

**Mário Carneiro Faustini**

**Proposta de Algoritmo para Determinação da  
Curva de Intersecção entre Superfícies NURBS**

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

**Área de Concentração:  
Engenharia Mecânica**

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

**São Paulo**

**1999**

**Mário Carneiro Faustini**

**Proposta de Algoritmo para Determinação da  
Curva de Intersecção entre Superfícies NURBS**

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

**São Paulo**

**1999**

## **Agradecimentos**

Aos meus pais pelos inestimáveis apoio e incentivo durante meu mestrado.

Ao Prof. Marcos Tsuzuki por seu auxílio fundamental e sempre estimulante ao longo deste trabalho de pesquisa.

À FAPESP pelo suporte financeiro dado ao autor e a este estudo, e à CAPES pelo apoio dado ao orientador.

## Resumo

O cálculo da curva de intersecção entre duas superfícies é um algoritmo básico para várias aplicações, tais como: máquinas de estereolitografia, operação de suavização entre superfícies, operações booleanas entre sólidos e superfícies, geração da trajetória da ferramenta para usinagem de superfícies, entre outras. Existem diversos métodos para solucionar tal problema, com abordagens restritas a alguns tipos de superfícies e possuindo diversas limitações quanto ao tratamento de casos especiais. Dentre estes métodos, o algoritmo mais amplamente utilizado é o algoritmo da Caminhada, e os seus problemas são: necessita de pontos iniciais como entrada para o algoritmo, e processamento de pontos singulares - *pontos de intersecção onde as superfícies possuem normais colineares ou nulas*.

Assim, o objetivo deste trabalho é definir e implementar um algoritmo que encontre eficientemente os pontos iniciais para um algoritmo de Caminhada que não seja limitado no tratamento de pontos singulares, de modo a torná-lo um algoritmo robusto.

## Abstract

The calculation of the intersection curve between two surfaces is a basic algorithm for a large number of applications, such as: stereolithography machines, smoothing surfaces, boolean operations between solids and surfaces, determination of cutting path in milling surfaces, and others. Several methods have been developed in order to solve such problem, most of them restricted to some types of mathematical surfaces and with strong limitations in handling special cases. Amongst these methods, the most widely used is the Marching algorithm, and its problems are: determination of initial points for the intersection curves and dealing with singular points – *intersection points where both surfaces share colinear or null normal vectors*.

Therefore, the objective of this work is to implement an algorithm which may efficiently find initial points for an improved Marching step with fewer limitations in handling singular points, resulting in a more robust algorithm.

## Sumário

1) Introdução.....	11
2) Revisão Bibliográfica sobre Curvas e Superfícies .....	14
2.1) Curvas .....	14
2.2) Representação Paramétrica de Curvas Sintéticas.....	15
2.3) Curvas de Bézier .....	17
2.4) Uma representação paramétrica de superfícies sintéticas: as Superfícies de Bézier .....	24
2.5) Exemplo de um Método de Determinação da intersecção entre superfícies estudado: O Método das Subdivisões Recursivas (ou Bézier-Bernstein).....	26
2.6) Exemplo de um Método de Determinação da intersecção entre superfícies estudado: o Algoritmo Geométrico de Caminhada com busca pelo primeiro ponto ....	27
2.7) Justificativa na escolha dos métodos implementados neste trabalho.....	31
3) Estudo do Problema .....	33
3.1) Método da Subdivisão dos Domínios Paramétricos.....	34
3.2) Método da Análise dos Domínios Paramétricos .....	40
3.3) Transformando um Polinômio para a Base de Bernstein.....	46
3.3.1) Convertendo Polinômios com uma Variável para a Base de Bernstein [Hoschek93].....	46
3.3.2) Convertendo Polinômios de duas Variáveis para a Base de Bernstein.....	48
3.3.3) Convertendo Polinômios com $n$ Variáveis para a Base de Bernstein .....	51
3.4) Solucionando Sistemas Polinomiais Não-Lineares.....	53
3.4.1) Método do Poliedro Projetado .....	53
3.4.2) Exemplo Prático .....	58
3.5) Determinando a Curva de Intersecção a partir de Pontos Singulares da mesma através do Método da Caminhada com equações diferenciais.....	64
3.5.1) Casos especiais: Situações de Indeterminação e a função Distância Orientada .....	65
3.6) Implementação e Exemplos Com Superfícies de Bézier .....	69
4) B-Splines e NURBS.....	76

4.1) Curvas e Superfícies B-Spline .....	76
4.2) Superfícies NURBS .....	81
4.3) Exemplos Resolvidos com o Sistema Implementado .....	83
5) Conclusões .....	88
6) Referências Bibliográficas .....	89
<u>Apêndice 1</u> : Demonstração de algumas propriedades dos Polinômios de Bernstein.....	91
A1.1 .....	91
A1.2.....	91
<u>Apêndice 2</u> : Demonstração de algumas das equações descritas no Método da Caminhada .....	93
A2.1 .....	93
A2.2.....	93
A2.3.....	94
A2.4.....	95
<u>Apêndice 3</u> : O Método Runge-Kutta de 4ª ordem .....	97

