids : Theory, Methods and Systems". ACM Computing Surveys - vol. 12 - n°. 4 - 1980.
- [REQU92] A. Requicha e J. Rossignac. "Solid Modeling and Beyond". IEEE Computer Graphics & Applications - Setembro - 1992.
- [RIVA92] M. Rivara. "Local Modification of Meshes for Adaptative and/or Multigrid Finite-Element Methods". Communications in Applied Numerical Methods - vol. 8 - 1992.
- [ROSS91] J. Rossignac e A. Requicha. "Constructive Non-Regularized Geometry". CAD - vol. 23 - n°. 1 - 1991.
- [SALI91] M. Salim, L. Bueno e R. Persiano. "Aproximação Linear por Partes de Variedades Implicitamente Definidas usando Poda do Espaço". Anais do SIBGRAPI'91 - IV Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens - 1991.

- [SALI92] M. Salim. "Aproximação Linear por Partes de Sólidos CSG Através de Subdivisão Simplicial Adaptativa". Dissertação de Mestrado (COPPE/UFRJ) - 1992.
- [SEID89] H. Seidel. "A General Subdivision Theorem for Bézier Triangles". In : "Mathematical Methods in Computer Aided Geometric Design", pp 573-581 - Academic Press - 1989.
- [SHAP91a] V. Shapiro e D. Vossler. "Efficient CSG Representation of Two-Dimensional Solids". Transaction of the ASME - vol. 113 - Setembro - 1991.
- [SHAP91b] V. Shapiro e D. Vossler. "Construction and Optimization of CSG Representations". CAD - vol. 13 - n°. 1 - 1991.
- [TILO84] R. Tilove. "A Null-Object Detection Algorithm for Constructive Solid Geometry". Communications of the ACM - vol. 27 - n°. 7 - 1984.
- [VELH90] L. Velho. "Adaptive Polygonization of Implicit Surfaces Using Simplicial Decomposition and Boundary Constrains". Proceedings of Eurographics - 1990.
- [WARR90] J. Warren e M. Hall. "Adaptative Polygonalization of Implicitly Defined Surfaces". IEEE Computer Graphics & Applications - vol.27 - n°. 7 - 1984.