

OK

CONSULTA
FD-3651
Ed.rev.

2004

SÃO PAULO

Dissertação apresentada à
Escola Politécnica da
Universidade de São Paulo
para obtenção do título de
Mestre em Engenharia.

**Representação de Curvas e Superfícies em Modeladores de
Sólido B-Rep**

WANG CONGLI

WANG CONGLI

Representação de Curvas e Superfícies em Modeladores de Sólido B-Rep

Dissertação apresentada à
Escola Politécnica da
Universidade de São Paulo
para obtenção do título de
Mestre em Engenharia.

Orientador: Prof. Dr. Marcos de Sales de Guerra Tsuzuki

SÃO PAULO

2004

Página 1

Agradecimentos:
Agradeço à CAPES pelo suporte financeiro e ao Prof. Dr. Marcos de Sales Guerra Tsuzuki pela orientação.

CONTÉUDO

AGRADECIMENTOS

CONTÉUDO

5	LISTA DE FIGURAS	5
9	LISTA DE TABELAS	9
10	LISTA DE SÍMBOLOS	10
11	RESUMO	11
12	ABSTRACT	12
13	1 INTRODUÇÃO	13
15	1.1 Técnicas em Modelagem de Sólidos	15
15	1.1.1 Representação CSG	15
16	1.1.2 Representação B-Rep	16
17	1.2 Definição Automática de Superfícies	17
18	1.3 Necessidade da Associação Entre Curvas e Arestas	18
20	2 CURVAS E SUPERFÍCIES	20
20	2.1 Modelagem de Curvas	20
20	2.1.1 Curvas de Hermite	20
20	2.1.2 Curvas de Bézier	20
21	2.1.3 Conversão entre Curvas de Bézier e Curvas de Hermite	21
23	2.1.4 Divisão de Curvas de Bézier	23
25	2.1.5 Conexão Entre Curvas de Bézier	25
28	2.2 Modelagem de Superfícies	28
28	2.2.1 Superfície de Coons	28
30	2.2.2 Superfície de Bézier	30
33	2.2.3 Conexão Entre Superfícies de Bézier	33
35	2.2.4 Gregory Patch	35
37	2.3 Superfícies definidas automaticamente	37
37	2.3.1 Face Triangular	37
39	2.3.2 Face com Cinco Lados	39
40	3 MODELAGEM DE SÓLIDOS	40

3.1	Estrutura de Dados	40
3.2	Operadores de Euler	44
3.3	OPERADORES LOCAIS	49
3.3.1	Cria arco de circunferência.....	50
3.3.2	Extrusão translacional	50
3.3.3	Extrusão rotacional	51
4	SINCRONIZAÇÃO DA MODELAGEM DE SÓLIDOS E MODELAGEM GEOMÉTRICA	52
4.1	Propostas de Turner para Sincronizar Faces e Superfícies.....	52
4.2	Proposta de Ueda para Sincronizar Arestas e Curvas	54
5	SINCRONIZANDO MODELAGEM DE SÓLIDOS E MODELAGEM GEOMÉTRICA	59
5.1	Estrutura de Dados para representar Curvas	61
5.2	Operadores de Euler para Manipular Curvas.....	63
5.3	Curvas Como Atributos de Meia-arestas	65
5.4	Estrutura de Dados para Representar Superfícies.....	66
5.5	Operadores de Euler para Manipulação de Superfícies	69
5.5.1	Operador de Euler	69
5.5.2	Operadores de Baixo Nível na Estrutura Superfície.....	71
5.5.3	Operadores de Baixo Nível na Estrutura Patch	72
5.5.4	Operadores de Alto Nível.....	72
5.6	Considerações Finais.....	74
6	CONCLUSÃO	76
	REFERÊNCIA	78

LISTA DE FIGURAS

15	Figura 1. Representação CSG de um sólido.
16	Figura 2. Componentes básicos de um modelo por fronteiro.
17	Figura 3. Suavização de sólidos.
19	Figura 4. Superfícies em posição T.
21	Figura 5. Curva de Hermite.
21	Figura 6. Exemplos de curva de Bézier.
22	Figura 7. Bases de Bernstein para $n=3$.
26	Figura 8. Divisão de curvas de Bézier.
28	Figura 9. Conexão entre duas curvas de Bézier de grau 3.
30	Figura 10. Exemplo de superfície de Coons.
32	Figura 11. Exemplo de superfície de Bézier.
33	Figura 12. Conversão entre superfície de Bézier e superfície de Coons.
33	Figura 13. Conexão entre duas superfícies de Bézier de grau 3.
34	Figura 14. Exemplo de conexão entre quatro superfícies de Bézier.
35	Figura 15. Exemplo de Gregory Patch.
36	Figura 16. Exemplo de carroceria modelada utilizando superfícies de Gregory.
37	Figura 17. Um sólido que possui uma face triangular e o sólido com uma face de cinco lados.
38	Figura 18. Caso de face com três lados.
39	Figura 19. Caso de face com cinco lados.

40	Figura 20. Estrutura <i>winged-edge</i> .
41	Figura 21. Estrutura <i>meia-aresta</i> .
42	Figura 22. Extensão para faces com múltiplos contornos.
42	Figura 23. Sólido com duas regiões.
43	Figura 24. Elementos topológicos de um modelo sólido.
44	Figura 25. Hierarquia de elemento da estrutura unificada.
45	Figura 26. Sólidos verificando a equação de Euler-Poincaré.
47	Figura 27. Representação gráfica dos operadores de Euler.
48	Figura 28. Sequência de operadores de Euler para criação de um cubo com furo passante.
49	Figura 29. Um sólido auto-interceptante.
50	Figura 30. Sequência de operadores de Euler que implementam a criação de um arco.
50	
50	Figura 31. Finalização da criação de uma circunferência.
51	Figura 32. Sólidos obtidos por extrusão translacional.
51	Figura 33. Sólidos gerados por extrusão rotacional.
52	Figura 34. Um cilindro aproximado por arestas.
52	Figura 35. Cilindro depois da aplicação do operador KEF.
53	Figura 36. Cilindro depois da aplicação do operador KEV.
53	Figura 37. Cilindro final após o enxugamento.
54	Figure 38. Mecanismo que mostra como aumentar o número de extremidades aproximadas.
55	Figura 39. Operador de alta nível de Develop Curve.

56	Figura 40. Remove as arestas de aproximação de curva.
57	Figura 41. Uma curva projectada.
59	Figura 42. Estrutura meia-aresta.
60	Figura 43. Modelo curva e modelo com superfícies.
61	Figura 44. Estrutura Hierárquica para representar curvas.
62	Figura 45. Processo que cria arestas de aproximação de uma curva.
62	Figura 46. Arestas com ponteiros para sua curva.
63	Figura 47. Operações para converter uma aresta reta em uma curva e a sua inversa.
64	Figura 48. Operação para trasladar o n-ésimo ponto de controle de uma curva por um vetor e a sua inversa.
65	Figura 49. Reajustamento quando pontos de controle sobre os vértices modificados.
65	Figura 50. Associação de curvas e meia-arestas, exemplo de aplicação do operador MEF.
67	Figura 51. Hierarquia da Estrutura de Superfícies.
68	Figura 52. Exemplo de superfícies desenvolvidas.
69	Figura 53. Uma face triangular associadas pelo três "patch".
70	Figure 54. Exemplo de superfície de Bézier.
70	Figura 55. A forma da superfície mudou depois os pontos de controle foram modificados.
73	Figura 56. Operador de alto nível "copyFace".
73	Figura 57. Criação da aproximação poliedral.
74,75	Figura 58. Exemplos em USPDesigner.

LISTA DE TABELAS

Tabela 1. Nomenclatura dos operadores de Euler.

46

Tabela 2. Operadores construtivos e destrutivos.

48

LISTA DE SÍMBOLOS

$B_n^i(t)$	Bases de Bernstein
$\tilde{Q}(t)$	Pontos sobre uma curva paramétrica
P_i	Pontos de controle de uma curva
i	Número inteiro
$S(u, v)$	Pontos sobre uma superficie paramétrica
n	Parâmetro que está na faixa $0 \leq n \leq 1$
w	Parâmetro que está na faixa $0 \leq w \leq 1$
f	Número inteiro
P^I	Ponto de controle de uma curva
P^{II}	Ponto de controle de uma curva
P^{III}	Ponto de controle de uma curva
P^{IV}	Ponto de controle de uma curva
$\tilde{Q}^u(u, v)$	Vetor tangente na direção v sobre o ponto (u, v) de uma superficie
$\tilde{Q}^n(u, v)$	Vetor tangente na direção u sobre o ponto (u, v) de uma superficie
C^1	Continuidade de derivada

RESUMO

Neste trabalho uma nova estrutura de dados é definida baseando-se em uma representação não-manifold para suportar a representação de curvas e superfícies em modelos sólidos B-Rep. Esta nova estrutura de dados suportará a sincronização entre a geometria do modelo geométrico e um modelo poliedral aproximado. Um modelo poliedral aproximado é obtido permitindo que sejam realizados cálculos de propriedades de massa utilizando vários algoritmos existentes na literatura. Sendo possível também reduzir o poliedro aproximador para um sólido com um número mínimo de elementos quando necessário. O elemento face, em modeladores de sólido comerciais, possuem duas funções: representar o contorno do sólido e representar a geometria da forma do contorno. Nesta nova estrutura, estaremos separando estas duas funções, um elemento topológico representará a forma geométrica como curvas e superfícies de forma livre. Este mecanismo facilitará a representação de faces irregulares (faces com três, cinco ou mais lados). Desde que não é comum para um sistema de CAD comercial suportar a representação direta de faces irregulares, geralmente estas faces são divididas em um número de faces com quatro lados. Este fato é uma inconsistência entre topologia e geometria, pois o número de lados em uma face é definido pela topologia e a geometria geralmente "força" a face a ter apenas quatro lados. Neste trabalho, nós propomos duas estruturas, a estrutura de dados principal contém o modelo sólido original, e a estrutura de dados auxiliar representa uma aproximação para o modelo sólido. Neste mecanismo, um modelo automático é definido para determinar a forma geométrica das faces irregulares. Assim, separaremos as intenções do usuário de modificações automáticas feitas pelo sistema. Para sincronizar ambas as estruturas de dados, definiremos alguns algoritmos para manter a consistência da estrutura de dados.

ABSTRACT

In this article a new data structure is defined based on non-manifold representation to support the representation of curves and surfaces in B-Rep solid models. This new data structure will support the synchronization between the geometry of the Geometric Model and an approximated polyhedral. An approximated polyhedral is obtained allowing the calculation of mass properties using several existent algorithms in the literature. And it is possible to reduce the approximated polyhedral for a minimum representation when necessary. The face, in commercial solid modelers, has two functions: represent the boundary of the solid and represent the geometrical shape of the contour. In this new structure, we will separate these two functions, one topological element will represent the boundary of the solid and another topological element will represent the geometrical shape as free form curves and surfaces. This mechanism facilitates the representation of irregular faces (faces with three, five or more sides). Since it is not common for a commercial CAD system to support the direct representation of a three-side surface, usually, those faces are subdivided in a number of four side faces. That fact is an inconsistency between topology and geometry, as the number of sides in a face is defined by topology and geometry usually "forces" a face to have four sides. In this article, we will propose two structures, the principal data structure contains the original solid model, and the auxiliary data structure represents an approximation of the solid model. In this mechanism, an automatic mode is defined that can determine the forms of the irregular faces separated to the mode that surfaces can be modified by users. To synchronize both data structures, we defined some algorithms maintaining the consistence of the data structure.

1 Introdução

Modelagem de sólidos e modelagem geométrica foram desenvolvidas separadamente como apontado por Farin [7]. É muito comum associar sólidos poliedrais para modelagem de sólidos, e curvas e superfícies para modelagem geométrica. Modelos poliedrais têm algumas desvantagens. Primeiro, a característica topológica significativa do modelo é perdida – por exemplo, um buraco cilíndrico simples é representado por muitas faces e não por uma expressão matemática. Segundo, a geometria do modelo é aproximada – o nível de precisão geométrica depende do número de faces utilizado para aproximar cada uma das faces curvas.

As necessidades tecnológicas atuais requerem que ambas as tecnologias estejam presentes nos modeladores de sólidos. Entretanto, na literatura temos poucas propostas para integrar ambas as tecnologias. Toriya e Chiyokura [2, 3, 14] descrevem as propostas utilizadas no modelador de sólidos desenvolvido na Ricoh. Eles representam curvas como atributos de arestas e superfícies como atributos de faces. Entretanto, nada foi apresentado para integrar de forma síncrona as duas tecnologias. Mäntylä [11] descreve as propostas utilizadas no modelador de sólidos GWB desenvolvido na Universidade de Tecnologia de Helsinki. As propostas apresentadas são muito semelhantes às propostas apresentadas por Toriya e Chiyokura [14]. Novamente, nada foi apresentado para integrar de forma síncrona as duas tecnologias.

Turner [16], foi mais além, as curvas são representadas como atributos de arestas e as superfícies são representadas como atributos de faces. A sua proposta incluía um mecanismo para subdividir a superfície em um conjunto de faces planas. O mecanismo para subdividir a superfície em faces poligonais se utiliza de um conjunto de operadores básicos conhecidos como Operadores de Euler. Segundo Turner, este mecanismo é muito importante para gerar faces poligonais que aproximam as superfícies geométricas de um sólido. É possível considerar que temos dois sólidos: um sólido com as superfícies geométricas exatas, e outro sólido contendo faces poligonais que aproximam as

Neste trabalho, é apresentada uma proposta que permite a convivência dos dois mecanismos aproximadores: aproximação de curvas por linhas poligonais proposto por Ueda [17], e aproximação de superfícies por faces poligonais proposto por Turner [16].

Entretanto, Ueda [17] descobriu que o seu mecanismo para criar linhas poligonais que aproximam uma curva não poderia ser utilizado em conjunto com o mecanismo proposto por Turner [16] para subdividir uma superfície em faces poligonais aproximadoras. O principal efeito colateral é que as faces poligonais aproximadoras também definem uma aproximação para as curvas de contorno que delimitam a superfície que está sendo aproximada. Sendo que outro fator complicador é a manutenção dos atributos, indicando quais arestas aproximam quais curvas e quais faces aproximam quais superfícies.

Ueda [17] estudou a proposta apresentada por Turner e propôs um mecanismo semelhante para determinar linhas poligonais que aproximam uma curva. Novamente, as curvas são representadas como atributos de arestas e as superfícies são representadas como atributos de faces.

Para alguns propósitos, a representação poliedral é necessária, por exemplo, para calcular propriedades de massa e visualizar os sólidos representados. Para se calcular as propriedades de massa, os modeladores B-Rep se utilizam de um algoritmo que varre todas as faces poligonais do sólido [15]. Este mesmo algoritmo não poderia ser utilizado diretamente sobre uma representação de curvas e superfícies. A representação de curvas e superfícies é necessária para ter acesso à representação exata da forma. Esta informação é útil para calcular o caminho de ferramentas de usinagem, dentre outras várias aplicações.

superfícies geométricas do sólido anterior. Conforme veremos as duas representações são necessárias.

1.1 Técnicas em Modelagem de Sólidos

Sendo que as curvas e superfícies podem ser aproximadas por arestas e faces poliedrais que são apresentadas no modelagem de sólidos, a seguir discutiremos as duas formas de representação para modelagem de sólidos mais significativas: CSG e B-Rep.

1.1.1 Representação CSG

Na representação CSG, um sólido é definido pela combinação de sólidos simples.

Estes sólidos simples são chamados primitivos. Um sólido é representado segundo uma árvore binária onde as suas folhas são os primitivos e os nós internos são operadores booleanos que relacionam os dois nós inferiores. A construção de modelos CSG combinando primitivos por operadores booleanos é relativamente fácil, pois é uma tarefa intuitiva. A Figura 1 (a) mostra um exemplo de sólido com representação CSG. O sólido E é definido pela combinação dos sólidos A e B, por meio de uma operação de intersecção. O sólido F é definido pela diferença entre os sólidos C e D. Finalmente, o sólido G é definido pela união dos sólidos E e F. Este processo sequencial de modelagem é

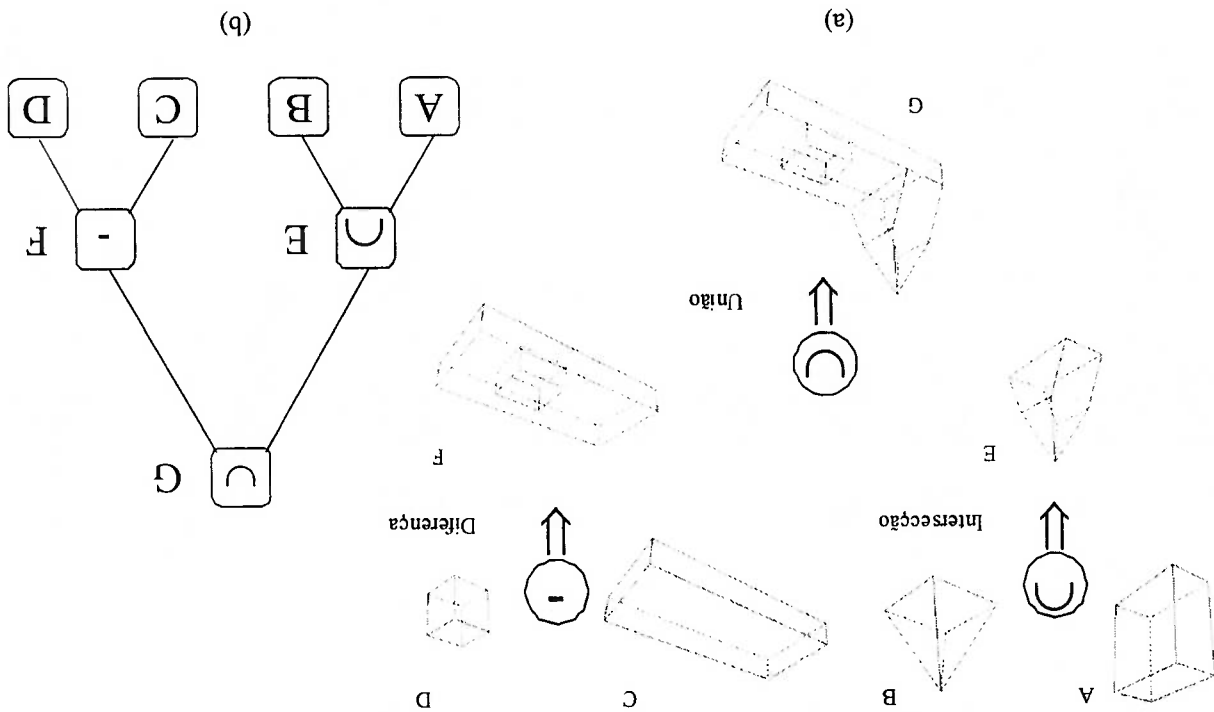


Figura 1. Representação CSG de um sólido.

1.1.2 Representação B-Rep

representado por uma árvore binária denominada árvore CSG, ilustrada na Figura 1 (b).

A representação B-Rep armazena detalhes de como as faces, arestas e vértices de um sólido se unem para representar um sólido. Um sólido representado em modelo B-Rep deve possuir, por exemplo, a capacidade de descrever como cada face está conectada as suas faces adjacentes, de modo a formar um volume totalmente fechado no espaço. Nos modelos CSG, a informação de adjacência entre faces pode ser obtida por técnicas numéricas que analisam as proximidades geométricas de seus componentes; contudo, estas técnicas, geralmente, envolvem um custo computacional considerável além de apresentarem problemas de precisão numérica. Nos modelos B-Rep estas informações estão presentes de forma explícita na estrutura de dados.

Os três tipos de elementos básicos de um sólido (face, aresta e vértice) e a informação geométrica relacionada aos mesmos formam os constituintes básicos dos modelos B-Rep. Junto com as informações geométricas, como equações do plano das faces e coordenadas de vértice, um modelo B-Rep deve também representar a relação entre as faces, arestas e vértices. Normalmente as informações geométricas dos elementos de um sólido são denominadas por geometria do modelo B-Rep, enquanto as informações sobre o compartilhamento dos elementos básicos são denominadas

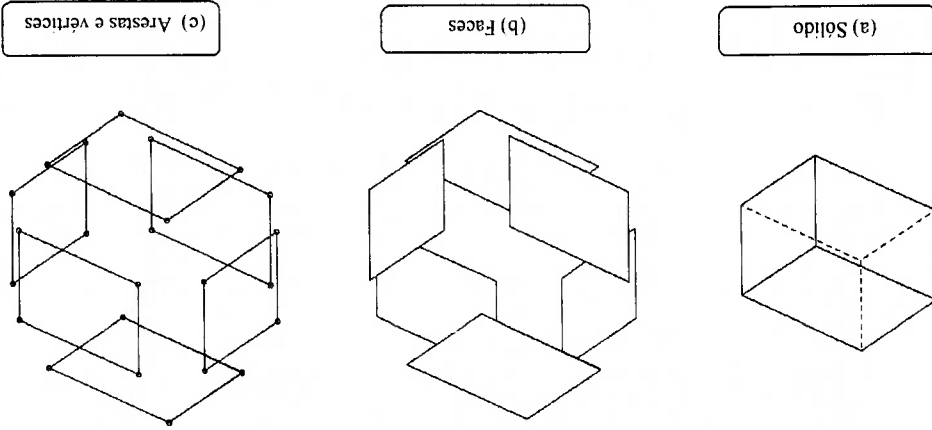
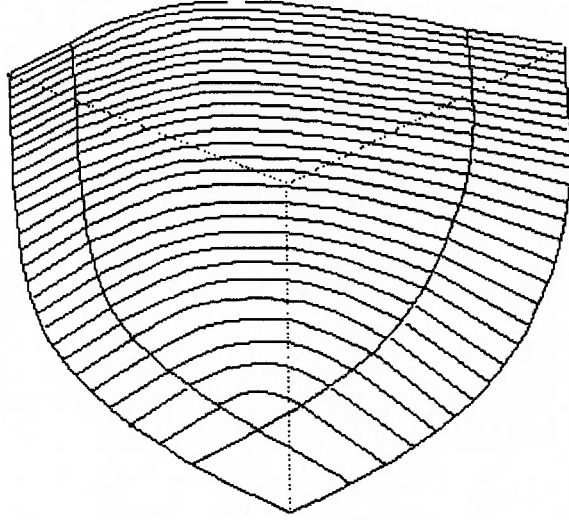


Figura 2. Componentes básicos de um modelo por frenteira.