## Lista de Figuras

Figura 2.1: Reta com representação analítica. ....	16
Figura 2.2: Curva obtida pela interpolação de pontos.....	16
Figura 2.3: Curva obtida pela aproximação de pontos.....	17
Figura 2.4: Curva de Bézier cúbica (nomenclatura) [Zeid91]. ....	18
Figura 2.5: Curvas de Bézier cúbicas para vários conjuntos de pontos de controle [Zeid91].....	18
Figura 2.6: Modificações sobre uma curva de Bézier cúbica [Zeid91]. ....	22
Figura 2.7: Ajuste de segmentos de curva de Bézier [Zeid91]. ....	23
Figura 2.8: Envoltória convexa de uma curva de Bézier. ....	24
Figura 2.9: Exemplo de uma superfície de Bézier de grau 3 [Zeid91]. ....	26
Figura 2.10: Exemplo do domínio paramétrico de uma superfície.....	26
Figura 2.11: Exemplo de curvas de intersecção entre uma superfície de Bézier e um plano determinadas através do Método das Subdivisões Recursivas. ....	27
Figura 2.12: Determinação de um ponto <b>P</b> sobre a curva de intersecção. ....	28
Figura 3.1: Exemplo de um conjunto solução da intersecção entre duas superfícies, formado por curvas disjuntas, vistas a partir do domínio paramétrico de uma das superfícies. ....	33
Figura 3.2: Exemplo de uma curva aberta de intersecção entre Sup. 1 e Sup. 2, vista a partir de cada domínio paramétrico, com os pontos das extremidades destacados. Notar que $A_1$ e $A_2$ representam o <b>mesmo</b> ponto no espaço 3D, bem como $B_1$ e $B_2$ . ....	34
Figura 3.3: domínio paramétrico subdividido através dos pontos com normais colineares assinalados. ....	36
Figura 3.4: Exemplo de intersecção num ponto de inflexão.....	36
Figura 3.5: Exemplo de superfícies tangentes ao longo de um laço de intersecção. ....	38
Figura 3.6: A superfície superior é tangente ao plano A ao longo de um laço de intersecção, o mesmo ocorrendo com a superfície inferior e o plano B, sendo A e B paralelos. ....	38
Figura 3.7: Exemplos de casos passíveis de ocorrer na aplicação do método. ....	39



Figura 3.8: Pontos iniciais das curvas de intersecção sobre as bordas de um domínio paramétrico.....	40
Figura 3.8: Possíveis topologias para as curvas de intersecção a partir dos pontos iniciais da Figura 3.6. ....	41
Figura 3.9: Linhas paralelas (pontilhadas) para varredura do espaço paramétrico.....	41
Figura 3.10: Exemplo de determinação dos pontos de virada das curvas de intersecção através das linhas de varredura. ....	42
Figura 3.11: Vetor de pesos $\mathbf{W}$ , indicando que o valor do peso $w_g$ depende dos valores dos pesos $w_0$ até $w_{g-1}$ , como exposto pela expressão ( 53 )......	48
Figura 3.12: Matriz de pesos $\mathbf{W}$ .....	50
Figura 3.13: Projeção dos pontos de controle de $F_1$ para a variável $x$ . ....	60
Figura 3.14: Projeção dos pontos de controle de $F_2$ para a variável $x$ . ....	60
Figura 3.15: Projeção dos pontos de controle de $F_1$ para a variável $y$ . ....	61
Figura 3.16: Projeção dos pontos de controle de $F_2$ para a variável $y$ . ....	61
Figura 3.17: Projeção dos pontos de controle de $F_2$ para a variável $y'$ .....	62
Figura 3.18: Exemplos de cusp point e saddle point (ponto de sela).....	67
Figura 3.19: Elemento da lista ligada que armazena os polinômios no sistema. ....	69
Figura 3.20: Superfícies de Bézier cuja intersecção se deseja determinar.....	71
Figura 3.21: Pontos iniciais relativos à Figura 3.20 obtidos através do Método da Subdivisão dos Domínios Paramétricos. Tempo para determinação destes pontos: 59.130s. ....	71
Figura 3.22: Pontos iniciais relativos à Figura 2.20 obtidos através do Método da Análise dos Domínios Paramétricos. Tempo para determinação destes pontos: 4.610s.....	72
Figura 3.23: Curvas de intersecção entre as superfícies da Figura 3.20, traçadas através do Método da Caminhada com equações diferenciais. ....	72
Figura 3.24: Superfícies de Bézier cuja intersecção se deseja determinar.....	73
Figura 3.25: Pontos iniciais relativos à Figura 3.24 obtidos através do Método da Subdivisão dos Domínios Paramétricos. Tempo para determinação destes pontos: 1508.530s. ....	73
Figura 3.26: Pontos iniciais relativos à Figura 3.24 obtidos através do Método da Análise dos Domínios Paramétricos. Tempo para determinação destes pontos: 51.360s.....	73

Figura 3.27: Curvas de intersecção entre as superfícies da Figura 3.24, traçadas através do Método da Caminhada com equações diferenciais. ....	74
Figura 3.28: Superfícies de Bézier cuja intersecção se deseja determinar. ....	74
Figura 3.29: Pontos iniciais relativos à Figura 3.28 obtidos através do Método da Análise dos Domínios Paramétricos. Tempo para determinação destes pontos: 4.230s. ....	75
Figura 3.30: Curvas de intersecção entre as superfícies da Figura 3.28, traçadas através do Método da Caminhada com equações diferenciais. ....	75
Figura 4.1: Regiões do domínio paramétrico de uma curva B-Spline. ....	78
Figura 4.2: Regiões do domínio paramétrico de uma superfície B-Spline. ....	80
Figura 4.3: Superfície NURBS, com os respectivos pontos de controle destacados. ....	84
Figura 4.4: Superfície NURBS, com os respectivos pontos de controle destacados. ....	84
Figura 4.5: Pontos iniciais para o primeiro retalho da superfície <b>F</b> com a superfície <b>G</b> , vistos dos respectivos domínios paramétricos. Tempo de processamento: 49,5s. ....	85
Figura 4.6: Pontos iniciais para o segundo retalho da superfície <b>F</b> com a superfície <b>G</b> , vistos dos respectivos domínios paramétricos. Tempo de Processamento: 58,6s. ...	85
Figura 4.7: Curvas de intersecção traçadas através do Método da Caminhada e vistas a partir dos respectivos domínios paramétricos. ....	85
Figura 4.8: Imagem renderizada das superfícies <b>F</b> e <b>G</b> . ....	86
Figura 4.9: Superfície NURBS, com os respectivos pontos de controle destacados. ....	86
Figura 4.10: Pontos iniciais para o primeiro retalho da superfície <b>F</b> com a superfície <b>H</b> , vistos dos respectivos domínios paramétricos. Tempo de processamento: 1658s. ....	87
Figura 4.11: Pontos iniciais para o segundo retalho da superfície <b>F</b> com a superfície <b>H</b> , vistos dos respectivos domínios paramétricos. Tempo de Processamento: 2339s. ...	87
Figura 4.12: Curvas de intersecção traçadas através do Método da Caminhada e vistas a partir dos respectivos domínios paramétricos. ....	87
Figura 4.13: Imagem renderizada das superfícies <b>F</b> e <b>H</b> . ....	88

## 1) Introdução

Para o desenvolvimento de Modeladores de Sólidos que suportem a representação geométrica de superfícies, é necessário que o problema de determinar a curva de intersecção entre superfícies seja resolvido. A aplicação deste algoritmo estaria na geração do sólido resultado de Operações Booleanas [Ueda96], na definição do caminho de corte de usinagem para fresas [Makita96], na definição de programas para máquinas de estereolitografia, e em muitos algoritmos para suavização de arestas os quais assumem que você conhece a curva de intersecção entre as duas superfícies que se deseja suavizar, dentre outras aplicações.

Este problema já foi inicialmente tratado em [Pincinato96], e a nosso entender é possível definir para cada algoritmo um domínio de aplicação. Entretanto, algumas situações muito comuns ainda não puderam ser incluídas no domínio de aplicação dos algoritmos estudados. Por exemplo, não é possível determinar a curva de intersecção se a curva possuir pontos singulares\*.

A grande diversidade de possíveis representações de superfícies já foi uma barreira para a definição de um algoritmo genérico. Se um Modelador de Sólidos suporta duas representações de superfícies (quádricas e Bézier), devemos possuir três algoritmos para determinar a curva de intersecção: um para determinar a intersecção entre duas quádricas, outro para determinar a intersecção entre duas superfícies de Bézier e um terceiro para determinar a intersecção entre Bézier e quádricas. Atualmente, com a possibilidade de converter as várias formas de representação para superfícies NURBS (Non-Uniform Rational B-Splines) [Piegel97, Fiorot92], é possível adotar um algoritmo único: *algoritmo para cálculo da curva de intersecção entre superfícies NURBS*.

---

\* Define-se por ponto singular os pontos no espaço onde os gradientes das duas superfícies cuja intersecção se deseja determinar são colineares ou nulos.

Para a implementação inicial dos métodos aqui descritos, foram utilizadas superfícies de Bézier pela sua maior facilidade de manipulação para efeito de testes comparativos entre os mesmos. Posteriormente, o algoritmo mais eficiente foi convenientemente estendido para que a proposta apresentada lidasse com seu objetivo final: superfícies NURBS.