Figura 3. Suavização de sólidos.



Quando utilizamos um software de CAD/CAM, imaginamos que temos o controle completo sobre a geometria da forma do sólido que está sendo editado. Entretanto, isto nem sempre é verdade. Para que expressões polinomiais sejam associadas a uma superfície é necessário que a superfície tenha quatro lados. Resta a seguinte pergunta: o que acontece com as superfícies que não possuem quatro lados? Existem propostas na literatura [2, 4, 5, 14] para subdividir uma superfície de 3, 5 e 6 lados em 3, 5 e 6

1.2 Definição Automática de Superfícies

No nosso trabalho, utilizamos modelos B-Rep para representar modelos poliedrais. Desde que aproximação das curvas e superfícies são principalmente modelos poliedrais, B-Rep é útil em representação dos modelos.

(c).

A Figura 2 ilustra os componentes básicos de um modelo B-Rep. Na figura (a), a superfície é dividida em conjuntos fechados de faces; cada face é representada através de um polígono de contorno (b), que pode ser representado em termos de arestas e vértices topológicas criam um vigaamento no qual as informações geométricas são posicionadas".

informalmente por topologia do modelo. Pode-se dizer que a topologia funciona como uma goma onde as informações geométricas são aglutinadas; ou então que "as informações

Este trabalho está estruturado da seguinte maneira. O Capítulo 2 apresenta uma introdução sobre curvas e superfícies. Bem como sobre o algoritmo para definir superfícies automaticamente proposto por Chiyokura e Kimura [4].

Neste trabalho, também propomos que a curva seja associada às meia-arestas e não às arestas. Para compreendermos melhor esta nova proposta é necessário que aprofundemos nossos conceitos sobre modelagem de sólidos e modelagem geométrica. Apesar de ser um fato simples, considerar a curva como um atributo de meia-aresta facilitará a manipulação das curvas e superfícies, principalmente quando existir arestas que terminam em T (vide Figura 4). Estas casos aparecem tipicamente na aplicação dos algoritmos que definem automaticamente a geometria de superfícies (por exemplo, algoritmo proposto por Chiyokura e Kimura [4]).

1.3 Necessidade da Associação Entre Curvas e Arestas

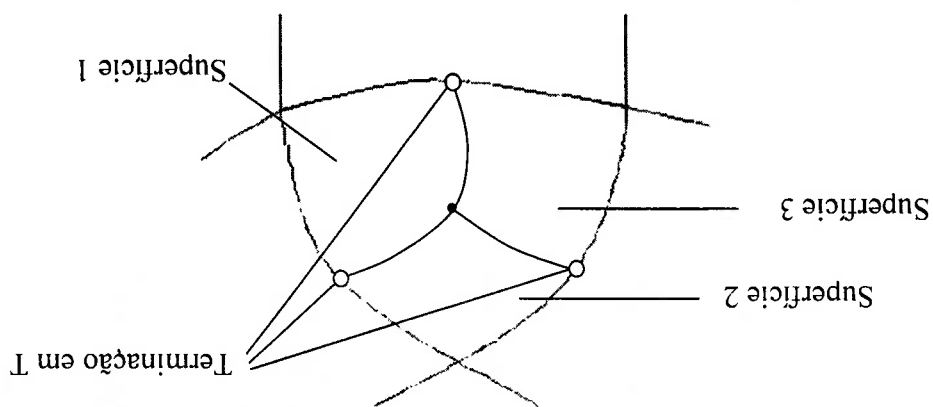
Neste trabalho, é apresentada uma representação que facilita a identificação, inserção e remoção de superfícies geradas automaticamente com a finalidade de associar a geometria à superfície. Na proposta de Turner [16], temos superfícies definidas pelo usuário, superfícies definidas automaticamente e faces poligonais aproximadas em um único nível. Entretanto, neste trabalho associamos um nível hierárquico a cada um destes três tipos de superfícies. As superfícies definidas pelo usuário estão no nível mais elevado, pois todas as informações são determinadas a partir dele. No nível intermediário temos as superfícies criadas automaticamente pelo sistema, para que a geometria possa ser associada. E finalmente, no nível mais baixo temos as faces poligonais que aproximam as superfícies que possuem geometria associada.

superfícies de quatro lados, respectivamente. Conforme veremos, estas superfícies que não possuem quatro lados são criadas com muita frequência nos modelos sólidos a partir de variadas operações geométricas. Sendo um caso típico a operação para suavizar arestas (vide Figura 3).

Com todos estes conceitos presentes, apresentamos no Capítulo 5 a nossa proposta para realizar a integração entre modelagem de sólidos e modelagem geométrica. Os métodos propostos por Turner [16] e Ueda [17] podem coexistir. Também identificamos e manipulamos com maior facilidade, quando comparado aos métodos anteriores, as superfícies definidas pelo usuário, as superfícies que são definidas automaticamente e as faces poligonais de aproximação. No Capítulo 6 apresentamos algumas conclusões.

O Capítulo 3 apresenta conceitos básicos sobre modelagem de sólidos. O Capítulo 4 apresenta em maior profundidade os métodos propostos por Turner [16] e Ueda [17] para realizar a integração entre modelagem de sólidos e modelagem geométrica.

Figura 4 Superfícies em posição T.



2 Curvas e Superfícies

Neste capítulo, apresentaremos uma breve introdução sobre modelagem geométrica. Esta introdução versa sobre a representação de curvas e superfícies. Apresentaremos também o algoritmo de subdivisão de superfícies com 3 e 5 lados proposto por Chiyokura e Kimura [4].

2.1 Modelagem de Curvas

Nesta sessão apresentaremos as curvas de Hermite e Bézier. Apresentaremos também o algoritmo para subdivisão de curvas de Bézier. De modo simplificado podemos dizer que a curva de Hermite é definida pelo seu contorno (ponto inicial, final e vetor tangente sobre estes pontos) e a curva de Bézier é definida pelo seu interior. Entretanto, é possível converter uma representação para outra, permitindo o uso de uma ou outra característica que melhor se adequa a solução de um dado problema. Apresentaremos também as condições que garantem continuidade de derivada entre curvas de Bézier.

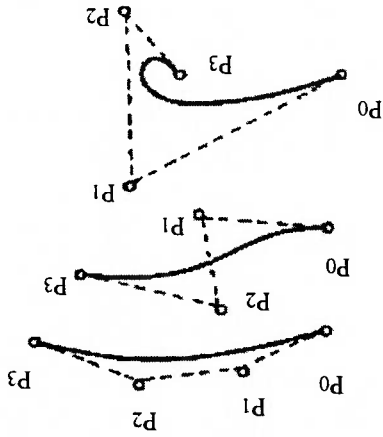
2.1.1 Curvas de Hermite

A curva de Hermite pode ser escrita segundo a equação matemática abaixo:

$$\tilde{Q}(n) = \begin{bmatrix} n^3 & n^2 & n & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} P_i \\ P_{i+1} \\ D'_i \\ D'_{i+1} \end{bmatrix} \quad (1)$$

Onde P_i e P_{i+1} são os pontos que definem os extremos da curva e D'_i e D'_{i+1} são os vetores tangentes sobre os extremos da curva. O parâmetro n está na faixa $0 \leq n \leq 1$. A Figura 5 exibe o exemplo de uma curva de Hermite.

Figura 6. Exemplos de curva de Bézier.



onde $B''_i(n)$ são bases de Bernstein, dadas por

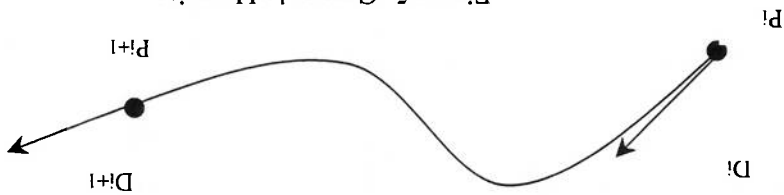
$$\tilde{Q}(n) = \sum_{i=0}^n B''_i(n) P_i \quad (0 \leq n \leq 1)$$

(2)

As curvas de Bézier [1] são curvas paramétricas. Ela é representada por um polígono de controle definido pelos pontos P_i . Exemplos de curvas de Bézier são exibidos na Figura 6. A curva de Bézier $Q(t)$ de grau n é criada com $n+1$ pontos de controle e sua representação matemática é dada por:

2.1.2 Curvas de Bézier

Figuras 5. Curva de Hermite.



A Figura 7 mostra o gráfico das bases de Bernstein para $n=3$. A curva de Bézier passa pelos pontos de início e fim do polígono de controle. E se um ponto de controle arbitrário é movimentado, a forma do polígono de controle muda e assim a forma da curva também se modifica. A derivada de primeira ordem da curva de Bézier de grau n é obtida derivando-se a equação (2):

$$B_3^0(u) = (1-u)^3, \\ B_3^1(u) = 3u(1-u)^2, \\ B_3^2(u) = 3u^2(1-u), \\ B_3^3(u) = u^3,$$

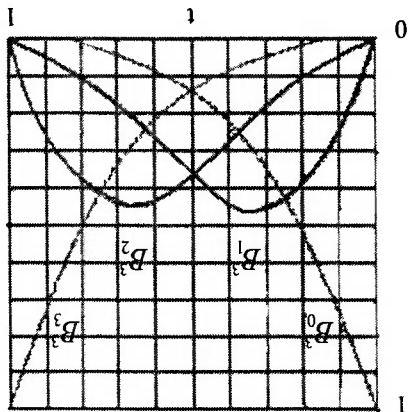
para $n=3$, temos

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}$$

e

$$B_n^i(u) = \binom{n}{i} u^i (1-u)^{n-i}$$

Figura 7. Bases de Bernstein para $n=3$.



(4)

(3)

É possível determinar uma fórmula de conversão entre curvas de Bézier de terceiro grau e curvas de Hermite também de terceiro grau. A curva de Bézier cúbica pode ser escrita em uma forma matricial pela expansão dos coeficientes dos polinômios de Bernstein presentes na definição analítica, e escrevendo estes coeficientes em uma forma matricial utilizando um polinômio em base de potências:

2.1.3 Conversão entre Curvas de Bézier e Curvas de Hermite

$$\frac{d\tilde{Q}(0)}{dn} = n(P^1 - P^0), \quad \text{e} \quad \frac{d\tilde{Q}(1)}{dn} = n(P^n - P^{n-1})$$

(7)

extremo da curva de Bézier são dadas por:

Então, baseando-se na equação (5), as derivadas de primeira ordem em cada

$$a'_i = P^{i+1} - P^i \quad (i = 0, \dots, n-1)$$

(6)

Onde

$$\frac{d\tilde{Q}(n)}{dn} = n \sum_{i=0}^{n-1} B'_{n-1}(n) a'_i \quad (0 \leq n \leq 1)$$

(5)

Assim, obtemos:

$$\begin{bmatrix} V_0' \\ V_1' \\ V_2' \\ V_3' \end{bmatrix} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_0' \\ P_1' \\ P_2' \\ P_3' \end{bmatrix} = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} D_0' \\ D_1' \\ D_2' \\ D_3' \end{bmatrix}$$

(10)

V_0, V_1, V_2, V_3 em função de P_0, P_1, P_2, P_3 e D_0, D_1, D_2, D_3 :

Eliminando os vetores com os parâmetros u é possível em seguida isolar os pontos

$$\begin{bmatrix} n_3 \\ n_2 \\ n_1 \\ n_0 \end{bmatrix} \begin{bmatrix} 1 \\ n \\ n^2 \\ n^3 \end{bmatrix} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_0' \\ V_1' \\ V_2' \\ V_3' \end{bmatrix} = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} D_0' \\ D_1' \\ D_2' \\ D_3' \end{bmatrix}$$

(9)

Para converter, devemos igualar as expressões (1) e (8) :

$$\begin{aligned} P(n) &= \sum_{i=0}^3 V_i B_i'(n) \\ &= (1-n)^3 \cdot V_0 + 3 \cdot n \cdot (1-n)^2 \cdot V_1 + 3 \cdot n^2 \cdot (1-n) \cdot V_2 + n^3 \cdot V_3 \\ &= \begin{bmatrix} V_0' \\ V_1' \\ V_2' \\ V_3' \end{bmatrix} \begin{bmatrix} (1-n)^3 \\ 3 \cdot n \cdot (1-n)^2 \\ 3 \cdot n^2 \cdot (1-n) \\ n^3 \end{bmatrix} \\ &= \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ n \\ n^2 \\ n^3 \end{bmatrix} \end{aligned}$$

(8)

Na última equação, $P_3^0(t)$ representa o ponto com parâmetro t sobre a curva de Bézier. Estas equações estão ilustradas na Figura 8. Quando a curva é dividida segundo o parâmetro t , os pontos de controle das duas curvas de Bézier criadas são:

$$\begin{aligned}
 P_0^1(t) &= (1-t)P_0^0(t) + tP_1^0(t) \\
 P_1^1(t) &= (1-t)P_1^0(t) + tP_2^0(t) \\
 P_2^1(t) &= (1-t)P_2^0(t) + tP_3^0(t) \\
 P_0^2(t) &= (1-t)P_0^1(t) + tP_1^1(t) \\
 P_1^2(t) &= (1-t)P_1^1(t) + tP_2^1(t) \\
 P_2^2(t) &= (1-t)P_2^1(t) + tP_3^1(t)
 \end{aligned}
 \tag{13}$$

expandida para:
 Bézier. Considerando o caso da curva de Bézier cúbica, a equação (12) pode ser $P_n^0(t)$ definido pela equação (12) é um ponto especificado pelo parâmetro t da curva de $P^i(t) = P^j$ representam os pontos de controle originais da curva. O vetor posição

$$\begin{aligned}
 P_r^i(t) &= (1-t)P_{r-1}^i(t) + tP_{r+1}^i(t) \\
 &\text{onde } r \text{ e } i \text{ variam como} \\
 &r=1, \dots, n \\
 &i=0, \dots, n-1
 \end{aligned}
 \tag{12}$$

Como apresentado acima, algumas curvas de contorno precisam ser divididas. A expressão abaixo define uma forma recursiva para a curva de Bézier:

2.1.4 Divisão de Curvas de Bézier

$$\begin{aligned}
 Y_0 &= P^j \\
 Y_1 &= P^j + 1/3 \cdot D^j \\
 Y_2 &= P^{j+1} - 1/3 \cdot D^{j+1} \\
 Y_3 &= P^{j+1}
 \end{aligned}
 \tag{11}$$

2.1.5 Conexão Entre Curvas de Bézier

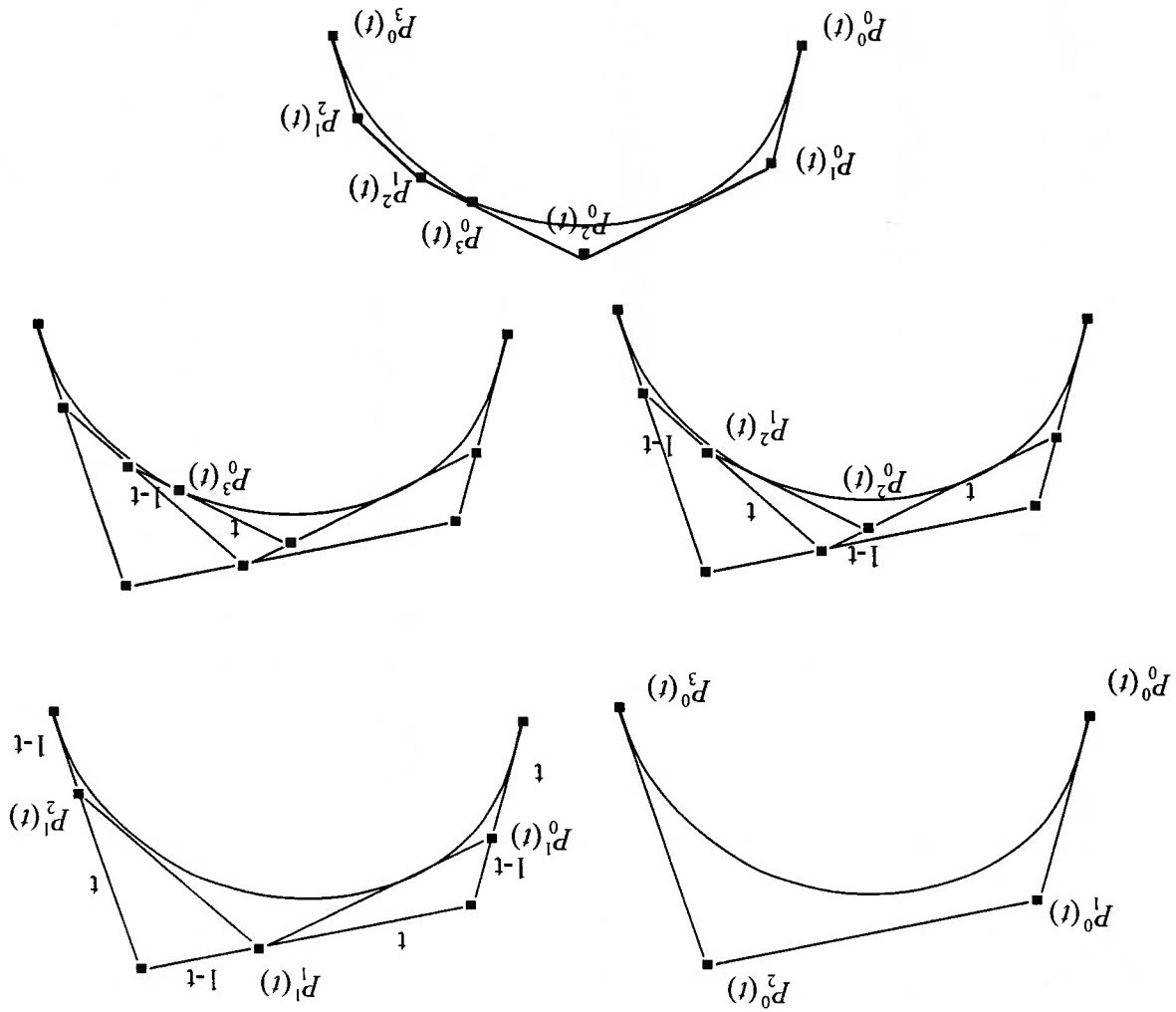
Os pontos de controle das duas curvas criadas pela divisão podem ser obtidos facilmente pela repetição de interpolações lineares. O mesmo método pode ser aplicado a qualquer curva de Bézier de grau n .

$$P_0^0(t), P_1^0(t), P_2^0(t), P_3^0(t)$$

e

$$P_0^3(t), P_1^3(t), P_2^3(t), P_3^3(t)$$

Figura 8. Divisão de curvas de Bézier.



$$\alpha \cdot 3 \cdot (P_{II}^1 - P_{II}^0) = 3 \cdot (P_1^3 - P_1^2) \quad (17)$$

Assim, no ponto de conexão, pela continuidade de posição, tem-se que $P_{II}^0 = P_1^3$. A Figura 9 ilustra a conexão entre duas curvas de Bézier de grau 3. A primeira derivada de uma curva de Bézier é dada pela expressão (14), igualando-se as derivadas nos pontos de conexão entre as duas curvas, obtém-se [14]:

$$P_{II}^1(u) = \sum_{l=0}^3 B_3^l(u) \cdot P_{II}^l \quad (16)$$

E, deseja-se conectar esta curva de Bézier com a curva de Bézier fornecida abaixo:

$$P_I^1(u) = \sum_{l=0}^3 B_3^l(u) \cdot P_I^l \quad (15)$$

Algo que ocorre com certa frequência em sistemas de CAD, é a necessidade de conectar duas curvas de Bézier de modo que exista continuidade C^1 no ponto de conexão. Portanto, considere a curva de Bézier abaixo:

$$\frac{d}{dn} P(u) = [3 \cdot u^2 \quad 2 \cdot n \quad 1 \quad 0] \cdot \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} \quad (14)$$

ao longo da curva de Bézier:

A partir da expressão (1) é possível determinar o valor da primeira derivada $d/dn P(u)$

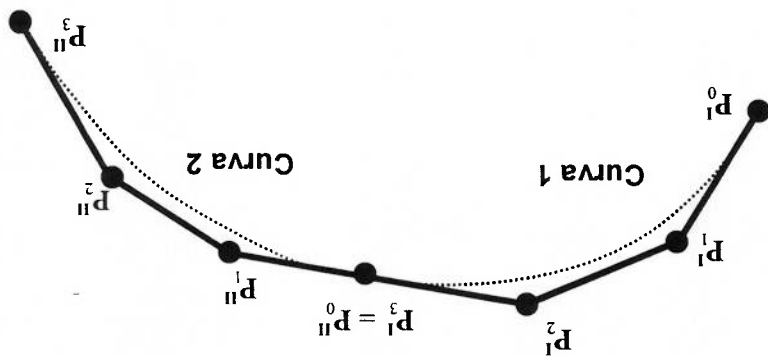
2.2.1 Superfície de Coons

As superfícies de Coons [6] são superfícies genéricas definidas pelas suas curvas de contorno. Sendo que as curvas de contorno são curvas, genéricas, podendo ser curvas de Bézier, Hermite, Interpolação Spline, B-Spline e outras. Neste capítulo apresentaremos um caso particular de superfície de Coons, onde as quatro curvas de contorno são curvas de Hermite. Apresentaremos também a superfície de Bézier. Mostraremos que é possível converter esta superfície de Coons particular para uma superfície de Bézier. Apresentaremos também como conectar superfícies de Bézier de modo que exista continuidade de derivada na transição entre uma superfície e outra. Será mostrado também que existe um grande problema em manter a continuidade, pois a modificação feita em uma superfície acarreta modificações nas superfícies vizinhas. Este problema será minimizado quando utilizarmos a superfície de Gregory.

2.2 Modelagem de Superfícies

Assim, percebe-se que os pontos de controle $P_1, P_2, P_3 = P_0 = P_{II}, P_1 = P_{II}$ e P_{II} devem ser colineares. Em algumas aplicações, entretanto, a continuidade C^1 não é exigida. Sendo necessário apenas, que os três pontos presentes na expressão (17) sejam colineares.