Vários autores estudaram a determinação de pontos iniciais para realizar o traçado de curvas de intersecção entre superfícies paramétricas [Kriezis92;Hu97;Grandine97]. Entretanto, conforme será verificado ao longo deste trabalho, comparando-se a proposta de [Hu97] e [Grandine97], o primeiro método possui um maior processamento, determinando um maior número de pontos. A segunda proposta possui um menor processamento, determinando um conjunto menos redundante de pontos iniciais. A determinação dos pontos iniciais em ambas as propostas envolve a solução de sistemas polinomiais. Para solucioná-los utilizaremos a proposta do poliedro projetado [Sherbrooke93]. Entretanto, para utilizarmos esta proposta é necessário um conversor de polinômios em forma de potência para um equivalente em base de Bernstein, que é a nova contribuição da presente tese. Desta maneira os pontos iniciais são determinados, restando realizar o traçado da curva de intersecção. Para isto utilizaremos a proposta feita por [Hu97] que determina a curva baseando-se na solução de equações diferenciais ordinárias dadas pelas derivadas paramétricas das duas superfícies no ponto em análise. Existe um caso singular quando os planos tangentes são coplanares; neste caso utiliza-se a função distância orientada proposta em [Wang91;Kriezis92] que permite processar mesmo nas situações singulares.

Este trabalho apresenta-se, então, estruturado da seguinte forma: no capítulo 2, faz-se uma revisão bibliográfica sobre os conceitos de curvas e superfícies a serem utilizadas ao longo da proposta, bem como descrevem-se brevemente dois exemplos de métodos de determinação da intersecção entre superfícies previamente estudados; no capítulo 3 aborda-se diretamente o problema em questão, mostrando as propostas para solução do mesmo, alguns métodos auxiliares para realizar tal solução e exemplos relativos à implementação utilizando-se superfícies de Bézier, fazendo-se, através destas,

um estudo comparativo entre as propostas descritas sobre a eficiência das mesmas; no capítulo 4 é realizada uma introdução do conceito de NURBS, discutindo-se em seguida a aplicação da metodologia mais eficiente apresentada no capítulo 3 para tratar este tipo de superfície, com exemplos de curvas de intersecção entre superfícies NURBS obtidos com a implementação final da proposta deste trabalho; o capítulo 5 traz as conclusões, o 6 as referências bibliográficas e, por fim, o Apêndice exibe demonstrações matemáticas de algumas expressões utilizadas ao longo do trabalho.

## 2) Revisão Bibliográfica sobre Curvas e Superfícies

Curvas espaciais ou tridimensionais e superfícies possuem um papel de destaque no projeto e manufatura de diversos produtos, como automóveis, cascos de navios, fuselagem e asas de aviões, etc.

### 2.1) Curvas

Uma curva [Ueda96] pode ser representada por um conjunto de pontos (representação sintética). Desde que os pontos estejam apropriadamente espaçados, eles podem ser conectados por segmentos de linha, fornecendo uma representação visual adequada para a curva. Em outras aplicações, uma representação analítica (representação segundo a equação associada à curva) será mais adequada. A representação analítica possui muitas vantagens sobre a representação sintética, tais como precisão, armazenamento mais compacto e facilidade de cálculo de pontos intermediários.

Matematicamente, tanto a forma paramétrica quanto a não-paramétrica são utilizadas para a representação analítica de uma curva. A representação não-paramétrica pode ser tanto implícita quanto explícita. Uma representação não-paramétrica explícita é dada por:

$$z = f(x, y) \quad (1)$$

Desta forma, para cada par de valores  $(x, y)$  apenas um valor de  $z$  é obtido. Conseqüentemente, curvas fechadas, como circunferências, não podem ser representadas explicitamente. Uma representação não-paramétrica implícita é dada por:

$$f(x, y, z) = 0 \quad (2)$$

Tanto a representação da curva não-paramétrica explícita quanto a implícita são dependentes do sistema de coordenadas. Além do mais, quando pontos da curva são

calculados, segundo incrementos iguais de  $x$ ,  $y$  ou  $z$ , os pontos freqüentemente estarão igualmente distribuídos, segundo o comprimento da curva. Esta desigualdade na distribuição dos pontos afetará a precisão da representação gráfica. Neste trabalho vamos limitar-nos ao estudo das curvas sintéticas e mais especificamente à curva de Bézier.

## 2.2) Representação Paramétrica de Curvas Sintéticas

Na forma paramétrica cada ponto sobre a curva é representado por um único parâmetro. Para uma curva tridimensional, com  $t$  como parâmetro, as coordenadas cartesianas de um ponto sobre a curva são dadas por:

$$\begin{aligned}x &= x(t) \\y &= y(t) \\z &= z(t)\end{aligned}\tag{3}$$

O vetor posição de um ponto sobre a curva é dado por:

$$\mathbf{P}(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix}\tag{4}$$

A representação mais simples para uma curva paramétrica ocorre para um segmento de reta. Dados dois pontos extremos do segmento de  $\mathbf{P}_1$  e  $\mathbf{P}_2$ , que definem um segmento de reta, a representação paramétrica para este segmento é dada por:

$$\mathbf{P}(t) = \mathbf{P}_1 + t \cdot (\mathbf{P}_2 - \mathbf{P}_1) \quad 0 \leq t \leq 1\tag{5}$$

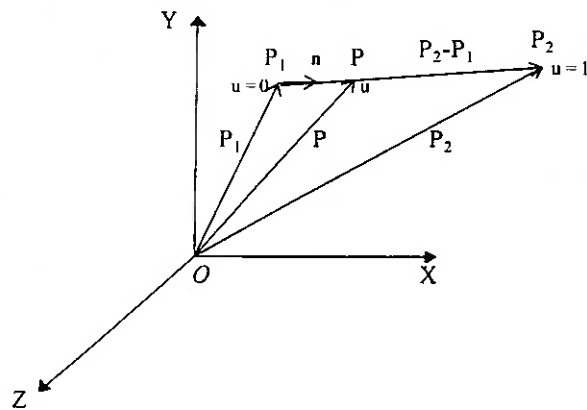


Figura 2.1: Reta com representação analítica.

Como  $P(t)$  é um vetor posição, cada um dos componentes de  $P(t)$  possui uma representação paramétrica (conforme visualizado na Figura 2.1):

$$\left. \begin{aligned} x(t) &= x_1 + t.(x_2 - x_1) \\ y(t) &= y_1 + t.(y_2 - y_1) \\ z(t) &= z_1 + t.(z_2 - z_1) \end{aligned} \right\} \quad 0 \leq t \leq 1 \quad (6)$$

Na literatura, existe uma grande variedade de formas de definir uma curva paramétrica. Cada uma delas pode ser classificada como sendo interpolação ou aproximação de pontos [Bartels87]. No caso de uma curva interpolada, esta deverá passar por um conjunto de pontos fornecidos  $P_i$ , em uma seqüência ordenada (vide Figura 2.2).

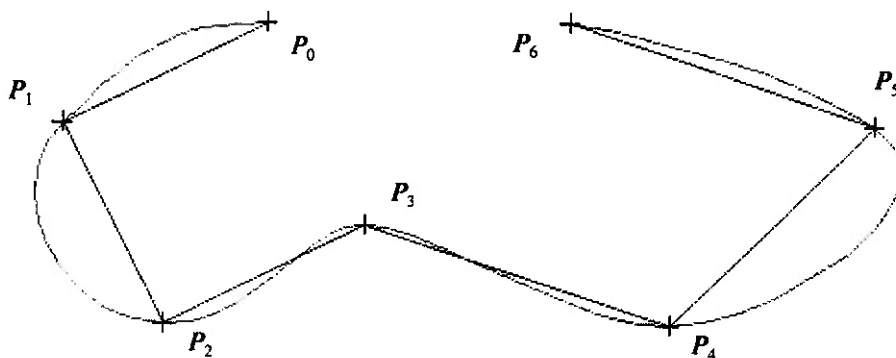
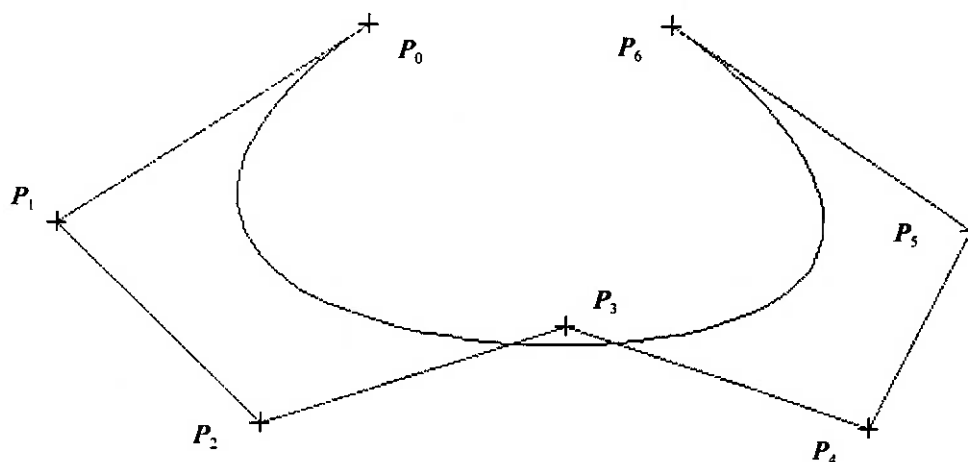


Figura 2.2: Curva obtida pela interpolação de pontos.



Para as técnicas baseadas na aproximação a curva deverá passar “próxima” aos conjunto de pontos fornecidos, que serão chamados pontos de controle (vide Figura 2.3). O significado exato de “próximo” dependerá de cada técnica em particular.