Figura 9. Conexão entre duas curvas de Bézier de grau 3.



$$\begin{aligned} \tilde{O}^n(n,1) &= \begin{bmatrix} P_w^{10} & P_w^{11} & P_{uw}^{10} & P_{uw}^{11} \end{bmatrix}^H M M \\ \tilde{O}^n(n,0) &= \begin{bmatrix} P_w^{00} & P_w^{10} & P_{uw}^{00} & P_{uw}^{10} \end{bmatrix}^H M M \\ \tilde{O}^n(1,w) &= \begin{bmatrix} P_u^{10} & P_u^{11} & P_{uw}^{10} & P_{uw}^{11} \end{bmatrix}^H M M \\ \tilde{O}^n(0,w) &= \begin{bmatrix} P_u^{00} & P_u^{10} & P_{uw}^{00} & P_{uw}^{10} \end{bmatrix}^H M M \end{aligned}$$

seguintes vetores tangentes cruzados:

Também será necessário considerar que sobre o contorno estão definidos os superfícies.

O parâmetro u e w estão na faixa $0 \leq u, w \leq 1$ segundo as direções U e W da

$$\begin{aligned} \tilde{O}(1,w) &= WM^H \begin{bmatrix} P_w^{10} & P_w^{11} & P_w^{10} & P_w^{11} \end{bmatrix} \\ \tilde{O}(0,w) &= WM^H \begin{bmatrix} P_w^{00} & P_w^{10} & P_w^{00} & P_w^{10} \end{bmatrix} \\ \tilde{O}(u,1) &= UM^H \begin{bmatrix} P_u^{10} & P_u^{11} & P_u^{10} & P_u^{11} \end{bmatrix} \\ \tilde{O}(u,0) &= UM^H \begin{bmatrix} P_u^{00} & P_u^{10} & P_u^{00} & P_u^{10} \end{bmatrix} \end{aligned}$$

Considere as quatro curvas definidas abaixo:

$$\text{Onde } U = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} e M^H = \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}$$

$$\tilde{O}(u) = UM^H G^H$$

como

Como mostrado pela equação (1), as curvas de Hermite podem ser expressas

A superfície de Bézier é definida por uma matriz de pontos de controle. Uma superfície de Bézier de grau $n \times m$ é representada por:

2.2.2 Superfície de Bézier

(figura 10).

Esta forma da superfície de Coons também é conhecida como Ferguson Patch

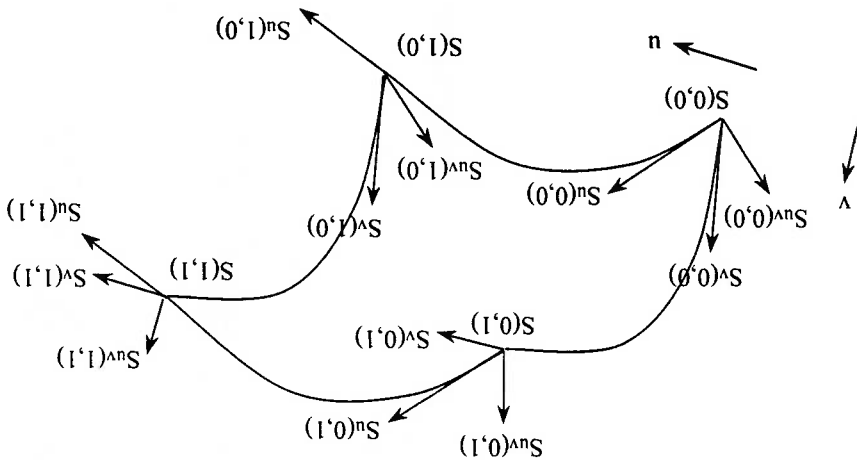
$$B = \begin{bmatrix} P_{00} & P_{01} & P_{10} & P_{11} \\ P_{10} & P_{11} & P_{00} & P_{01} \\ P_{00} & P_{01} & P_{10} & P_{11} \\ P_{10} & P_{11} & P_{00} & P_{01} \end{bmatrix} \begin{bmatrix} P_u^{10} \\ P_u^{00} \\ P_w^{10} \\ P_w^{00} \end{bmatrix}$$

Onde

$$\mathcal{Q}(u, w) = UM^H B M^H W^T \quad (19)$$

A superfície de Coons onde as quatro curvas de contorno são curvas de Hermite pode ser representada como [12]:

Figura 10. Superfície de Coons.



(22)

$$UM^H BM^T W^T = UM^B P M^T W^T$$

pelas equações (18) e (21) obtemos:

E para o caso (m=n=3) é possível observar que 12 pontos de controle definem o contorno da superfície, e 4 pontos de controle ($P_{22}, P_{23}, P_{32}, P_{33}$) não estão sobre o contorno da superfície, mas definem a sua forma. A Figura 11 ilustra um exemplo de superfície de Bézier com m=n=3. Se temos a mesma superfície representada por Coons e Bézier, então

$$M_B = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P_{41} & P_{42} & P_{43} & P_{44} \end{bmatrix}$$

onde:

(21)

$$\tilde{Q}(u, w) = UM^B P M^T W^T$$

Coons, as superfícies de Bézier podem ser expressas como:

Aqui, $P_{i,j}$ indica os pontos de controle. Sendo que n+1 pontos de controle ficam na direção u e m+1 pontos de controle ficam na direção w, totalizando (n+1) × (m+1) pontos de controle. $B_{i,n}(u)$ e $B_{j,m}(w)$ são polinômios de Bernstein. Tal como as superfícies de

(20)

$$\tilde{Q}(u, w) = \sum_n \sum_{i=0}^n B_{i,n}(u) B_{j,m}(w) P_{i,j}, \text{ onde } 0 \leq u, w \leq 1$$

O ponto P_{00} da superfície de Coons é equivalente ao ponto Q_{11} da superfície de Bézier. O ponto Q_{12} da superfície de Bézier é determinado pela soma vetorial $P_{00} + \frac{1}{3}P_{0n}$. O ponto Q_{21} da superfície de Bézier é determinado pela soma vetorial $P_{00} + \frac{1}{3}P_{n0}$. O ponto Q_{22}

onde $P_{uv}^{00}, P_{uv}^{01}, P_{uv}^{10}$ e P_{uv}^{11} são vetores de torção que são nulos em nosso caso. A equação (23) determina os pontos de controle da superfície de Bézier a partir dos parâmetros da superfície de Coons. A Figura 12 ilustra uma interpretação para a equação

$$P = M_B^{-1} M_H^B M_T^H [M_T^B]^{-1} = \begin{bmatrix} P_{00} & P_{00} + \frac{1}{3}P_{n0} & P_{00} + \frac{1}{3}P_{0n} & P_{10} \\ P_{00} + \frac{1}{3}P_{uv}^{00} + \frac{1}{9}(P_{n0} + P_{00}) + \frac{1}{9}P_{uv}^{00} & P_{01} + \frac{1}{3}P_{uv}^{01} + \frac{1}{9}(P_{n0} - P_{01}) - \frac{1}{9}P_{uv}^{01} & P_{01} + \frac{1}{3}P_{uv}^{01} + \frac{1}{9}(P_{n1} + P_{11}) + \frac{1}{9}P_{uv}^{11} & P_{10} + \frac{1}{3}P_{uv}^{10} \\ P_{00} + \frac{1}{3}P_{uv}^{00} & P_{01} - \frac{1}{3}P_{uv}^{01} & P_{01} - \frac{1}{3}P_{uv}^{01} & P_{10} + \frac{1}{3}P_{uv}^{10} \\ P_{00} & P_{01} + \frac{1}{3}P_{n0} & P_{01} + \frac{1}{3}P_{0n} & P_{11} \end{bmatrix} \quad (23)$$

então

Figura 11 Exemplo de superfície de Bézier.

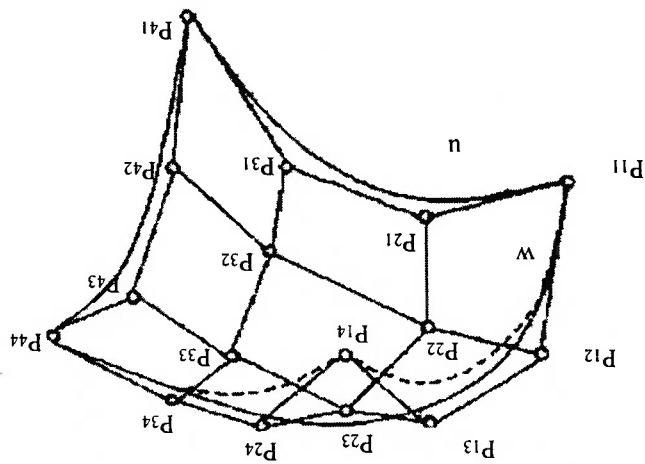
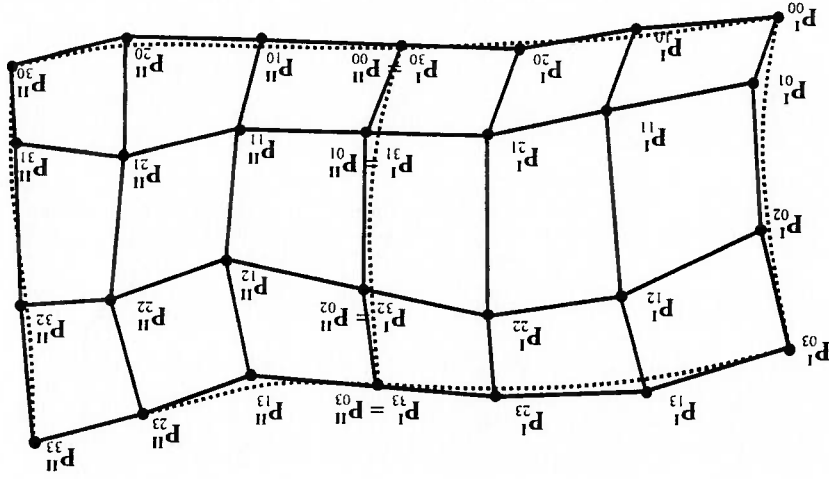


Figura 13. Conexão entre duas superfícies de Bézier de grau 3.



Semelhantemente ao problema de conectar duas curvas de Bézier, é necessário

2.2.3 Conexão Entre Superfícies de Bézier

portos da superfície de Bézier são determinados de forma semelhante.

da superfície de Bézier é determinado pela soma vetorial $P_{00} + \frac{3}{1}(P_{10}^n + P_{01}^w) + \frac{1}{9}P_{00}^{nw}$. Os outros

Figura 12 Conexão entre superfície de Bézier e superfície de Coons.

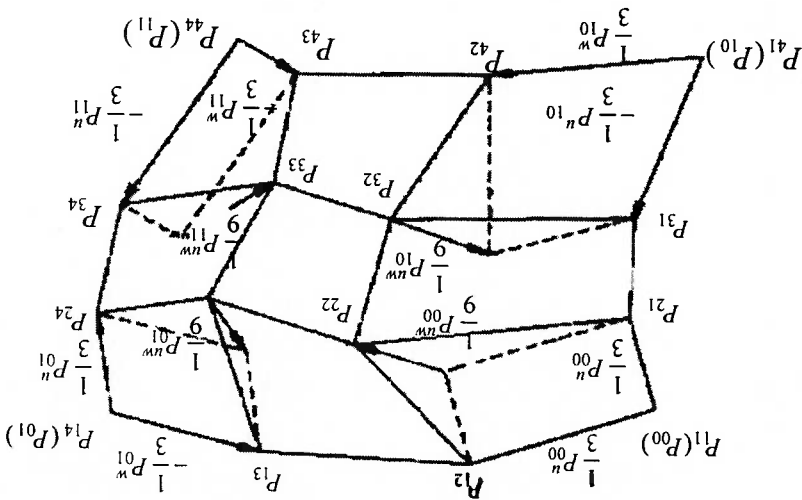
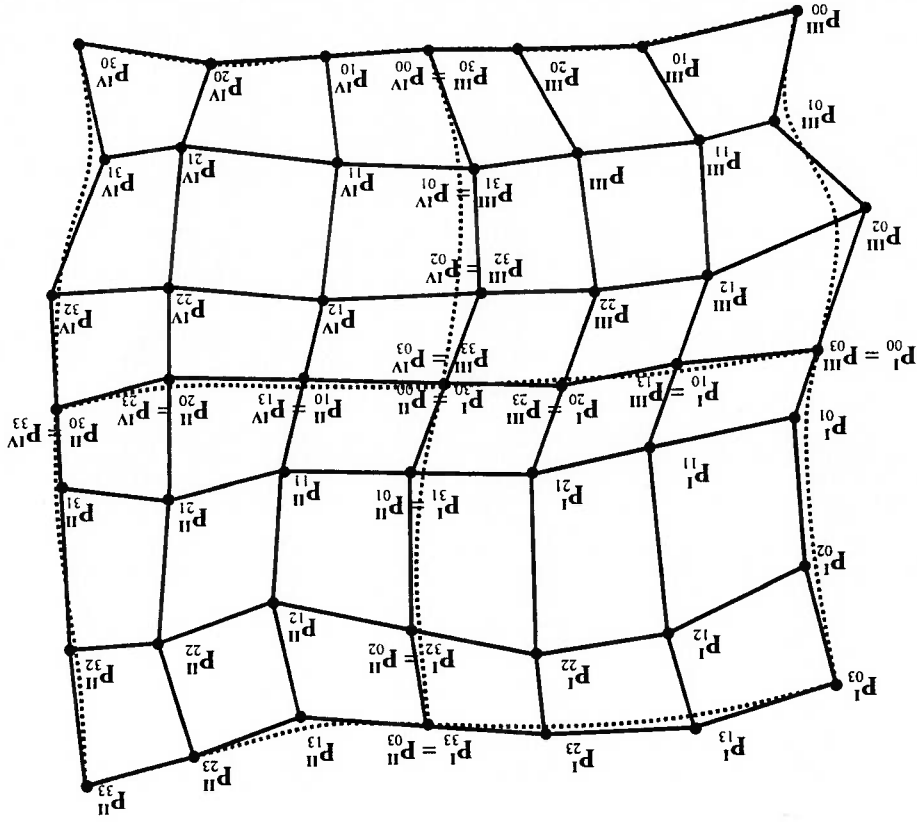


Figura 14. Exemplo de conexão entre quatro superfícies de Bézier.

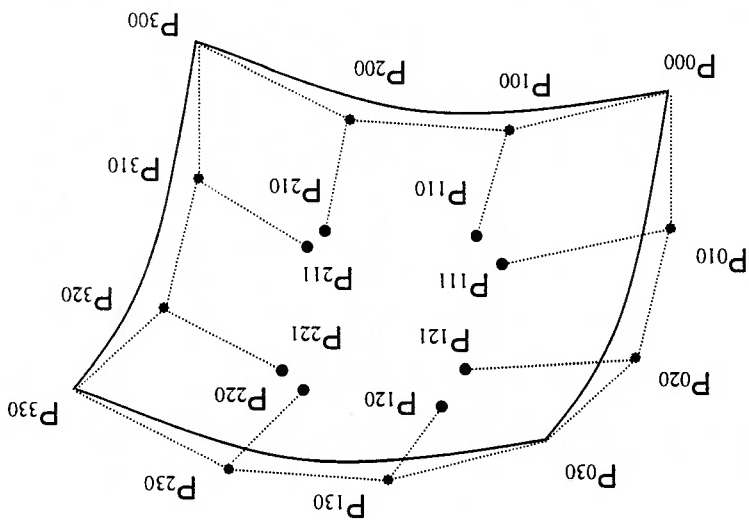


Caso a análise feita anteriormente seja estendida, e quatro superfícies de Bézier $P^I(u,w)$, $P^{II}(u,w)$, $P^{III}(u,w)$, $P^{IV}(u,w)$ adjacentes (vide Figura 14) sejam consideradas, teremos então quatro curvas comuns de contorno. A restrição de continuidade C^1 deve ser aplicada ao longo de todas as quatro curvas comuns de contorno. Para a curva comum de contorno entre as superfícies $P^I(u,w)$ e $P^{II}(u,w)$ têm-se que os pontos de controle P^I_{31} ,

$$P^{II}_{01} \text{ (} i=0,1,2,3 \text{)} \text{ devem ser colineares [14].}$$

requisitos para que esta continuidade exista é que os pontos de controle P^I_{31} , P^{II}_{01} e pelos pontos de controle $P^{II}_{31} = P^{III}_{01}$ ($i=0,1,2,3$). É possível demonstrar que um dos pontos de controle P^I_{31} , P^{II}_{01} e P^{III}_{01} devem ser colineares [14].

Figura 15 Exemplo de Gregory Patch.



superfície de Gregory são as seguintes:

conjunto de 20 pontos de controle P_{ij}^k ($i = 0, \dots, 3$, $j = 0, \dots, 3$, $k = 0, 1$). As equações da Bézier [8,13]. Como ilustrado na Figura 15, uma superfície de Gregory é definida por um algumas das propriedades da superfície de Gregory na forma de superfície cúbica de Chiyokura e Kimura [4] estudaram a superfície de Gregory. A seguir, serão discutidas modificações da forma da superfície. Tentando diminuir esta condição de amarramento, (exibidos na Figura 14) é muito forte, restringindo de modo acentuado a possibilidade de A condição de amarramento entre os quatro pontos de controle P_{11}^I , P_{11}^{II} , P_{11}^{III} e P_{11}^{IV}

2.2.4 Gregory Patch

continuidade C^1 ao longo das quatro curvas comuns de contorno. outros três pontos também deverão ser movimentados para manter a condição de estão intrinsecamente relacionados. Assim, se um destes pontos for movimentado, os contorno é possível obter um resultado semelhante, conclui-se que P_{11}^I , P_{11}^{II} , P_{11}^{III} e P_{11}^{IV} devem ser colineares. Como para as outras curvas comuns de $P_{11}^I = P_{11}^{II} = P_{11}^{III} = P_{11}^{IV}$ ($i = 0, 1, 2, 3$)

$$\begin{aligned}
 \mathcal{Q}^v(n,0) &= \mathcal{E} \cdot \sum_{i=0}^3 B'_i(n) (P^{i,0} - P^{0,i}) & \mathcal{Q}^n(0,v) &= \mathcal{E} \cdot \sum_{i=0}^3 B'_i(v) (P^{i,0} - P^{0,i}) \\
 \mathcal{Q}^v(n,1) &= \mathcal{E} \cdot \sum_{i=0}^3 B'_i(n) (P^{i,30} - P^{30,i}) & \mathcal{Q}^n(1,v) &= \mathcal{E} \cdot \sum_{i=0}^3 B'_i(v) (P^{i,30} - P^{30,i})
 \end{aligned}$$

(26)

forma independente [5]:

A superfície de Gregory pode ser degenerada para uma superfície de Bézier no caso em que os pontos internos de controle satisfizerem $P^{j_0} = P^{j_1}$. Também é possível determinar os quatro vetores tangentes cruzados às curvas de contorno, que serão especificados de

$$\begin{aligned}
 \mathcal{Q}_{11}(u,v) &= \frac{n + v}{n \cdot P^{1,10} + v \cdot P^{1,11}} \\
 \mathcal{Q}_{21}(u,v) &= \frac{(1-n) \cdot P^{2,10} + v \cdot P^{2,11}}{(1-n) + v} \\
 \mathcal{Q}_{12}(u,v) &= \frac{n \cdot P^{1,20} + (1-v) \cdot P^{1,21}}{n + (1-v)} \\
 \mathcal{Q}_{22}(u,v) &= \frac{(1-n) \cdot P^{2,20} + (1-v) \cdot P^{2,21}}{(1-n) + (1-v)}
 \end{aligned}$$

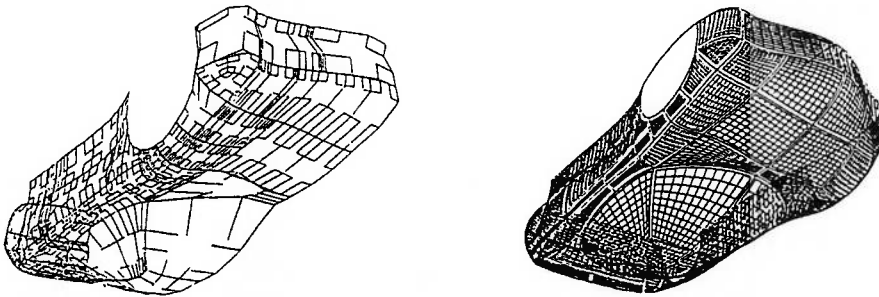
(25)

Os pontos $\mathcal{Q}_{11}(u,v)$, $\mathcal{Q}_{12}(u,v)$, $\mathcal{Q}_{21}(u,v)$ e $\mathcal{Q}_{22}(u,v)$ são definidos por expressões especiais:

$$\mathcal{Q}(u,v) = \sum_{i=0}^3 \sum_{j=0}^3 B'_i(u) \cdot B'_j(v) \cdot \mathcal{Q}_{ij}(u,v) \quad 0 \leq u, v \leq 1$$

(24)

Figura 16. Exemplo de carroceria modelada utilizando superfícies de Gregory.



Esta última propriedade permite definir métodos localizados de interpolação. Assim, as superfícies de Gregory apresentam uma grande vantagem em relação às superfícies de Bézier, uma vez que os pontos internos de controle estão associados a apenas uma curva de contorno. Na modelagem foram utilizada 140 superfícies de Gregory para representar uma carroceria (vide figura 16) [4].

2.3 Superfícies definidas automaticamente

Como já discutido no capítulo anterior, diversos sistemas de CAD/CAM requerem que a superfície tenha quatro lados para que expressões polinomiais sejam associadas. As superfícies que não possuem quatro lados são subdivididas de modo que apenas superfícies com quatro lados sejam criadas. A seguir explicaremos em maior detalhe o algoritmo proposto por Chiyokura e Kimura [4]. A Figura 17 ilustra casos de superfícies que não possuem quatro lados criada por operações de suavização de arestas e vértices.

2.3.1 Face Triangular

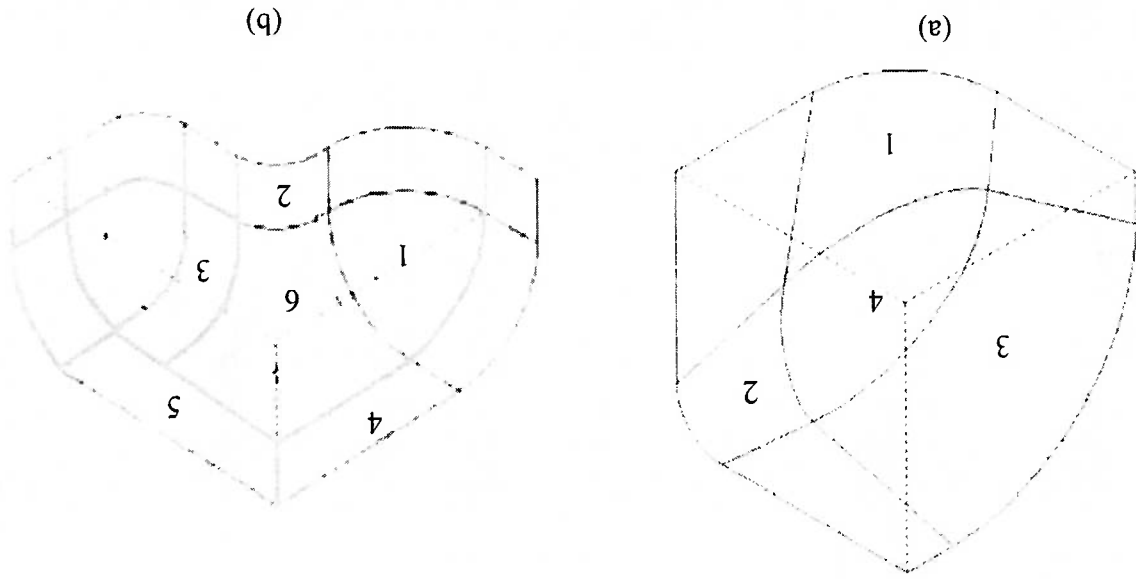


Figura 17 Um sólido que possui uma face triangular e um sólido com uma face de cinco lados.

podem ser determinados: $Q_i = \frac{3P_{i+1} - P_{i,0}}{2}$ ($i = 1, \dots, 3$). Em seguida, será determinado o ponto

Nesta condição, os pontos de controle das curvas quadráticas de Bézier, Q_1, Q_2 e Q_3 . Agora, assuma que as curvas internas são quadráticas para simplificar o algoritmo.

determinados.

os vetores normais são conhecidos então os pontos de controle $P_{i,j}$ ($i = 1, \dots, 3$) podem ser igual a 0,5. Estes pontos são também os extremos das curvas internas. Considerando que Na Figura 18, $P_{1,0}, P_{2,0}, P_{3,0}$ são pontos sobre as curvas com valores de parâmetro

e ilustrado pela Figura 15.

determinam-se três curvas que conectem o ponto P_0 aos pontos $P_{3,0}, P_{2,0}$ e $P_{1,0}$. O algoritmo Em seguida determina-se um ponto P_0 que está sobre a superfície. Por último fazendo com que o número de curvas dobre. Os pontos determinados são $P_{3,0}, P_{2,0}$ e $P_{1,0}$. Esta proposta foi apresentada por Chiyokura e Kimura [4], sendo que a proposta se concentra em determinar um ponto central sobre cada uma das curvas de contorno,

Figura 18. Caso de face com três lados.

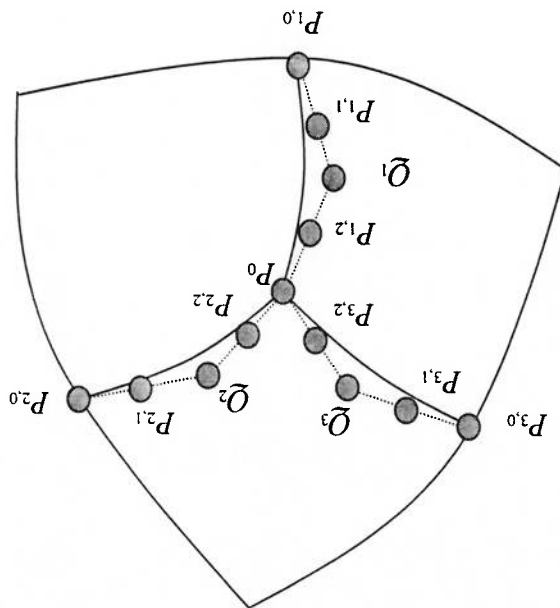
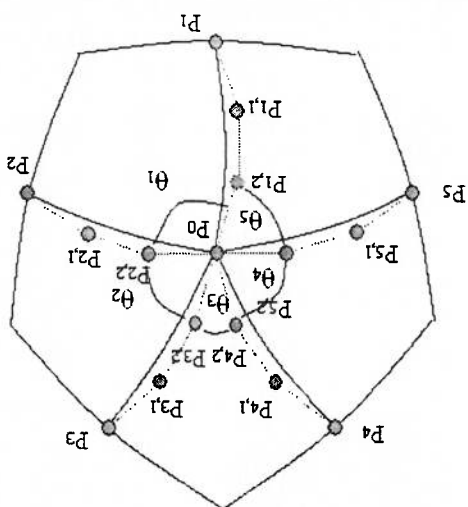


Figura 19. Caso de face com cinco lados.



Como ilustrado na Figura 17, é possível que surja um polígono com cinco lados e Chiyokura e Kimura [4] propuseram um algoritmo semelhante ao da proposta anterior para permitir que a geometria seja associada à face. Primeiro determina-se o ponto sobre cada uma das curvas de contorno, fazendo com que o número de curvas dobre. Em seguida um ponto R_0 que está sobre a superfície é determinado. E por último determinam-se cinco curvas que conectam o ponto R_0 aos outros cinco pontos determinados inicialmente (Vide Figura 19). Conforme na proposta de Chiyokura e Kimura [4].

2.3.2 Face com Cinco Lados

curvas cúbicas pela elevação do grau da curva de Bézier. pontos $P_i, Q_i, e R_0$ ($i = 1, \dots, 3$). Finalmente, as curvas internas são representadas como dos pontos $Q_1, Q_2, e Q_3$ ($R_0 = \frac{Q_1 + Q_2 + Q_3}{3}$), uma curva quadrática de Bézier é definida pelos R_0 onde as curvas internas se encontram. O ponto R_0 é determinado pela posição média

3 Modelagem de Sólidos

Neste capítulo apresentaremos a técnica de modelagem de sólidos conhecida como B-Rep (Boundary Representation) [10, 11]. Inicialmente detalharemos as propostas existentes para a sua estrutura de dados. Em seguida apresentaremos os Operadores de Euler que foram definidos para facilitar a manipulação da estrutura de dados. E por último, detalharemos algumas funções de alto nível, conhecidas como Operadores Locais, que criam alguns sólidos primitivos. Os operadores locais fazem uso exclusivamente de Operadores de Euler para editar a estrutura de dados.

3.1 Estrutura de Dados

Nesta parte explicaremos como os elementos primitivos são representados de modo a definir um sólido B-Rep. Um modelo B-Rep possui sete tipos de elementos primitivos (vértice, meia-aresta, aresta, laço, face, shell e região). Temos três estruturas de dados propostas na literatura que implementam a representação B-Rep. Todas baseadas na aresta como elemento de referência: estrutura *winged-edge*, estrutura *meia-aresta* e estrutura unificada. Neste trabalho, será utilizada a estrutura unificada.

A estrutura *winged-edge* mantém as informações de adjacência por meio de ponteiros a

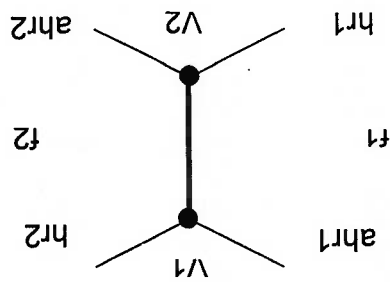


Figura 20. Estrutura *winged-edge*.
 hr = aresta no sentido horário
 ahr = aresta no sentido anti-horário

v1	v2
hr1	hr2
ahr1	ahr2
f1	f2

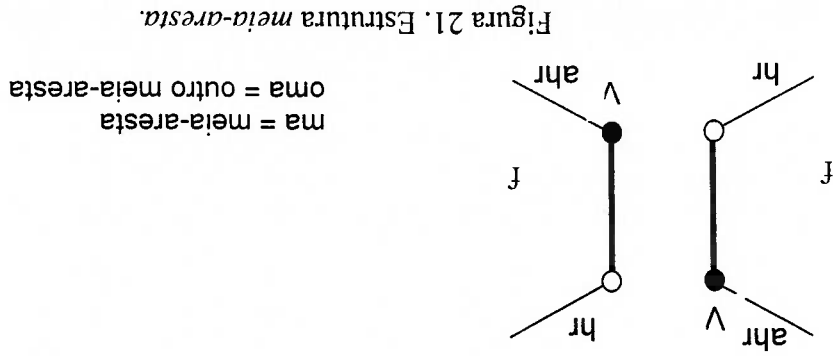


Figura 21. Estrutura meia-aresta.

Até este momento, foi assumido que cada face possui apenas um contorno. Entretanto, casos práticos requerem que uma face possua mais de um contorno (como uma face com furos). Faces com mais de um contorno podem ser simuladas pela técnica de aresta-ponte (*bridge-edge*) no qual uma aresta une os contornos de uma face entre si. A aresta-ponte, portanto, possui a mesma face adjacente em ambas as laterais (vide Figura 22 (a)).

A estrutura *meia-aresta* representa a metade das informações de adjacência da estrutura *winged-edge*. Cada *meia-aresta* possui apenas uma orientação, e cada face possui um circuito direcional de *meia-arestas* (vide figura 21). Onde v é ponto para o vértice da *meia-aresta*, hr e ahr são pontos para as duas *meia-arestas* adjacentes e f é ponto para a face adjacente à *meia-aresta*. Para evitar a duplicidade das informações associadas à aresta (por exemplo, atributo de cor) nas duas *meia-arestas*, foi proposta a criação de uma nova estrutura para a aresta que associaria as duas *meia-arestas* que definem a aresta de referência, evitando assim o ponto para a outra *meia-aresta*. Esta é a estrutura unificada.

vários elementos adjacentes à aresta de referência: duas faces, dois vértices e quatro arestas. Cada uma das quatro arestas compartilha com a aresta de referência um vértice e uma face (vide Figura 20). Onde v_1 e v_2 são pontos para os dois vértices adjacentes, hr_1 , ahr_1 , hr_2 e ahr_2 são pontos para as quatro arestas adjacentes, e f_1 e f_2 são pontos para as duas faces adjacentes.

Entretanto, a técnica aresta-ponte não é muito eficiente porque será necessário determinar como os contornos devem ser conectados pelas arestas-ponte, o que criará a necessidade de complexos algoritmos para implementar operações de modelagem. Como exemplo, podemos citar os Operações Booleanas que provavelmente interseccionarão as arestas-ponte. Alterações na estrutura de dados unificada de maneira a suportar faces

Figura 23. Sólido com duas regiões.

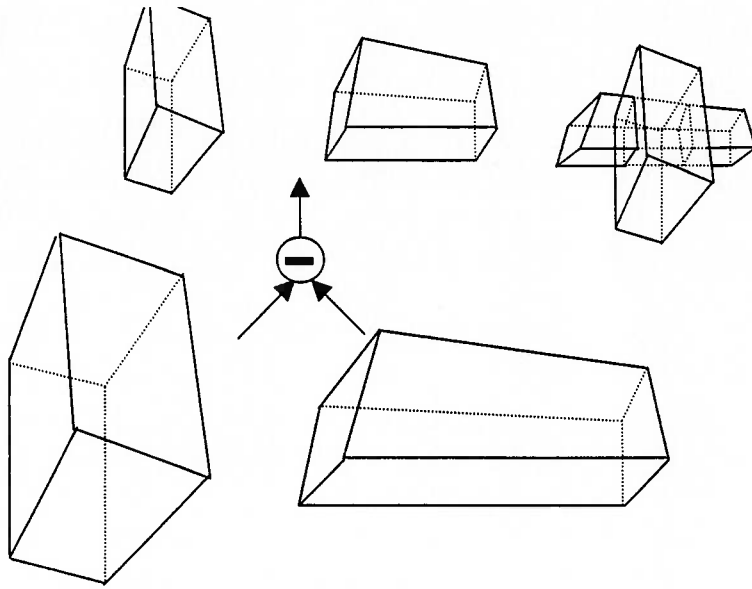
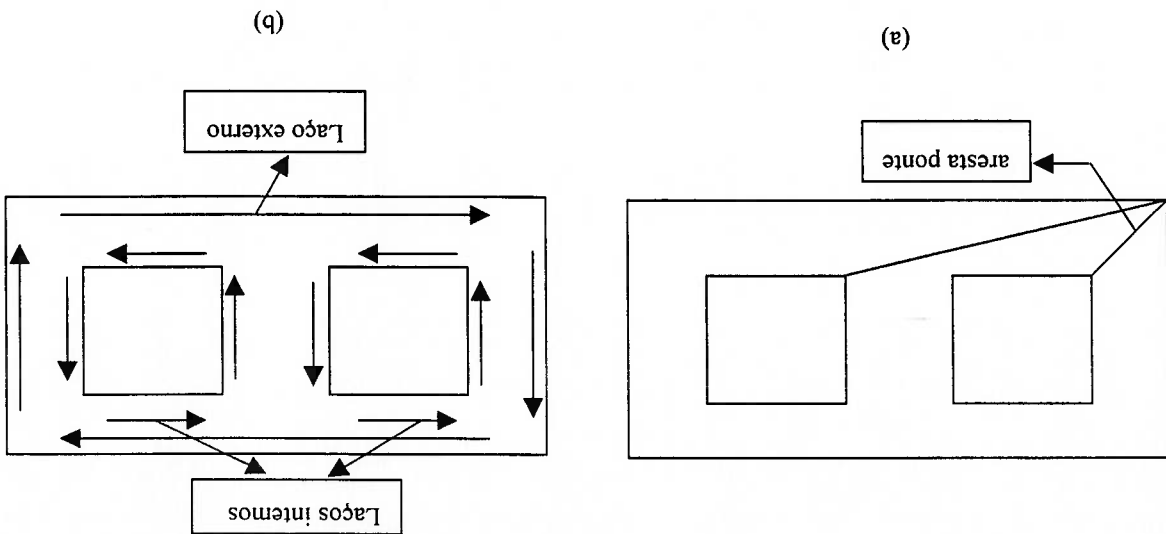


Figura 22. Extensão para faces com múltiplos contornos.



com mais de um contorno não afetará a estrutura B-Rep ao nível de aresta, mas sim, ao nível de face. Uma técnica muito comum é adicionar uma estrutura de tamanho fixo chamada laço (loop) que é associada a cada contorno da face. A estrutura laço simplesmente fornece à estrutura face um mecanismo para manter uma lista ligada de ponteiros para os seus múltiplos contornos. Cada face possui um laço externo e zero ou mais laços internos (vide Figura 22 (b)).

Devido ao formalismo das Operações Booleanas, ao combinarmos dois sólidos por uma Operação Booleana, em geral, o resultado será um sólido. Entretanto, mesmo em situações especiais, como a situação exemplificada na Figura 23, onde o resultado da Operação Booleana aparenta apresentar dois sólidos, o resultado é considerado como sendo apenas um sólido. Entretanto, nesta situação em especial, considera-se que o sólido resultante possui duas regiões. Em outras palavras, para representar esta situação especial, foi criado o elemento região que representa conjuntos disjuntos de faces no espaço. A Figura 24 ilustra os elementos primitivos de um modelo sólido. A Figura 25

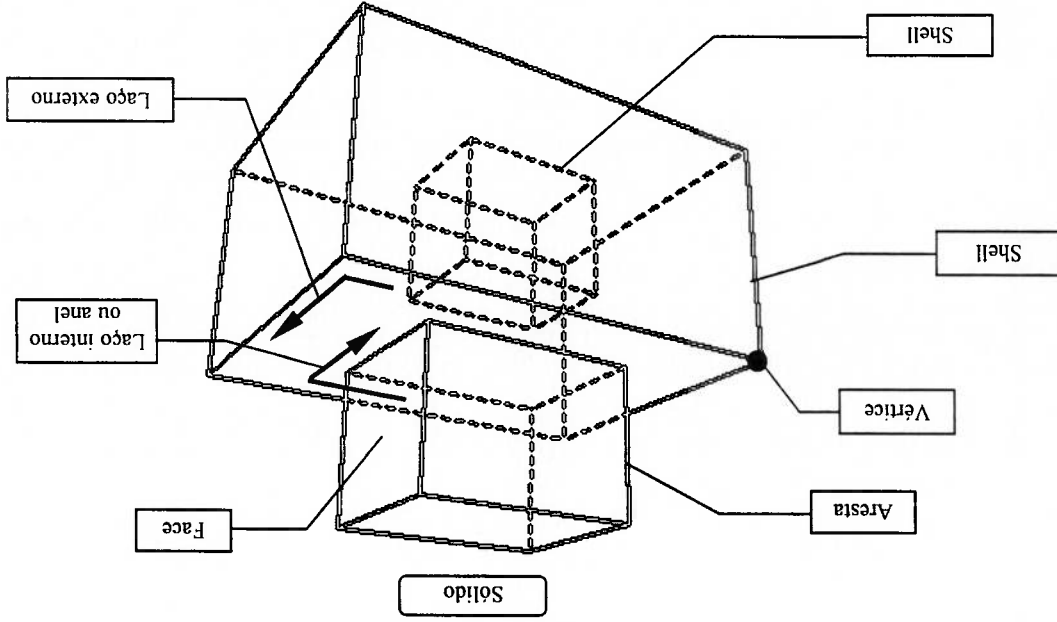


Figura 24. Elementos topológicos de um modelo sólido.

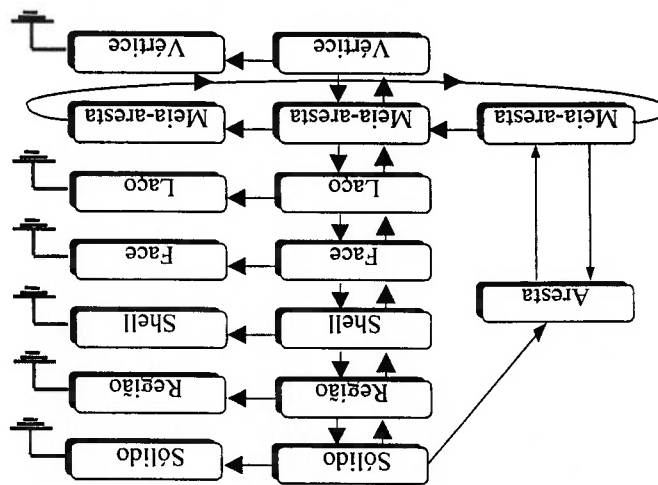
A equação de Euler-Poincaré diz que um sólido polidédrico é topologicamente válido se a seguinte relação entre as suas quantidades de elementos for verificada (vide figura 26):

segundo nível de representação do modelador B-Rep. todos os detalhes de implementação da estrutura de dados. Estes operadores formam o segundo nível de representação do sólido possa ser executada passo a passo, escondendo a construção de estruturas de dados da representação B-Rep mais intuitiva. Eles permitem que a manipulação dos dados da representação B-Rep seja intuitiva. Eles permitem que a construção do sólido possa ser executada passo a passo, escondendo a construção de estruturas de dados da representação B-Rep mais intuitiva. Eles permitem que a construção do sólido possa ser executada passo a passo, escondendo a construção de estruturas de dados da representação B-Rep mais intuitiva.

3.2 Operadores de Euler

Para modificar e manipular elementos da estrutura para modificar e manipular elementos da estrutura. Neste trabalho, utilizamos a estrutura unificada para representar sólidos polidédricos, e desenvolvemos uma estrutura adaptada da estrutura unificada para representar curvas e superfícies. Deste modo, foi possível desenvolver algoritmos e operadores consistentes para modificar e manipular elementos da estrutura.

Figura 25. Hierarquia de elemento da estrutura unificada.



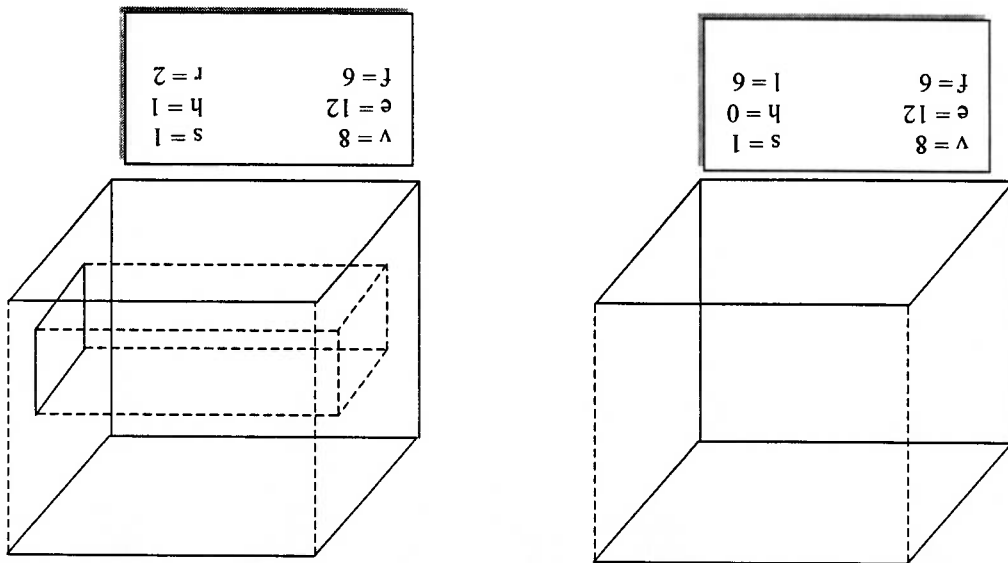
Vários autores demonstram que sete operadores são suficientes para construir todos os sólidos. Enquanto estes sete operadores podem ser escolhidos de várias maneiras, considerações de modularidade e independência criaram apenas pequenas variações na coleção encontrada na literatura [11, 14]. Com a finalidade de facilitar a memorização, os Operadores de Euler estão definidos utilizando a nomenclatura da Tabela 1.

(28)
$$v - e + f = 2(s - h) + r$$

onde v é o número de vértices do sólido, e o número de arestas, f o número de faces, s o número de shells, h o número de furos e l o número de laços. A forma mais conhecida na literatura da equação de Euler-Poincaré supõe ainda a existência de r anéis no sólido, onde $r = l - f$. Ficando a equação da seguinte forma:

(27)
$$v - e + 2f = 2(s - h) + l;$$

Figura 26. Sólidos verificando a equação de Euler-Poincaré.