*Figura 2.3: Curva obtida pela aproximação de pontos.*

Em ambos os casos, ao movimentarmos um ponto do conjunto fornecido, a curva é alterada. Neste trabalho vamos estudar com maior detalhe as curvas de Bézier e curvas NURBS que são curvas aproximadoras. Um dos fatores que determina o uso das curvas em uma aplicação específica é a influência do conjunto de pontos sobre a curva. O projetista pode preferir controlar a forma da curva de modo local ou global, alterando a posição de um dos pontos. Se a mudança deste ponto resulta em uma mudança localizada da curva, nas vizinhanças do ponto, temos um controle local da curva; em caso contrário temos um controle global. A seguir detalharemos as curvas de Bézier.

### **2.3) Curvas de Bézier**

As curvas e superfícies de Bézier são creditadas a P. Bézier da firma automobilística francesa Regie Renault que desenvolveu e utilizou estas curvas no sistema de software chamado UNISURF, o qual foi utilizado pelos projetistas para definir os painéis que compõem a carroceria de vários automóveis Renault. Estas curvas, chamadas curvas de Bézier, foram também desenvolvidas independentemente por P. DeCasteljau da companhia de automóveis Citroën. O sistema UNISURF de Bézier foi

publicado primeiro na literatura; esta é a razão das curvas levarem agora o nome de Bézier [Bartels87, Zeid91].

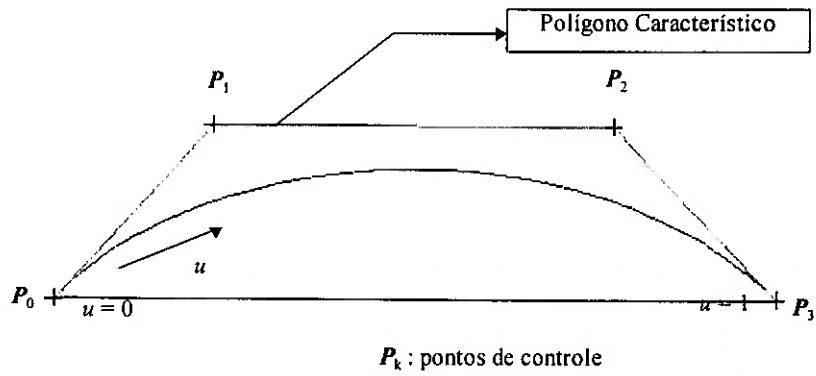


Figura 2.4: Curva de Bézier cúbica (nomenclatura) [Zeid91].

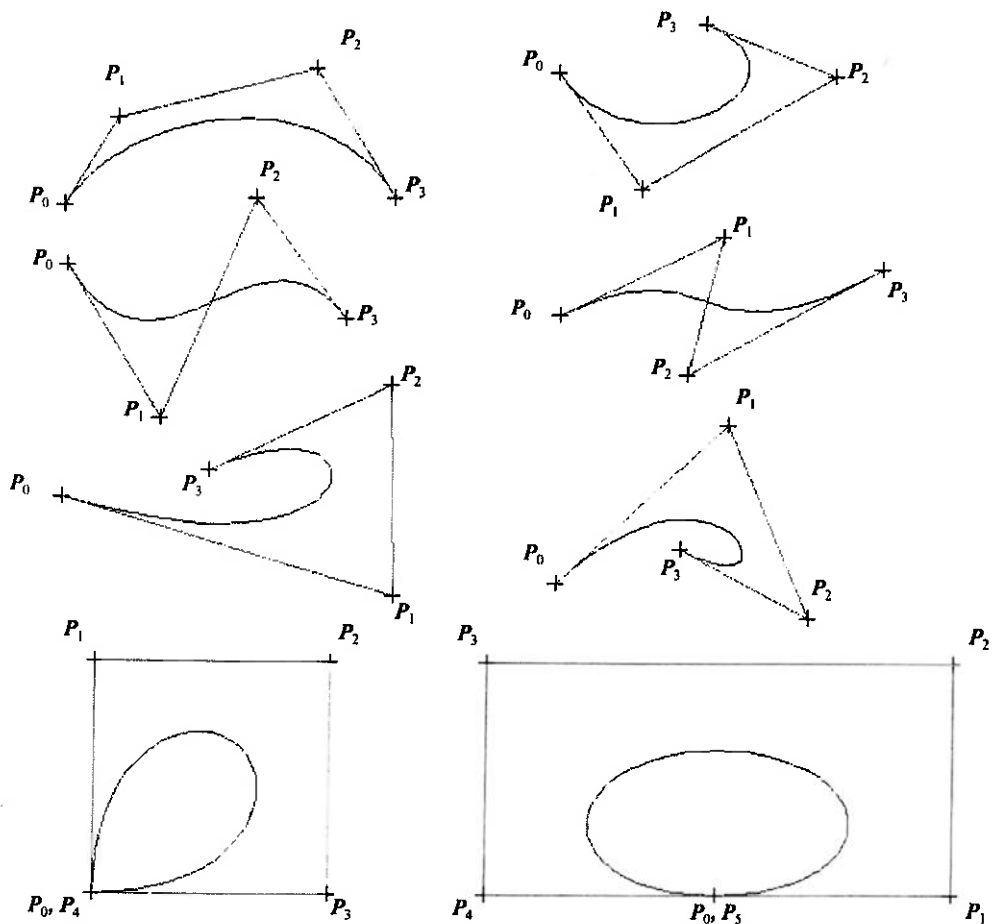


Figura 2.5: Curvas de Bézier cúbicas para vários conjuntos de pontos de controle [Zeid91].