- **MVSF** (*Make Vertex Solid Face*) : este operador cria um sólido inicial com apenas uma face e um vértice;
- **MEV** (*Make Edge Vertex*) : este operador adiciona a um sólido uma aresta e um vértice. A aresta é criada conectando-se um vértice já existente ao novo vértice criado;
- **MEF** (*Make Edge Face*) : este operador adiciona ao sólido uma aresta e uma face. A face é criada pela divisão de uma face já existente acrescentando-se a nova aresta;
- **KEMR** (*Kill Edge Make Ring*) : este operador divide o contorno de uma face em dois laços pela remoção de uma aresta-ponte;
- **KFMRH** (*Kill Face Make Ring Hole*) : nenhum dos operadores discutidos anteriormente é capaz de modificar as propriedades topológicas globais da estrutura de dados, como dividir um sólido em dois componentes ou criar um furo passante. O operador **KFMRH** possui este objetivo;
- **MSFKR** (*Make Shell Face Kill Ring*) : este é outro operador que manipula informações globalmente. Ele transforma o anel de uma face em uma nova face, e todo o conjunto de faces associadas à nova face constituirá uma novo *shell*;

27.

Por exemplo, o operador **MEV** deve ser traduzido por *Make Edge, Vertex* (Crie uma aresta e um vértice). A seguir são descritos os sete operadores mais comumente utilizados na literatura e uma representação gráfica para cada operador pode ser observada na Figura

Simbolo	Significado
M	make
K	kill
V	vertex
E	edge

Simbolo	Significado
F	face
S	shell ou solid
H	hole
R	ring ou region

Tabela 1. Nomenclatura dos Operadores de Euler

Os sete operadores detalhados acima são todos construtivos entretanto, para que o sólido possa ser alterado satisfatoriamente ele precisa também de algumas operações destrutivas. Por isto, cada operador construtivo possui um operador correspondente destrutivo. São eles:

- **MRSFKR (Make Region Shell Face Kill Ring)**: este é outro operador que manipula informações globalmente. Ele transforma o anel de uma face em uma nova face, e todo o conjunto de faces associadas à nova face constituirá uma nova região.

Figura 27. Representação gráfica dos operadores de Euler.

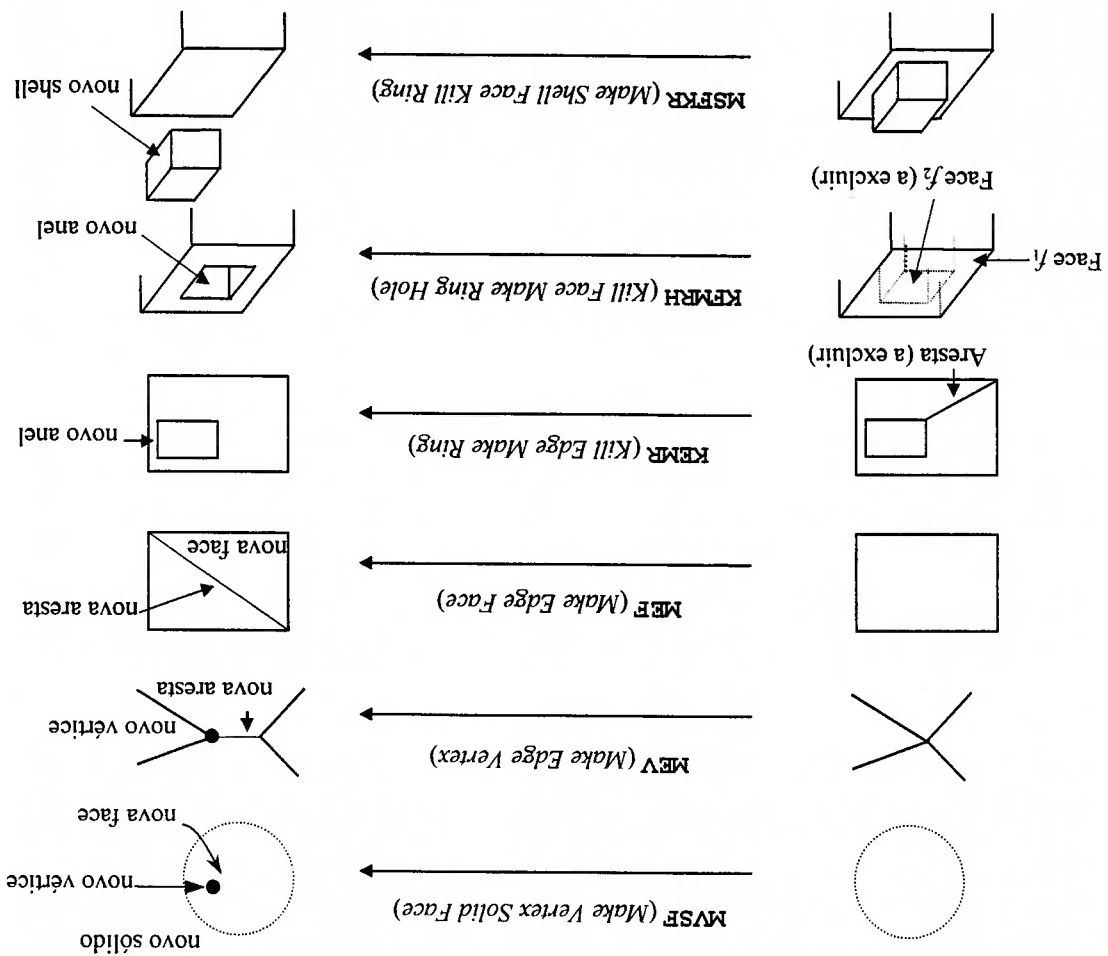
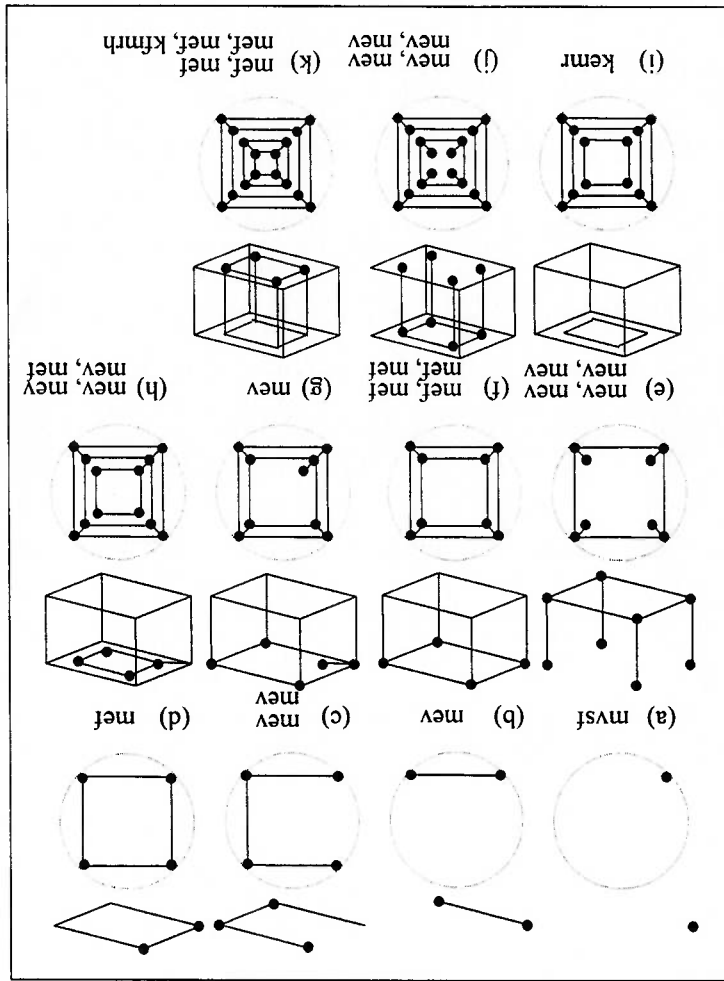


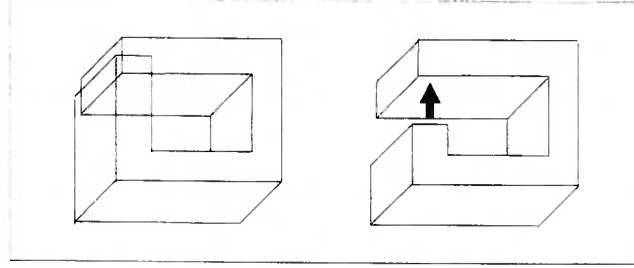
Figura 28. Sequência de Operadores de Euler para criação de um cubo com furo passante.



\Rightarrow KRSTMR (Kill Region Shell Face Make Ring)	\Rightarrow MRSFKR (Make Region Shell Face Kill Ring)
\Rightarrow MSFMR (Kill Shell Face Make Ring)	\Rightarrow MSFKR (Make Shell Face Kill Ring)
\Rightarrow MEKRH (Make Face Kill Ring Hole)	\Rightarrow KFRBH (Kill Face Make Ring Hole)
\Rightarrow MEKR (Make Edge Kill Ring)	\Rightarrow KEMR (Kill Edge Make Ring)
\Rightarrow KEF (Kill Edge Face)	\Rightarrow MEF (Make Edge Face)
\Rightarrow KEV (Kill Edge Vertex)	\Rightarrow MEV (Make Edge Vertex)
\Rightarrow KVSF (Kill Vertex Solid Face)	\Rightarrow MVSF (Make Vertex Solid Face)

Tabela 2. Operadores construtivos e destrutivos

Figura 29. Um sólido auto-interceptante.



Euler que a implemente.

Um operador local pode ser considerado uma extensão das Operações de Euler. Um operador local, ao ser acionado pelo usuário, definirá uma sequência de Operadores de

Um operador local pode ser considerado uma extensão das Operações de Euler. Um operador local, ao ser acionado pelo usuário, definirá uma sequência de Operadores de Euler que a implemente. Um operador local pode ser considerado uma extensão das Operações de Euler. Um operador local, ao ser acionado pelo usuário, definirá uma sequência de Operadores de Euler que a implemente. Um operador local pode ser considerado uma extensão das Operações de Euler. Um operador local, ao ser acionado pelo usuário, definirá uma sequência de Operadores de Euler que a implemente.

3.3 OPERADORES LOCAIS

Durante o processo de construção de um sólido pela utilização de Operadores de Euler, a validade topológica do mesmo é mantida observando-se a equação de Euler-Poincaré. Entretanto, é comum o agrupamento de Operadores de Euler em uma certa sequência para mantermos também a geometria válida. É importante observar que não há como manter a geometria válida em todos os estágios da construção. Por isto, os Operadores de Euler devem ser agrupados em sequências que possuam algum significado. Através de um exemplo simples, um bloco retangular com um furo passante retangular, é possível ilustrar a utilização dos Operadores de Euler (Figura 28).

A operação de extrusão translacional define um sólido a partir de um polígono plano. Este polígono plano será extrudado em uma direção segundo um comprimento fornecido.

3.3.2 Extrusão translacional

Esta rotina define uma aproximação poligonal de um arco de circunferência, a partir de um ponto fornecido. Assim, o algoritmo que implementa esta rotina pode ser definido utilizando-se apenas o Operador MEV (vide Figura 30). Para criar uma circunferência, será necessário definir um arco de 360° e o último Operador de Euler MEV deve ser substituído por um operador MEF (vide Figura 31).

3.3.1 Cria arco de circunferência

Figura 31. Finalização da criação de uma circunferência.

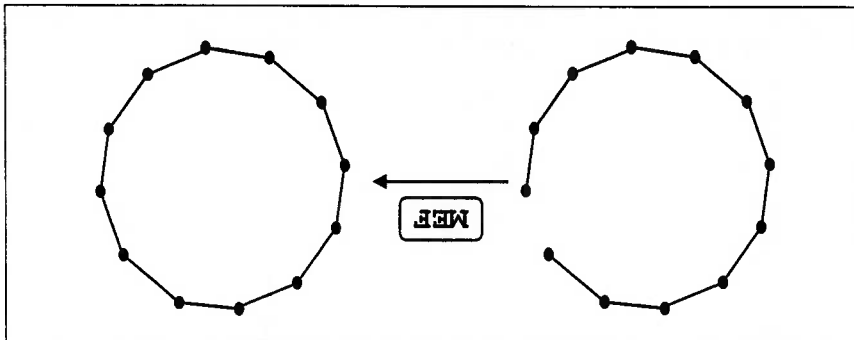


Figura 30. Sequência de Operadores de Euler que implementam a criação de um arco.

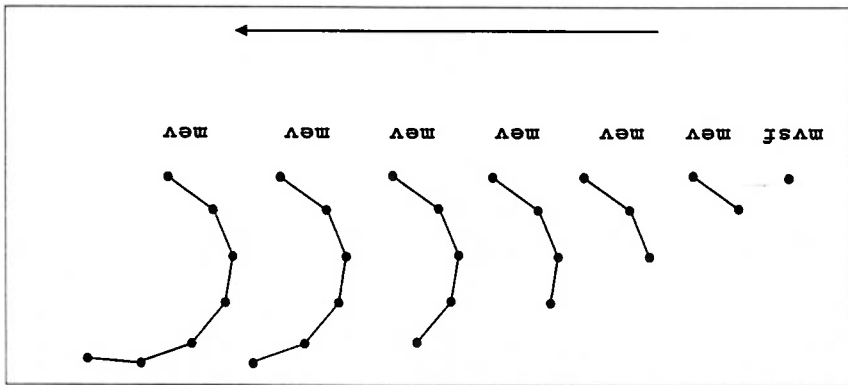


Figura 33. Sólidos gerados por extrusão rotacional.

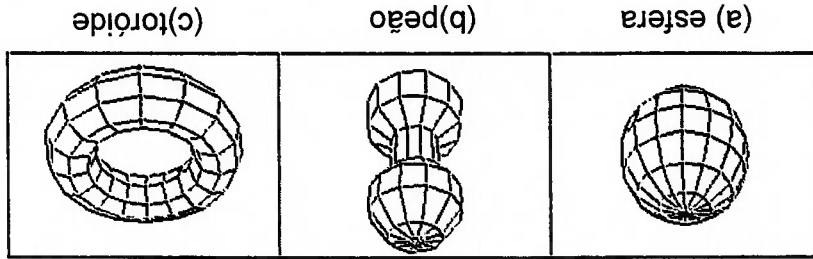


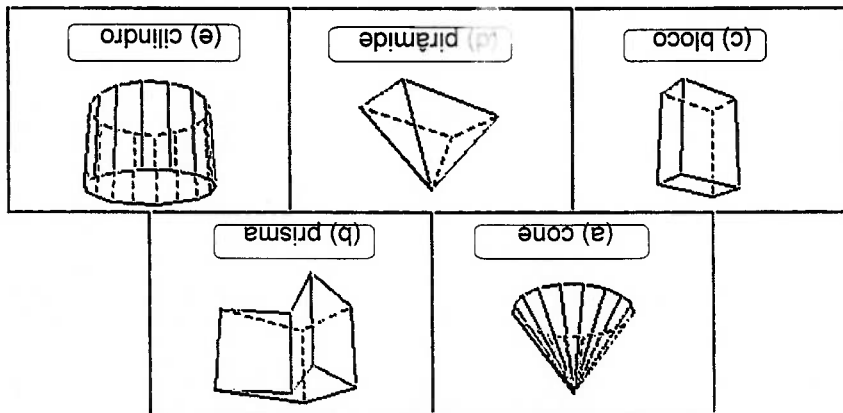
Figura 33).

Sólidos criados por meio deste operador são: a esfera, o péão de xadrez e o toróide (vide deverá ser com contorno aberto. Em caso negativo, ele deverá ser com contorno fechado. polígono será rotacionado. Caso o polígono interseccione o eixo de rotação, o polígono contorno aberto ou fechado. É necessário fornecer o eixo de rotação ao redor do qual a operação de extrusão rotacional define um sólido a partir de um polígono plano com

3.3.3 Extrusão rotacional

Como exemplo de sólido primitivo criado por meio deste operador temos o cubo (vide Figura 32 (c)), o cilindro (vide Figura 32 (e)) e o prisma (vide Figura 32 (b)). Se estiver disponível uma função que degeneere uma face em um ponto, será possível criarmos sólidos primitivos como o cone (vide Figura 32 (a)) e a pirâmide (vide Figura 32 (d)).

Figura 32. Sólidos obtidos por extrusão translacional.



4 Sincronização da Modelagem de Sólidos e Modelagem Geométrica

Neste capítulo apresentaremos a proposta feita por Turner [16], em que as superfícies são representadas como atributos de faces, e que inclui um mecanismo para subdividir a superfície em um conjunto de faces planas. Este mecanismo faz uso de Operadores de Euler. Também apresentaremos a proposta feita por Ueda [17] para determinar linhas poligonais que aproximam uma curva. Nesta proposta as curvas são representadas como atributos de arestas. Finalmente, entenderemos porque os dois mecanismos não são compatíveis entre si.

4.1 Propostas de Turner para Sincronizar Faces e Superfícies

A proposta de Turner [16], está baseada no uso de atributos para faces e arestas. Considere a Figura 34 que apresenta um cilindro aproximado por faces planas e arestas lineares. As faces planas que representam o topo e o fundo do cilindro são não "faces". Ou seja, estas faces não estão aproximadas e se mantêm na forma original. As

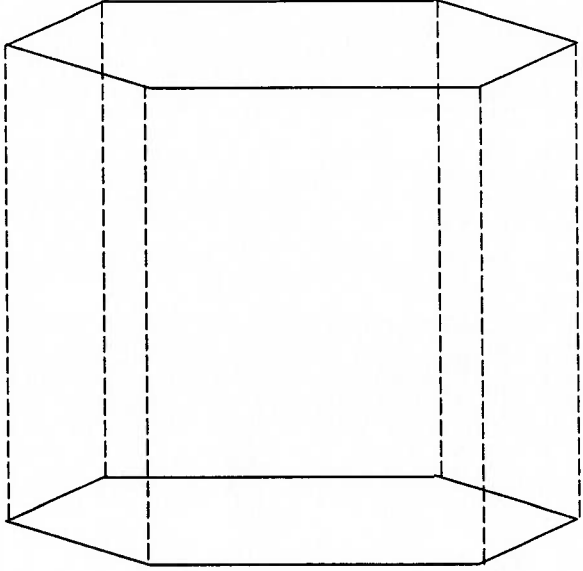


Figura 34. Um cilindro Aproximado por faces e arestas.

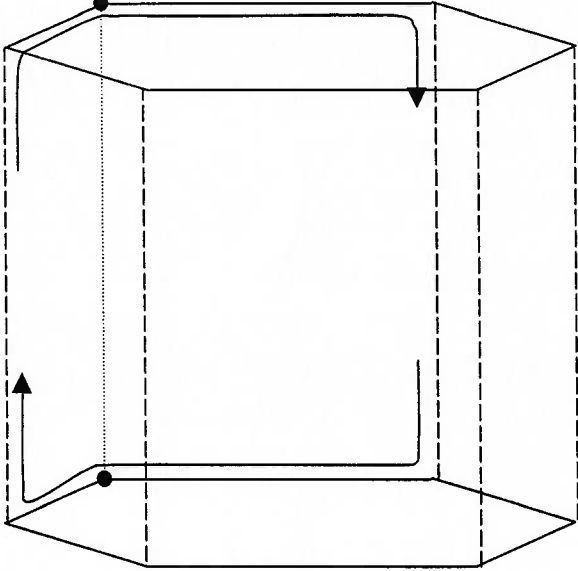


Figura 35. Cilindro depois da aplicação do operador KEF.

faces planas laterais são classificadas como "facet faces", pois são aproximações da superfície original. Cada "facet face", possui um ponteiro para a representação matemática exata da superfície que ela aproxima. As arestas também podem ser classificadas como "facet edge" e não "facet edge". As arestas classificadas como "facet edge" separam duas faces classificadas como "facet face". As demais arestas são classificadas como não "facet edge". Turner apresentou dois mecanismos: desenvolvimento que cria as faces planas aproximadoras; e enxugamento que remove as faces planas aproximadoras mantendo o número mínimo de elementos primitivos para manter o sólido topologicamente válido e consistente com a geometria associada.

A seguir ilustramos o algoritmo de enxugamento. A Figura 34 apresenta um cilindro totalmente desenvolvido com suas faces aproximadoras. Assim, ele possui seis "facet faces" e seis "facet edge". Na Figura 35 temos que uma face e uma aresta foram removidas pela aplicação do operador de Euler KEF. Na Figura 36 temos que um vértice e uma aresta superiores foram removidas pela aplicação do operador de Euler KEV. Finalmente, após a aplicação de cinco operadores KEF, onze operadores KEV e um operador KEVR obtemos o sólido apresentado na Figura 37.

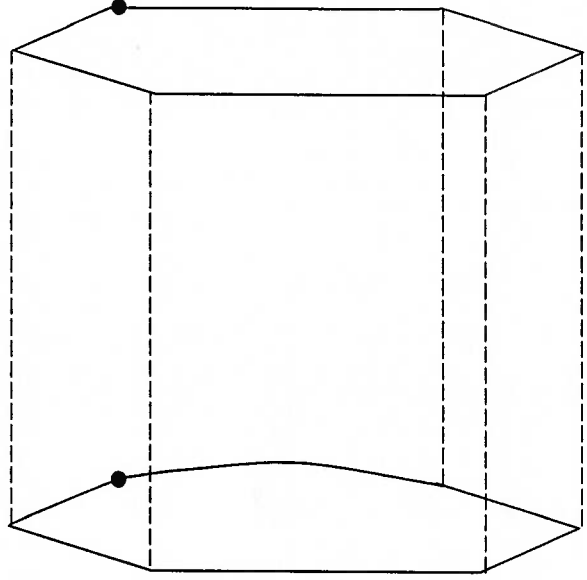


Figura 36. Cilindro depois da aplicação do operador KEV.

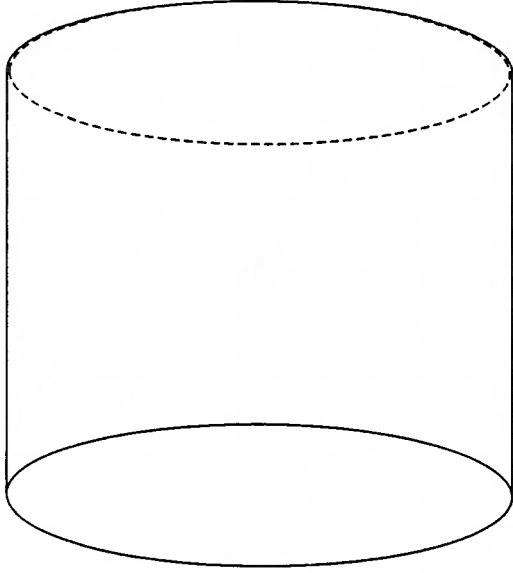


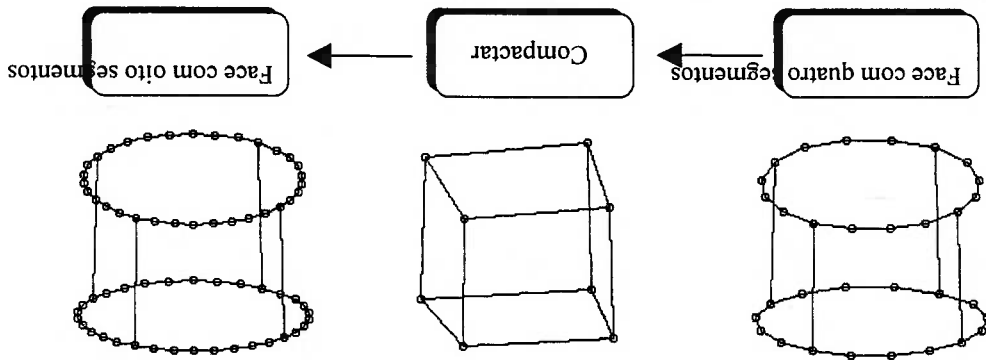
Figura 37. Cilindro final após o enxugamento.

É possível imaginar que em algumas situações será necessário aumentar o número de arestas aproximadoras ou reduzi-lo. A Figura 38 exibe um exemplo onde um cilindro é inicialmente criado com quatro arestas. Na figura intermediária, cada curva é associada a apenas uma aresta e na última figura a curva é aproximada por oito arestas. Neste caso precisamos de dois operadores, um para desenvolver a curva segundo um número de arestas e outro para compactar a curva.

Ueda [17] propôs dois mecanismos: desenvolvimento que cria arestas aproximadoras; e enxugamento que remove as arestas lineares aproximadoras mantendo o número mínimo de elementos primitivos, para manter o sólido topologicamente válido e consistente com a geometria associada.

4.2 Proposta de Ueda para Sincronizar Arestas e Curvas

pesquisas. situação ilustrada na figura 34. Foi pensando neste problema que Ueda [17] iniciou suas pesquisas. O conceito apresentado por Turner é que durante o algoritmo de enxugamento, as arestas classificadas como "facet edge" devem ser removidas pela aplicação do operador de Euler KEF. Entretanto, Turner não comenta sobre como decidir se uma aresta classificada como não "facet edge" deve ser removida ou não, que é exatamente a situação ilustrada na figura 34. Foi pensando neste problema que Ueda [17] iniciou suas pesquisas.



- **ShrCur (Shrink Curve):** O operador de enxugamento remove as arestas de aproximação da curva deixando apenas uma única aresta associada à curva. A Figura 40 ilustra a aplicação deste operador que é implementado com uma sequência de operadores KEV. Antes que um operador de desenvolvimento seja utilizado é necessário aplicar o operador de enxugamento, pois o operador de desenvolvimento assume que existe uma única aresta associada à curva a ser desenvolvida.
- **Develop Curve:** este operador de alto nível é utilizado para criar um conjunto de arestas que aproximam a curva associada a uma aresta específica. Este operador calcula pontos de aproximação para a curva e aplica operadores MEV (Make Edge Vertex). Um ponto importante neste operador é identificar o correto sentido da curva quando ocorrer a aplicação dos operadores MEV para que a curva não seja criada de modo invertido. Como, para uma aresta temos duas meia-arestas, é necessário verificar qual meia-aresta é a mais adequada para efetuar o desenvolvimento da curva. Esta verificação é feita pela comparação entre os vértices da arestas e os pontos de controle da curva. Para toda nova aresta criada, associa-se a aresta à curva que ela aproxima. Este relacionamento permitirá identificar quais são as arestas que aproximam uma curva. A Figura 39 ilustra a aplicação deste operador.

Figura 39. Operador de alto nível Develop Curve.

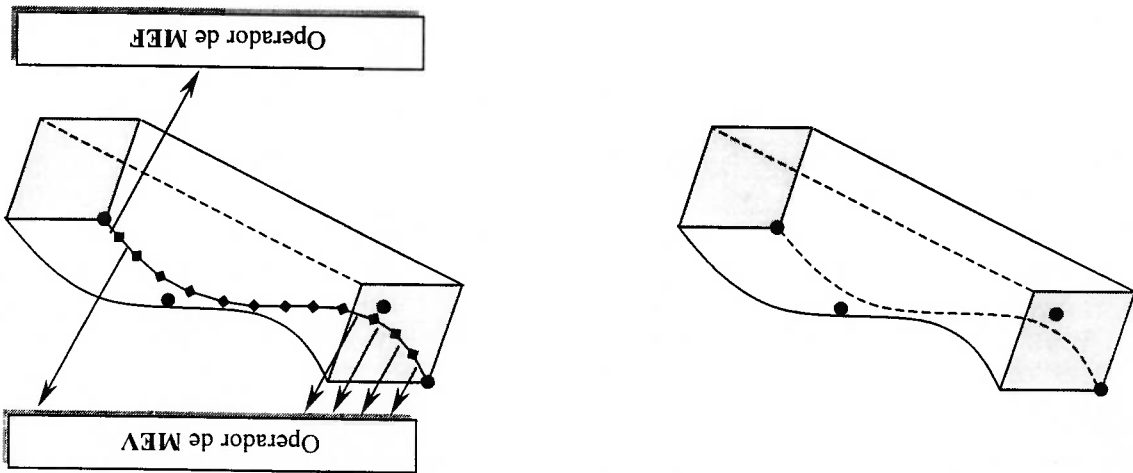
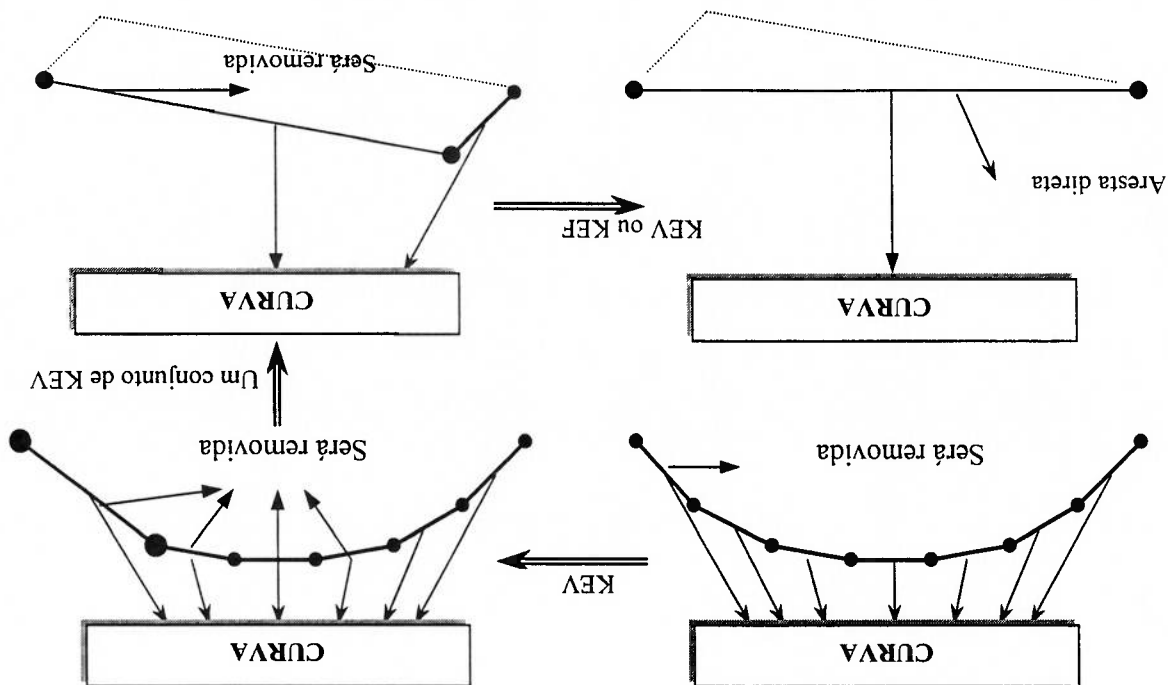


Figura 40 Remove as arestas de aproximação de curva.



Uma possibilidade para solucionar este problema seria enxugar todas as arestas aproximadoras para que as faces planas que aproximam as superfícies possam ser determinadas. Após a determinação das faces planas, a propriedade integral pode ser

solicitar o valor de uma propriedade integral.

O sólido com superfícies aproximadas por faces planas será utilizado para calcular propriedades integrais de massa. O sólido com curvas aproximadas por arestas lineares é utilizado para exibir o sólido de modo wire-frame como na Figura 39. Uma possibilidade inconsistente será se o usuário estiver observando um sólido em modo wire-frame e

significa que o algoritmo deve ser terminado.

aproximada. O operador de enxugamento identifica as arestas vizinhas que possuem a mesma curva associada. Se uma aresta vizinha com a mesma curva associada for encontrada, ela deve ser removida pelo operador KEV. Caso nenhuma aresta vizinha com a mesma curva associada seja encontrada,

calculada. Finalmente, ocorre o enxugamento das faces planas aproximadoras para que as arestas aproximadoras sejam criadas a fim de que o usuário possa visualizar o sólido em modo wire-frame.

Uma segunda solução é observar que as arestas classificadas como não "facet edge" são arestas lineares que aproximam as curvas existentes no modelo. Assim, para exibir em modo wire-frame bastaria varrer todas as arestas existentes no sólido com as superfícies aproximadas por faces planas e exibir apenas as arestas classificadas como não "facet edge". Entretanto, desta maneira, o nível de detalhamento de aproximação das curvas seria o mesmo que o nível de detalhamento da aproximação das superfícies, e isto não é desejável.

O nível de detalhamento para a aproximação das curvas está associado à razão da dimensão da curva sobre a dimensão da câmera [12]. A figura 41 ilustra um exemplo de câmera, onde é possível observar o olho que é o centro de projeção, o anteparo onde o sólido será projetado e uma curva que foi projetada sobre o anteparo. A curva é exibida como uma sequência de linhas, onde $e/p \leq \epsilon$, (ϵ é uma constante). Assim, a curva terá a

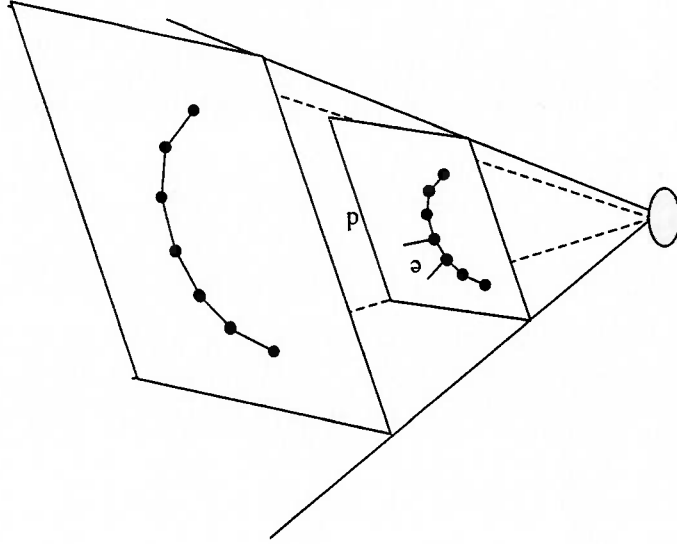


Figura 41. Uma curva projectada.

aparência de curva e não de uma sequência de linhas. O nível de detalhamento para a aproximação das superfícies está associado à dimensão da superfície sobre a tolerância suportável pelo cálculo da propriedade integral. Deste modo, o ideal é que ambos os níveis de detalhamento sejam independentes. Assim, neste ponto é que iniciamos as nossas pesquisas.

5 Sincronizando Modelagem de Sólidos e Modelagem

Geométrica

Neste trabalho, foi utilizada a estrutura meia-aresta proposta por Mäntylä [11]. A estrutura é apresentada na Figura 42. Neste capítulo apresentaremos a nossa proposta para suporte ao sincronismo entre modelagem de sólidos e modelagem geométrica. Inicialmente discutiremos alguns requisitos que tal proposta deve atender. Em seguida apresentaremos em maior profundidade a proposta apresentada por Ueda [17], incluindo: estrutura de dados operadores básicos e operadores locais. Em seguida apresentaremos a nossa proposta para representar superfícies em modeladores de sólidos B-Rep,

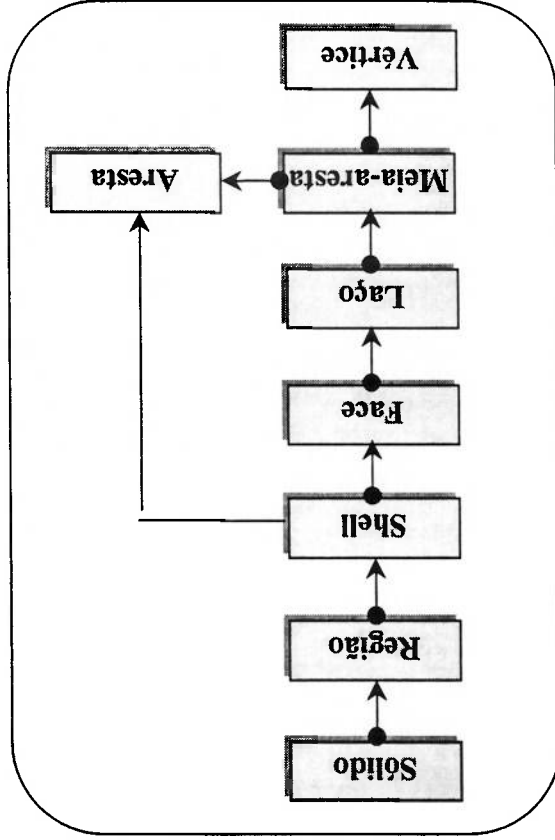
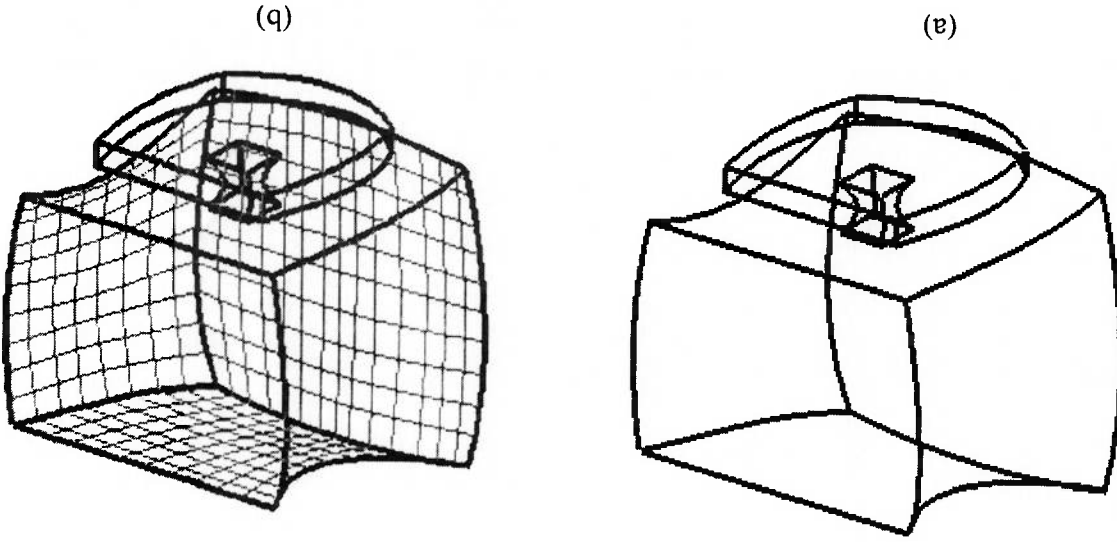


Figura 42. Estrutura meia-aresta.

Figura 43 Modelo curva e modelo com superfícies.

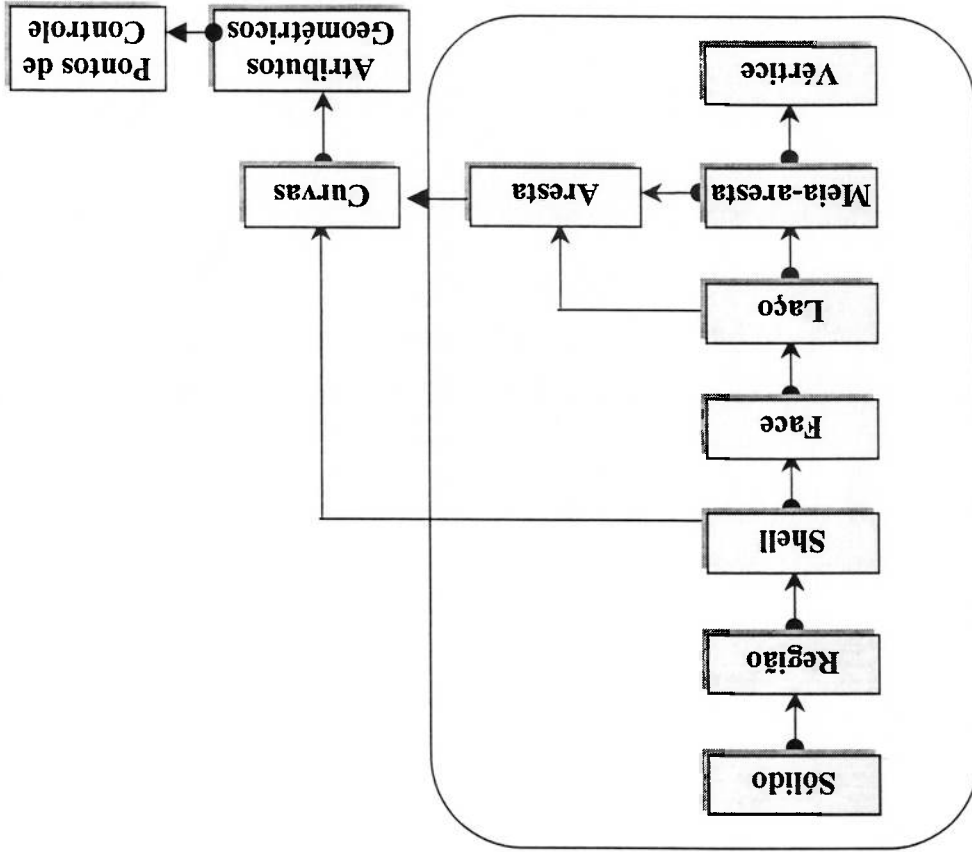


Como veremos neste capítulo, a solução para o problema de associar faces poligonais e superfícies consiste em definir um novo mecanismo para associar a geometria das superfícies às faces poligonais. Isto será feito pela ampliação da estrutura de dados

aproximadoras estão associada à curva que foi modificada?
 diversas faces planas aproximadoras. Como devemos identificar quais faces associadas à superfície que foi modificada para então atualizá-las. Veja o modelo exibido na Figura 43, cada curva está associada a duas superfícies, e cada superfície possui coordenadas adequadas. O modelador de sólidos deve saber quais faces poligonais estão decide modificar um ponto de controle, selecionando-o e posicionando-o segundo as conjunto de informações em arestas e polígonos. Após uma certa análise, o usuário controle para uma superfície, em seguida, o modelador de sólidos transforma este homem-máquina. Considere o seguinte exemplo: o usuário define o conjunto de pontos de aproximações lineares e poligonais, ainda permanecem vários problemas de interação Apesar de ser possível representar qualquer curva ou superfície por meio de suas

incluindo: estrutura de dados, operadores básicos e operadores locais.

Figura 44 Estrutura Hierárquica para representar curvas.

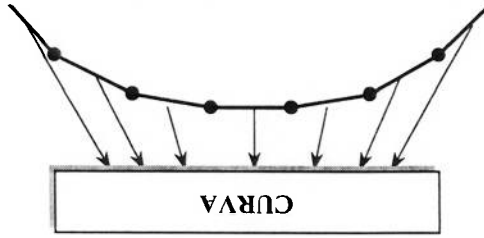


Mantyla [11] e Toriya e Chiyodura [14] representam curvas como atributos de arestas. Ueda [17] propôs utilizar o mesmo princípio e adaptou o mecanismo de desenvolvimento e enxugamento de superfícies proposto por Turner [16], para ser aplicado às arestas. Assim, o principal objetivo desta proposta é permitir que curvas sejam exibidas e controladas como se fossem seqüências de arestas lineares. A estrutura de dados

5.1 Estrutura de Dados para representar Curvas

utilizada no modelo básico, definindo novos conceitos para representar o modelo sólido e modificando os algoritmos para que o modelador de sólidos trabalhe corretamente. Apresentaremos também alguns operadores que foram definidos para encapsular e facilitar a utilização destes novos conceitos.

Figura 46. Arestas com ponteiros para sua curva.

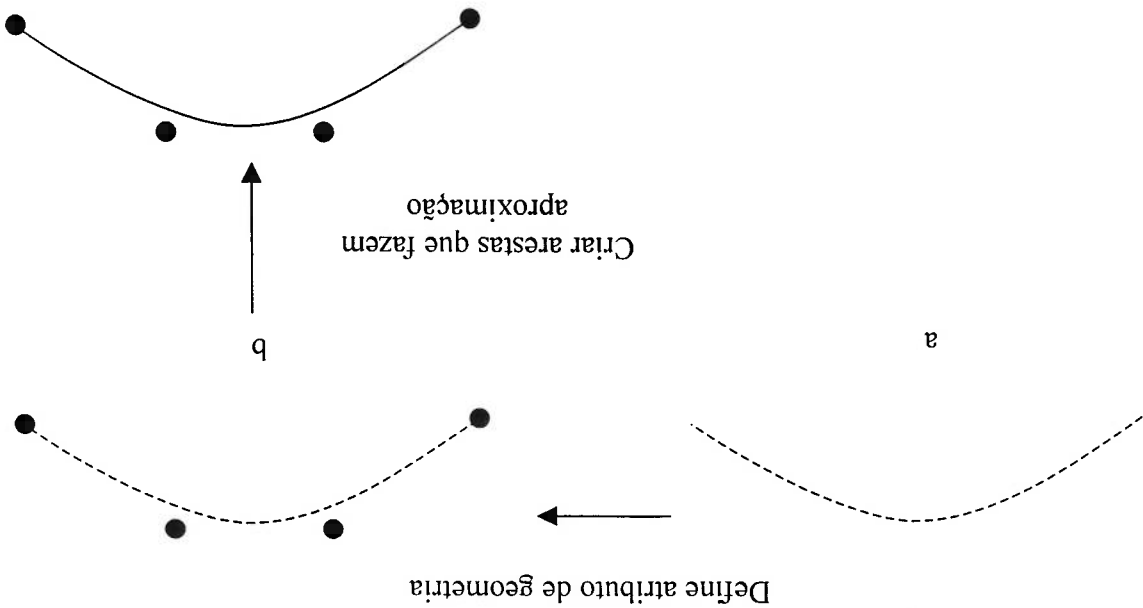


Segundo esta representação, para representarmos uma curva de Bézier, é necessário inicialmente definir os seus pontos de controle, em seguida um operador determina o valor das coordenadas dos vértices das arestas que aproximam a curva de Bézier. Ao modificarmos um ponto de controle da curva de Bézier, é relativamente simples definir um algoritmo que atualize os valores das coordenadas dos vértices das arestas que aproximam a curva de Bézier (vide Figura 45).