A curva de Bézier é definida em termos da localização dos  $n+1$  pontos de controle. Eles formam os vértices do que chamamos de polígono de controle ou polígono característico de Bézier como é mostrado na Figura 2.4.

A Figura 2.5 ilustra várias curvas de Bézier cúbicas para diversos conjuntos de pontos de controle. Matematicamente, para  $n+1$  pontos de controle, a curva de Bézier é definida pelo seguinte polinômio de grau  $n$ :

$$P(u) = \sum_{i=0}^n P_i \cdot B_i^n(u), \quad 0 \leq u \leq 1 \quad (7)$$

onde  $P(u)$  é um ponto da curva e  $P_i$  é o  $i$ -ésimo ponto de controle.  $B_i^n(u)$  são os polinômios de Bernstein, que atuam como funções de base ou de ajuste para a curva de Bézier e são dados por:

$$B_i^n(u) = \binom{n}{i} \cdot u^i \cdot (1-u)^{n-i} \quad (8)$$

onde  $\binom{n}{i}$  é o coeficiente binomial:

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} \quad (9)$$

Os polinômios de Bernstein, por sua vez, têm, dentre suas principais propriedades, as seguintes, a serem usadas ao longo deste trabalho:

$$B_i^m(u) \geq 0 \quad \forall u \in [0,1] \quad (10)$$

e

$$\sum_{i=0}^m B_i^m(u) = 1 \quad (11)$$

Utilizando as equações ( 7 ) e ( 8 ) e observando que  $\binom{n}{0} = \binom{n}{n} = 1$ , a equação ( 7 ) pode ser expandida para:

$$\begin{aligned}
\mathbf{P}(u) = & \mathbf{P}_0 \cdot (1-u)^n + \mathbf{P}_1 \cdot \binom{n}{1} \cdot u \cdot (1-u)^{n-1} + \mathbf{P}_2 \cdot \binom{n}{2} \cdot u^2 \cdot (1-u)^{n-2} \\
& + \dots + \mathbf{P}_{n-1} \cdot \binom{n}{n-1} \cdot u^{n-1} \cdot (1-u) + \mathbf{P}_n \cdot u^n, \quad 0 \leq u \leq 1
\end{aligned} \tag{12}$$

As características da curva de Bézier são baseadas nas propriedades dos polinômios de Bernstein e podem ser resumidas da seguinte forma:

1. A curva interpola o primeiro e o último ponto de controle; isto é, ela passa por  $\mathbf{P}_0$  e  $\mathbf{P}_n$ , o que pode ser sentido substituindo  $u = 0$  e  $u = 1$  respectivamente na equação (12).
2. A curva é tangente ao primeiro e ao último segmentos do polígono característico. Utilizando a (7) e a (8), a  $r$ -ésima derivada nos pontos inicial e final da curva pode ser dada respectivamente por [Farin88]:

$$\mathbf{P}'(0) = \frac{n!}{(n-r)!} \sum_{i=0}^r (-1)^{r-i} \cdot \binom{r}{i} \cdot \mathbf{P}_i \tag{13}$$

$$\mathbf{P}'(1) = \frac{n!}{(n-r)!} \sum_{i=0}^r (-1)^{r-i} \cdot \binom{r}{i} \cdot \mathbf{P}_{n-i} \tag{14}$$

Logo as derivadas primeiras nos pontos extremos são:

$$\mathbf{P}'(0) = n \cdot (\mathbf{P}_1 - \mathbf{P}_0) \tag{15}$$

$$\mathbf{P}'(1) = n \cdot (\mathbf{P}_n - \mathbf{P}_{n-1}) \tag{16}$$

onde  $(\mathbf{P}_1 - \mathbf{P}_0)$  e  $(\mathbf{P}_n - \mathbf{P}_{n-1})$  definem o primeiro e o último segmentos do polígono característico da curva.

3. A curva é simétrica com respeito a  $u$  e  $(1-u)$ . Isto significa que a seqüência de pontos de controle que definem a curva pode ser revertida sem alterar a forma da curva; isto é, revertendo a direção de parametrização não ocorre mudança na forma da curva. Isto pode ser conseguido substituindo-se  $1-u = v$  em (12) e notando que  $\binom{n}{i} = \binom{n}{n-i}$ . Isto é resultante do fato de que  $B_i^n(u)$  e  $B_{n-i}^n(u)$  são simétricos se plotados como função de  $u$ .