Figura 44.

adaptada para representar curvas em modeladores de sólido B-Rep esta ilustrada na

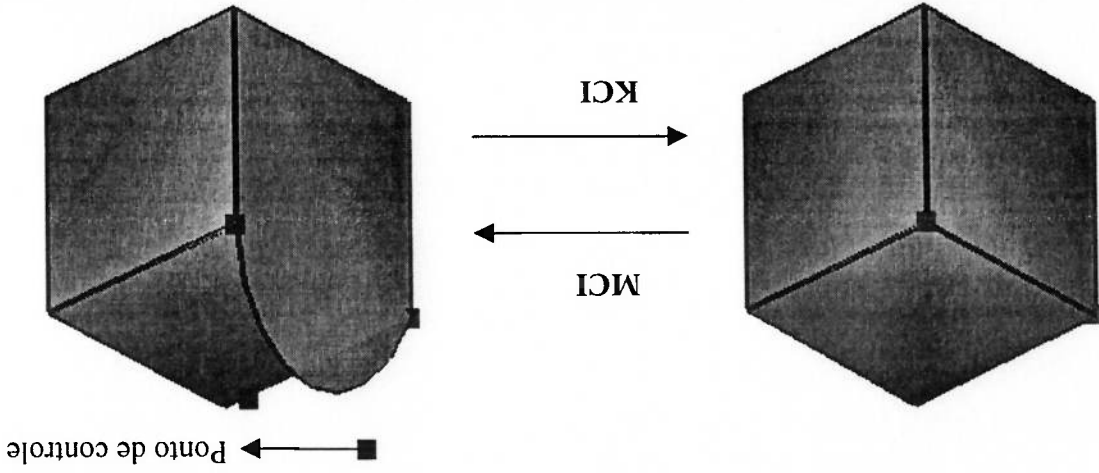
Figura 45. Processo que cria arestas de aproximação de uma curva.



Define atributo de geometria

Criar arestas que fazem aproximação

Figura 47. Operações para converter uma aresta reta em uma curva e a sua inversa.



- **ADPP (Add Point to Polyline):** acrescenta um ponto de controle à curva.
- **REMP (Remove Point from Polyline):** remove um ponto de controle da curva.
- **MCI (Make Curve Information):** é uma operação para associar uma curva a uma aresta.
- **KCI (Kill Curve Information):** é a operação inversa do MCI que remove a curva.

É conhecido que os Operadores de Euler podem criar e remover elementos topológicos como faces, arestas e vértices dos sólidos. Os operadores de Euler apresentados no capítulo anterior não permitem modificar características geométricas, assim será necessário definir novos operadores de Euler para este fim (por exemplo, a operação para converter uma linha reta em uma curva). Os operadores definidos são os seguintes :

5.2 Operadores de Euler para Manipular Curvas

Esta estrutura associa a cada meia-aresta um atributo de curva, de modo que elas possam ser aproximadas por uma sequência de meia-arestas no circuito de contorno do laço definindo uma lista ligada de modo que o seu acesso e a definição de algoritmos ficam facilitados.

do sólido. A Figura 47 mostra um exemplo de remoção de uma curva de superfície).

- **MCP (Move Control Point):** é uma operação para trasladar o n-ésimo ponto de controle de uma curva por um vetor. A operação inversa também é MCP (vide Figura 48). Um fato é que os vértices da aresta correspondem a vértices sobre a curva associada a esta aresta. Assim, existe uma duplicação de informação que necessita ser mantida de modo consistente. Como os operadores de Euler trabalham em um nível básico, a consistência destas informações é garantida pela existência de um operador de nível mais elevado que garante que ao ser feita uma modificação no vértice da aresta, a mesma modificação deve ocorrer para o ponto de controle e vice-versa. A Figura 49 ilustra este tipo de operador em funcionamento, ambos as vértices da aresta e da curva foram movimentados. Para verificar se todo o sistema está funcionando corretamente será utilizada uma função que verifica se as informações entre arestas e curvas estão consistentes.

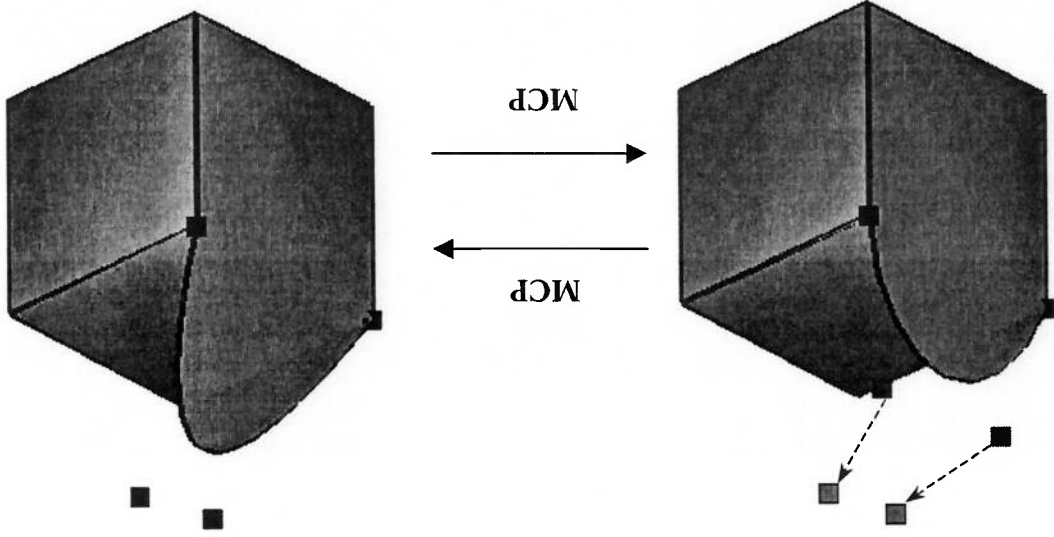
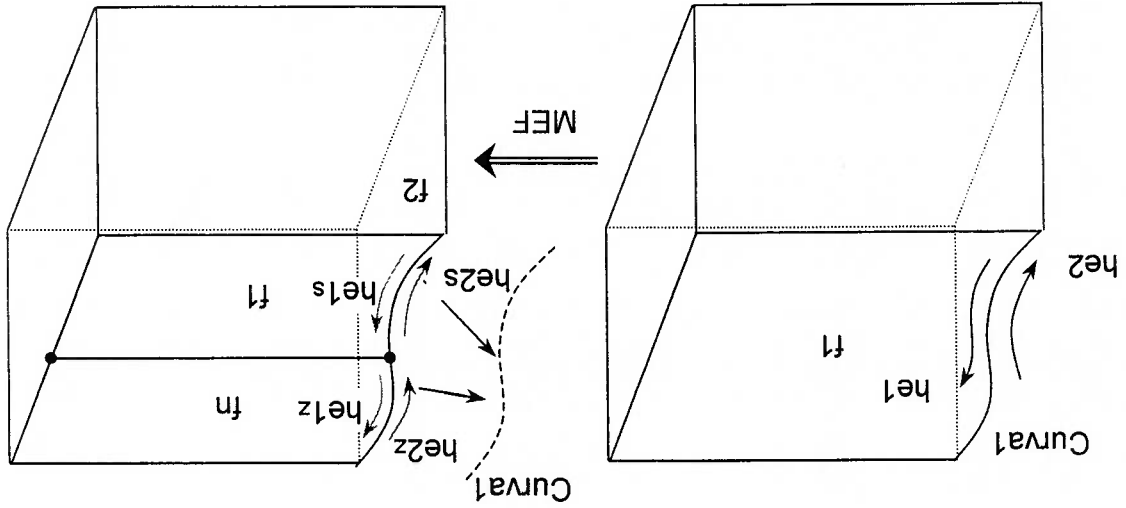


Figura 48 Operação para trasladar o n-ésimo ponto de controle de uma aresta curva por um vetor e a sua inversa.

Figura 50 Associação de curvas as meia-arestas, exemplo de aplicação do operador MEF.



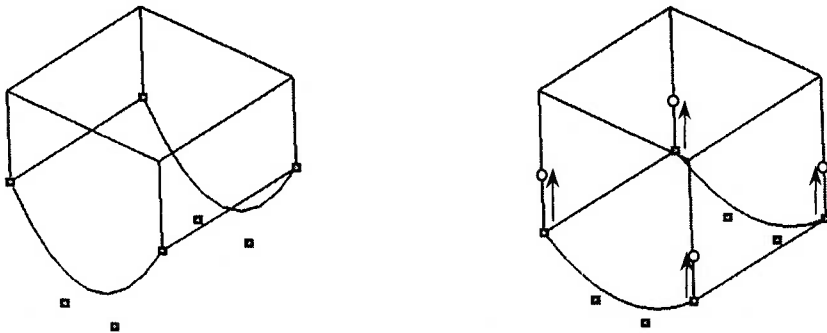
A face f1 foi dividida em duas, e a aresta que contém as meia-arestas he1 e he2 foi dividida em duas. As meia-arestas he1s e he1z estão associadas às novas curvas. A união

operador MEF foi aplicado.

Devemos ver, então, em qual situação isto não ocorre. Considere a Figura 50, em que o caso, as duas meia-arestas de uma mesma aresta possuem a mesma curva associada. Nenhum autor propôs associar as curvas às meia-arestas. Em verdade, todos comentam que as curvas devem ser associadas às arestas. Talvez, porque na maioria dos

5.3 Curvas Como Atributos de Meia-arestas

Figura 49 Rejustamento quando pontos de controle sobre os vértices modificados.



quatro lados.

tenha n lados ($n \neq 4$), então a superfície associada a esta face terá n faces, cada uma com associada a esta face terá apenas uma face. Em outra situação, onde a face do sólido vértices. Caso a face do sólido tenha quatro arestas como contorno, então a superfície atribuído da superfície pode conter uma ou mais faces que contêm laços, meia-arestas e possua um atribuído de superfície contendo as informações sobre a geometria da face. O A Figura 51 ilustra a estrutura de dados proposta. Estamos propondo que a face

conflicto entre representar a topologia do modelo e representar a geometria do modelo. uma face com mais que quatro lados em várias superfícies de quatro lados. Este é o a superfície tenha quatro lados. Assim, por limitações matemáticas, é conveniente dividir Assim, um fator limitante é a representação geométrica das faces, que requer que

abertura pode ter sua geometria associada diretamente.

Qualquer abertura pode ser fechada topologicamente por uma face. Entretanto, nem toda geometria de fechamento do sólido. Aparentemente estas duas funções são idênticas. o fechamento do sólido, topologicamente falando. A segunda função é representar a exerce para facilitar a definição de novos algoritmos. A primeira função é o de representar 42. A nossa principal proposta é desenvolver duas principais funções que o elemento face Inicialmente discutiremos sobre a função do elemento face apresentado na Figura

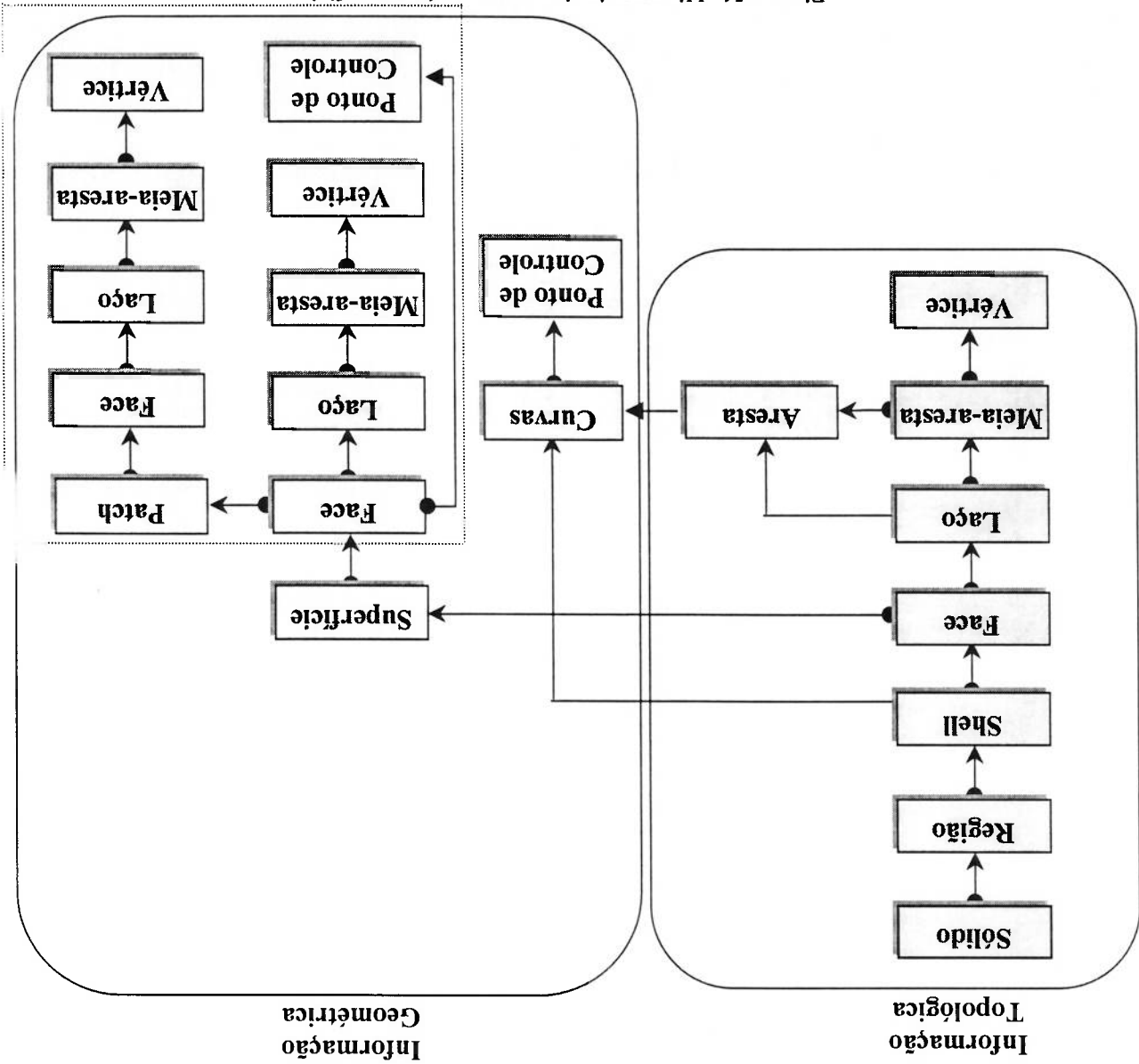
5.4 Estrutura de Dados para Representar Superfícies

estaremos minimizando este efeito.

isto, criaremos novas faces com cinco lados. Assim ao associarmos a curva à meia-aresta, deve ser dividida para que todas as suas subfaces tenham quatro lados, e ao fazermos obrigatoriamente dividida e cinco curvas estariam definindo a face f_2 . Neste caso a face f_2 fosse associada à aresta, como proposto pelos outros autores, a curva original seria contamos o número de curvas que definem a face f_2 , teremos quatro curvas. Se a curva associadas à curva original, como se nenhuma divisão tivesse ocorrido. Assim, ao destas duas curvas define a curva original. Mas as meia-arestas h_{2z} e h_{2s} estão

Deste modo fica a pergunta, qual a diferença entre as faces de um sólido e as faces da superfície? As faces do sólido podem ter qualquer número de lados, e representam as modificações feitas pelo usuário. Ou, seja as intenções do usuário estão representadas nesta estrutura. As faces da superfície possuem todos os quatro lados, e simultaneamente, todas possuem a sua geometria representada. Assim, caso a face da estrutura original do sólido não possua quatro lados, ela é dividida por um algoritmo de modo que surja um número conveniente de faces para que a sua geometria possa ser definida. Deste modo,

Figura 51. Hierarquia da estrutura de superfícies.



as faces da estrutura superficial nem sempre representam as intenções do usuário, pois em sua maioria o modo de realizar a divisão, e o algoritmo de atribuir geometria são automáticos. Conforme explicamos anteriormente sobre a divisão e atribuição de geometria para os casos em que as faces possuem 3 e 5 lados.

Existe uma terceira estrutura denominada patch que está associada a cada face da estrutura superficial. Esta terceira estrutura existe para criar uma aproximação poliedral da geometria definida na estrutura superficial. Deste modo é possível criar algoritmos que manipulam aproximações poliedrais para calcular propriedades de massa e visualizadores tridimensionais, uma vez que nem todas as superfícies podem ser visualizadas diretamente em aceleradores gráficos tridimensionais, como o OpenGL [19]. Em nosso caso particular, as superfícies de Gregory devem ser aproximadas, por um conjunto de superfícies poliedrais para que possam ser visualizadas. A Figura 52 ilustra um exemplo de patches que aproximam uma superfície.

Deste modo, a face do sólido não possui geometria associada, mas possui o seu contorno definido. Ou seja, todas as curvas de seu contorno já estão definidas. As faces da estrutura superficial possuem geometria definida e todas possuem quatro lados. A face da estrutura patch representa a aproximação poliedral da geometria definida na estrutura

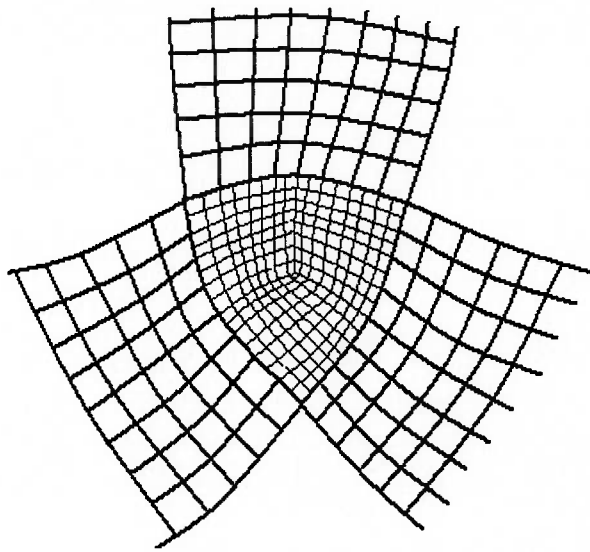


Figura 52 Exemplo de superfícies desenvolvidas.

• **ADDPs (Add Point to Surface):** acrescenta um ponto de controle a uma superfície. Por exemplo, uma superfície de Bézier tem 16 pontos de controle, inicialmente o contorno da superfície é definido (12 pontos de controle). Para completar o definição da superfície, é necessário acrescentar os 4 pontos de

5.5.1 Operador de Euler

totalmente compatível.

Para manipular a informação de geometria no modelador de sólidos, desenvolvemos alguns operadores que podem construir e modificar superfícies. Eles serão descritos neste capítulo. Apesarda os operadores de baixo nível serem bem semelhantes aos operadores de Euler originais do Modelador de Sólidos, tivemos que implementar um novo conjunto, que apresentamos a partir de agora, pois a funcionalidade necessária não é

5.5 Operadores de Euler para Manipulação de Superfícies

Deste modo, separamos o mecanismo de detalhamento de curvas do mecanismo de detalhamento de superfícies, sendo possível que ambas sejam detalhadas com níveis distintos, conforme a real necessidade (Vide Figura 53).

superfície.

Figura 53 Uma face triangular associadas pelo três "patch".

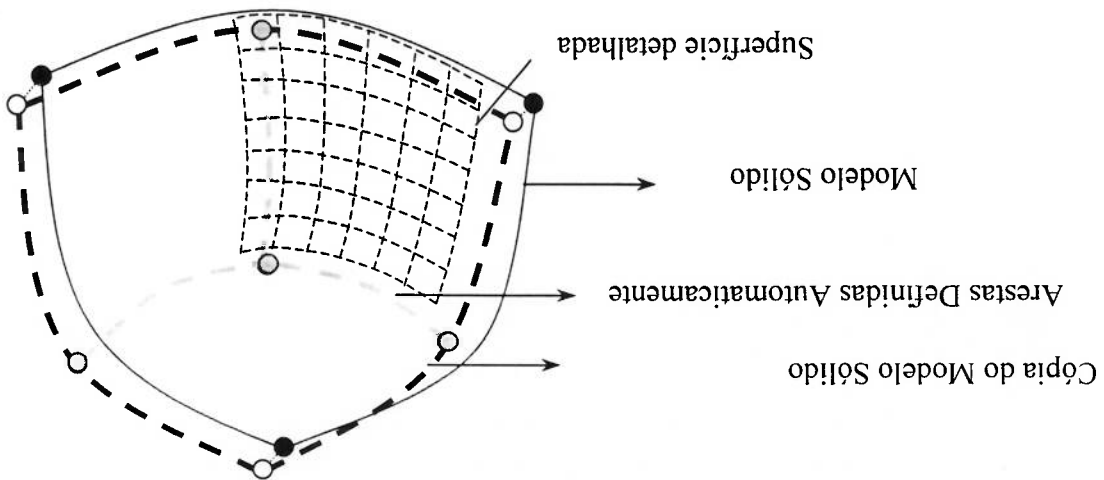


Figura 55 A forma da superfície mudou depois que os ponto de controle foram modificados.

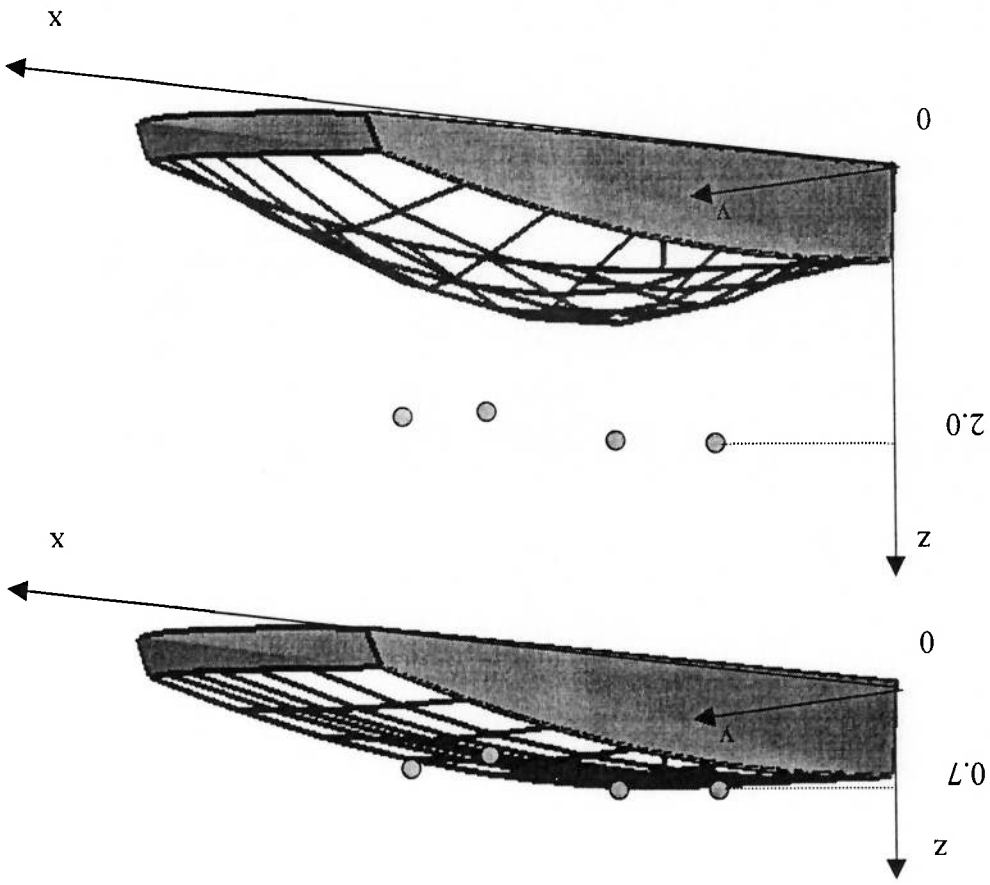
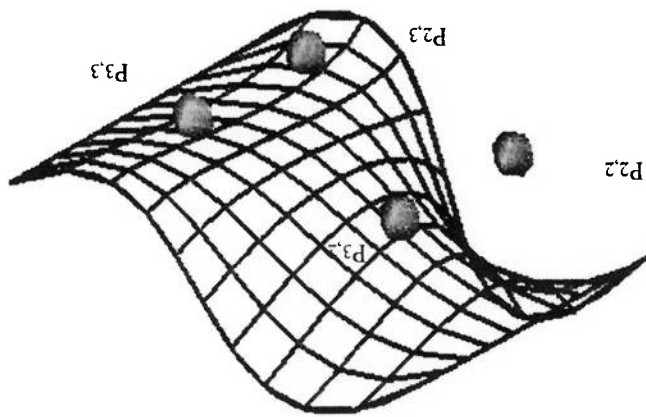


Figura 54 Exemplo de superfície de Bézier.



controle internos por este operador. A figura 54 ilustra que os pontos de controle P_{2,2}, P_{2,3}, P_{3,2} e P_{3,3} foram acrescentados para completar a definição da superfície.

5.5.2 Operadores de Baixo Nível na Estrutura Superfície

- **REMPs (Remove Point from Surface):** é o operador inverso ao ADPFS. A uma face A Figura 54 mostra uma superfície que foi definida e acrescentada a uma face.
- **KSI (Kill Surface Information):** é a operador inverso ao MSI que remove a superfície do sólido.
- **TSV (Transform Surface form to a Vector):** é o operador para transformar um ponto de controle de uma superfície. A Figura 55 ilustra um exemplo de aplicação deste operador.

- **MSuVF (Make Surface Vertex Face):** é um operador semelhante a MVSF (Make Vertex Solid Face). Ele é utilizado para definir o primeiro vértice e a primeira face na estrutura superfície.
- **MSUEV (Make Surface Edge Vertex):** operador semelhante a MEV (Make Edge Vertex). Ele é utilizado para definir uma aresta e um vértice na estrutura superfície.
- **MSUEF (Make Surface Edge Face):** operador semelhante a MEF (Make Edge Face). Ele é utilizado para definir uma aresta e uma face na estrutura superfície (vide figura 56).
- **KSUVF (Kill Surface Vertex Face):** operador semelhante a KVSF (Kill Vertex Solid Face). Ele é utilizado para remover o último vértice e a última face da estrutura superfície.
- **KSUEV (Kill Surface Edge Vertex):** operador semelhante a KEV (Kill Edge Vertex). Ele é utilizado para remover uma aresta e um vértice da estrutura superfície.
- **KSUEF (Kill Surface Edge Face):** operador semelhante a KEF (Kill Edge Face). Ele é utilizado para remover uma aresta e uma face da estrutura superfície.

Estes operadores são conjuntos de operadores de Euler. Alguns são apresentados abaixo:

- **CopyFace**: Este operador constrói uma nova face a nível de superfície idêntica à que existe ao nível do modelo sólido. Primeiro, um Operador de Euler MSUVF

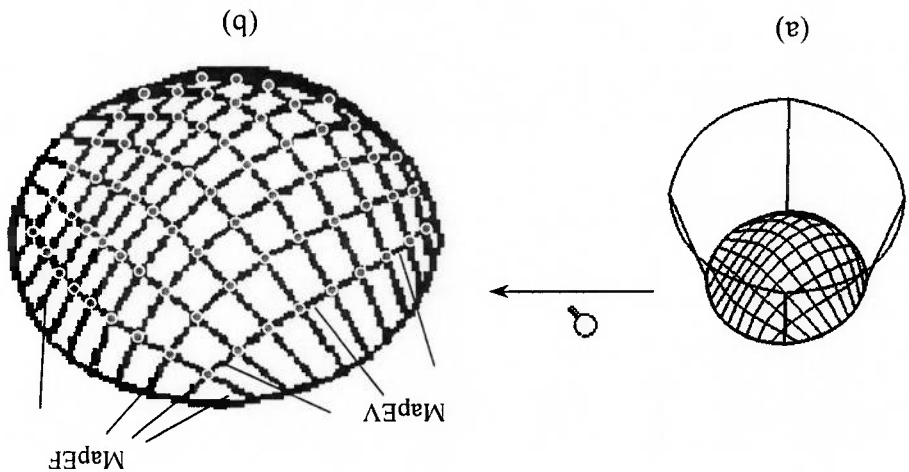
5.5.4 Operadores de Alto Nível

- **KAPEF (Kill Surface Edge Face)**: é um operador semelhante a KEF (Kill Edge Face). Ele é utilizado para remover uma aresta e uma face da estrutura patch.
- **KAPEV (Kill Surface Edge Vertex)**: é um operador semelhante a KEV (Kill Edge Vertex). Ele é utilizado para remover uma aresta e um vértice da estrutura patch.
- **KAPVF (Kill Surface Vertex Face)**: é um operador semelhante a KVSF (Kill Vertex Solid Face). Ele é utilizado para remover o último vértice e a última face da estrutura patch.
- **MAPVF (Make Surface Vertex Face)**: é um operador semelhante a MVSF (Make Vertex Solid Face). Ele é utilizado para definir o primeiro vértice e a primeira face na estrutura patch.
- **MAPEV (Make Surface Edge Vertex)**: é um operador semelhante a MEV (Make Edge Vertex). Ele é utilizado para definir uma aresta e um vértice na estrutura patch.
- **MAPEF (Make Surface Edge Face)**: é um operador semelhante a MEF (Make Edge Face). Ele é utilizado para definir uma aresta e uma face na estrutura patch (vide figura 57).
- **MAPVF (Make Surface Vertex Face)**: é um operador semelhante a MVSF (Make Vertex Solid Face). Ele é utilizado para definir o primeiro vértice e a primeira face na estrutura patch.
- **MAI (Make Approximation Information)**: é o operador para acrescentar a informação de aproximação a uma face na estrutura patch.
- **KAI (Kill Approximation Information)**: é o operador para remover a informação de aproximação de uma face na estrutura patch.

Implementamos um conjunto de operadores de baixo nível para manipular a estrutura patch.

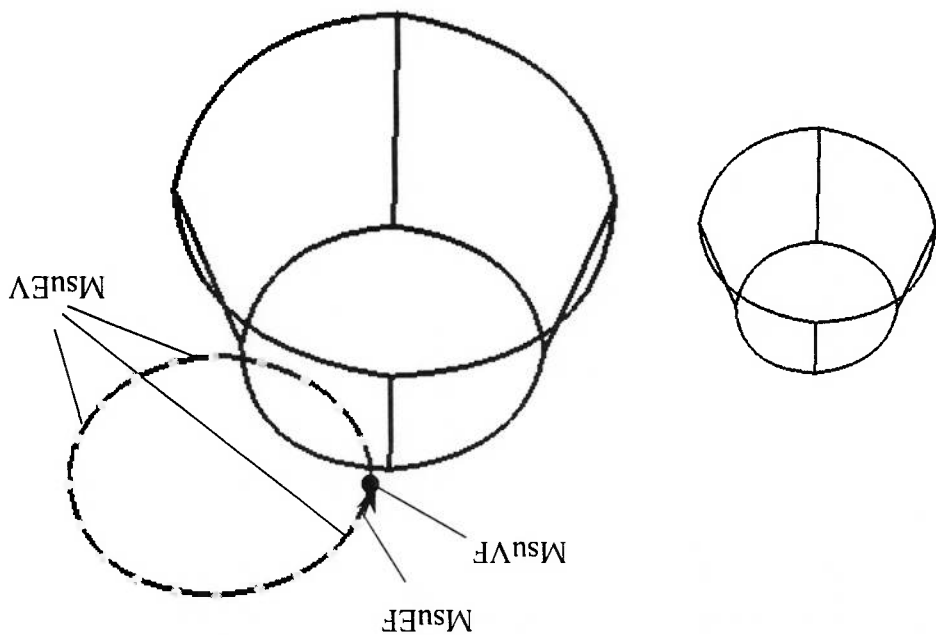
5.5.3 Operadores de Baixo Nível na Estrutura Patch

Figura 57 Criação da aproximação poliedral.



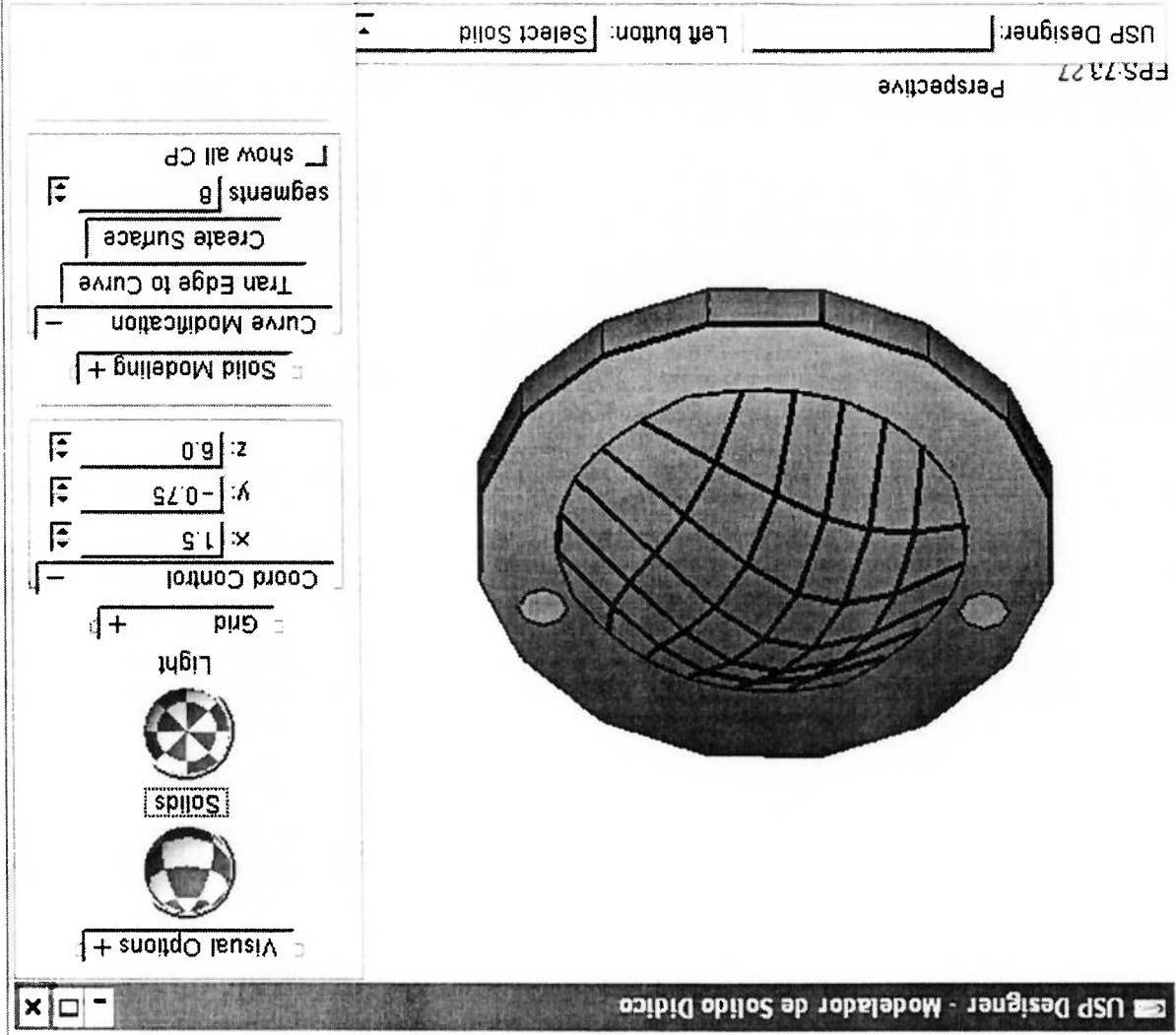
- **Render:** Cria um malha de poligonos que aproximam a superficie, facilitando a sua visualização bem com o cálculo de propriedades de massa (figura 57 (a)).
será executado para construir um novo sólido na superficie, depois utiliza MSUEV e MSUEF para criar uma nova face com as mesmas informações de topologia (Figura 56) .

Figura 56. Operador de alto nível copyFace.



- Inicialmente as curvas de contorno são definidas. Segundo, os pontos internos são calculados de acordo com a equação (17) para em seguida serem aplicados os operadores MsueV e MsueF (Figura 57 (b)).
- TransformThreeSidedIntoFourSidedFaces: Este operador cria três faces com quatro lados segundo o algoritmo proposto por Chiyokura e Kimura [4]).

5.6 Considerações Finais



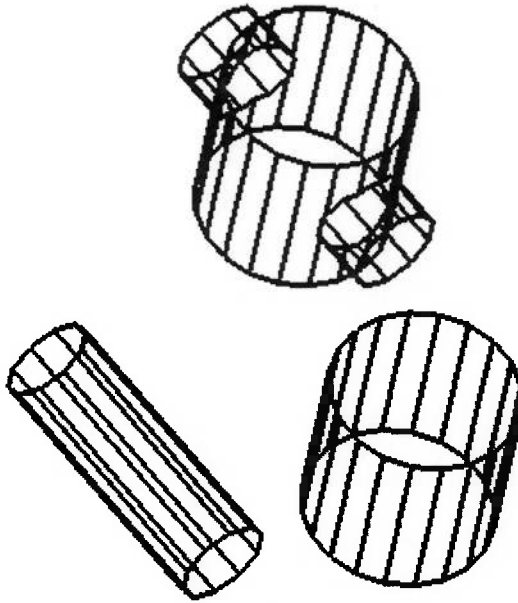
(a). Interface em USPDesigner para modificar superfícies.

O USPDesigner dispõe de um ambiente de visualização e interação com os objetos criados. E nesse ambiente aciona os comandos e analisa o resultado das funções. O OpenGL está sendo utilizado para meio das bibliotecas GLUT e GLUI. O OpenGL está sendo utilizado para exibição e seleção dos objetos. Atualmente o USPDesigner está sendo compilado no ambiente Windows utilizando o Visual C++ 6.0. A Figura xx mostra a tela do USPDesigner. As funções do USPDesigner podem ser acionadas por duas maneiras: por linha de comando ou por menu.

Todos os algoritmos apresentados foram implementados e acrescentados ao modelador de sólidos USPDesigner, que está sendo desenvolvido no Laboratório de Geometria Computacional do Departamento de Engenharia Mecatrônica da EPUSP (Wang, 2003).

Figura 58.Exemplos em USPDesigner.

(b). Operação Booleano em USPDesigner.



6 Conclusão

Neste trabalho apresentamos uma estrutura com três níveis de detalhamento para suportar simultaneamente e de forma independente o mecanismo de criar arestas lineares que aproximam uma curva e o mecanismo de criar faces planas que aproximam uma superfície. Ambas baseadas nas propostas apresentadas por Ueda [17] e Turner [16].

Este conceito se baseia principalmente no fato de que a face representada no modelo sólido possui duas funções: representar o fechamento topológico do sólido e representar algumas situações é necessário dividir a face de forma conveniente para que a geometria seja associada. Assim, com a estrutura de dados proposta, separamos os elementos que foram criados automaticamente para satisfazer esta necessidade, dos elementos criados satisfazendo as intenções do usuário.

Nesta nova estrutura, propusemos que a curva seja associada à meia-aresta. Deste modo evitaremos a necessidade de provocar divisões em cascata. A figura 59(a)



Figura 59 (a) Um sólido com superfícies

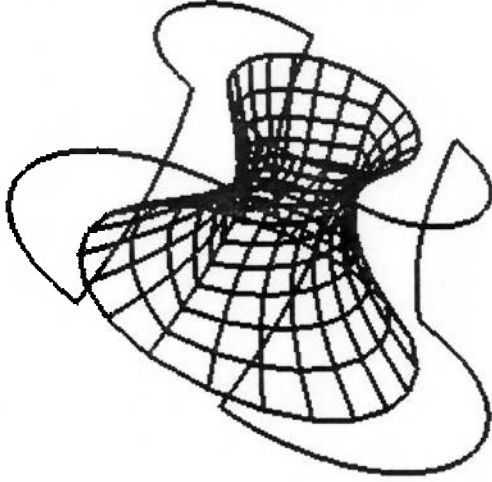


Figura 59 (b) Contorno do sólido e superfícies

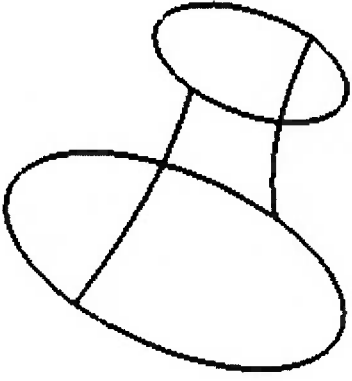


Figura 59 (c) Contorno do sólido

apresenta um exemplo de superfície desenvolvida com nível de detalhamento distinto do aplicado para as curvas (vide Figura 59 (c)).

Como sugestão de trabalho futuro temos as pesquisas sobre as implicações de utilizar esta estrutura com Operadores Booleanos. Realmente, é uma grande tarefa de pesquisa para representar sólido resultante da Operadores Booleanos representado com sua geometria exata. Neste caso, será necessário também criarmos algoritmos para determinar a curva de intersecção entre superfícies.

Referência

- [1] P.E. Bezier, *Numerical Control-Mathematics and Applications*, John Wiley and Sons, 1972.
- [2] H.Chiyokura, *An Extended Rounding Operation for Modeling Solids with Free-Form Surfaces*, *Computer Graphics*, vol.7, n.12, 249-268, 1987.
- [3] H.Chiyokura, *Solid Modelling with DesignBase: Theory and Implementation*, Addison-Wesley, 1988.
- [4] H Chiyokura e F Kimura, *A New Surface Interpolation Method for Irregular Curve Models*, *Computer Graphics Forum*, vol.3, n.3, 209-218, 1984.
- [5] H.Chiyokura e F.Kimura, *Design of Solids with Free Form Surfaces*, *Computer Graphics*, vol.17, n.3, pp. 289-298, 1983.
- [6] S.A.Coons, *Surfaces for Computer Aided Design of Space Forms*, MIT Project Report MAC TR-41, June 1967.
- [7] G.Farin, *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press, 1991.
- [8] M.Hosaka, *Modeling of Curves and Surfaces in CAD/CAM*, Springer-Verlag, 1992.
- [9] K.Kado, A.Shimamura e K.Iinoda, *A Surface Interpolating Method for a Car-Styling Designer's CAD Work Tool*, *Proceeding of CG International*, 1992, pp.517-534.
- [10] S.Kawabe e M.Taguchi, *A Solid Modeling System with Modularized Structure*, *Proceeding of Computer Applications in Production and Engineering*, North Holland, 199-214, 1987.
- [11] M.MÄNTYLÄ, *An Introduction To Solid Modeling*, Computer Science Press, 1988..
- [12] D.F.Rogers e J.A.Adams, *Computer Graphics*, McGraw-hill, 1976.
- [13] T.Takamura, *Subdivision Method of the Gregory Patch and its Application*, *Ricoh Technical Report*, 18:4-9, October 1988.
- [14] H.Toriya. e H.Chiyokura, *3D CAD Principles and Applications*, Springer-Verlag, 1991.

- [15] M.S.G.Tsuzuki, **Modelagem de Sólidos: Modelos Limitantes (B-Rep)**, Apostila do curso PMR 2520, Departamento de Engenharia Mecatrônica e de Sistemas Mecânicas da EPUSP, 2004.
- [16] J.U.Turner, **Accurate Solid Modeling Using Polyhedral Approximations**, IEEE Computer Graphics and Application, vol.8, n.3, 1988, pp 14-28.
- [17] R.Y.Ueda, **Representando curvas e superfícies em modeladores poliedrais B-Rep por meio de atributos associados aos elementos aresta e face**, Dissertação de mestrado, Departamento de Engenharia Mecânica da EPUSP, 1996.
- [18] M.S.G. Tsuzuki e Wang Congli, **Representação de Curvas e Superfícies em Modeladores de Sólido B-Rep**, Proceeding of COBEM, 2003.
- [19] Woo Mason, Neider Jackie, Davis Tom e Shreiner Dave, **OpenGL Programming guide, the official Guide to learning OpenGL Version 1.2**, Addison Wesley, 1999.