4. O polinômio de interpolação  $B_i^n(u)$  tem valor máximo de  $\binom{n}{i} \cdot (i/n)^i \cdot (1-i/n)^{n-i}$ , que ocorre em  $u = i/n$ . (Obtido da equação  $d(B_i^n(u))/du = 0$ ). Isto implica que cada ponto de controle influencia de forma mais acentuada um determinado trecho da curva. Por exemplo para uma curva de Bézier cúbica, os pontos  $P_0$ ,  $P_1$ ,  $P_2$  e  $P_3$  têm maior influência nos arredores de  $u = 0$ ,  $1/3$ ,  $2/3$  e  $1$  respectivamente [Farin88].
5. A forma da curva pode ser modificada pelo reposicionamento de um ou mais vértices do polígono, ou mantendo o polígono fixo e definindo múltiplos pontos coincidentes em um mesmo vértice. A Figura 2.6(a) mostra um ponto  $P_2$  sendo movimentado para uma nova posição  $P_2^*$  e Figura 2.6(b) ilustra um vértice  $P_2$  com várias multiplicidades. Quanto maior a multiplicidade de  $P_2$  mais a curva é “puxada” para este vértice [Farin88].
6. Para qualquer valor válido de  $u$ , a soma das funções  $B_i^n(u)$  associadas aos pontos de controle é igual à unidade, para qualquer grau de curva de Bézier [Farin88].

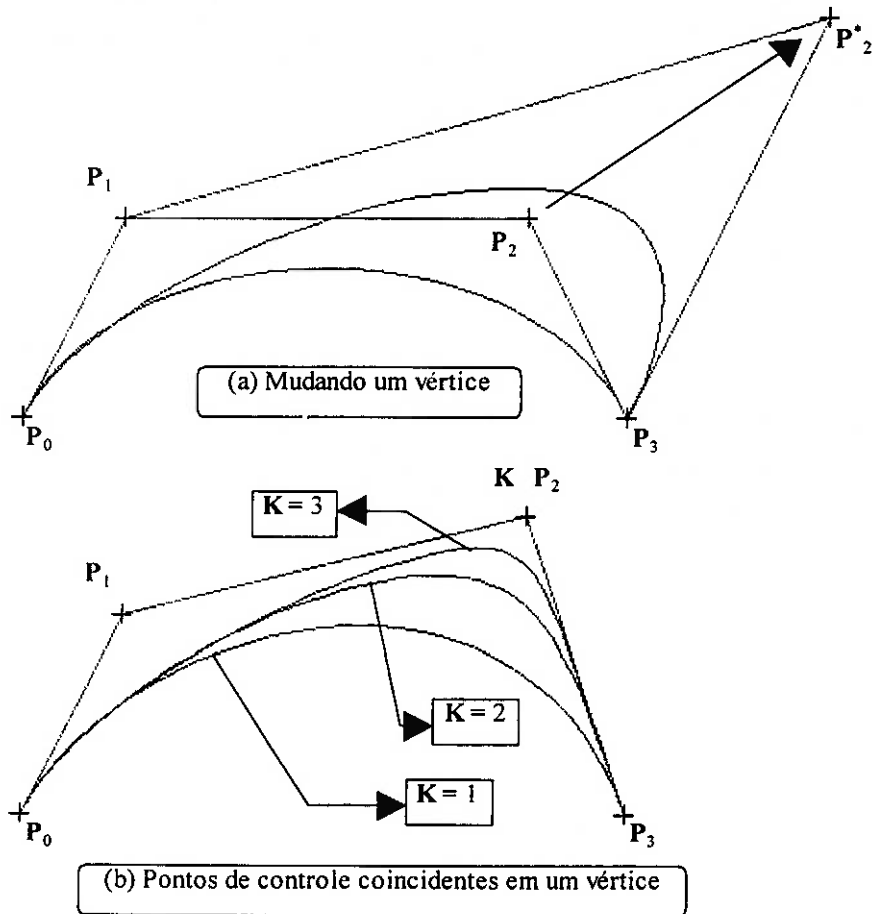


Figura 2.6: Modificações sobre uma curva de Bézier cúbica [Zeid91].

Até agora somente um segmento de curva de Bézier foi considerado. Entretanto, em aplicações práticas, muitas vezes é necessário compor vários segmentos de curva. Nestas aplicações é desejável manter a continuidade entre as curvas. A Figura 2.7 mostra dois conjuntos de pontos  $P_1, P_2, P_3$  e  $P_4$ ; e  $P_5, P_6, P_7$  e  $P_8$ . Para obter a continuidade de ordem zero é suficiente colocar os pontos extremos coincidentes, na Figura 2.7 (a) é o ponto  $P_4$ . Para obter a continuidade de primeira ordem é preciso que as tangentes do final do primeiro segmento e do início do segundo segmento tenham a mesma direção e sejam proporcionais segundo uma constante [Yamaguchi88]. No exemplo, os pontos  $P_3, P_4$  e  $P_5$  devem ser colineares, como mostra a Figura 2.7 (b). Utilizando ( 15 ) e ( 16 ) pode-se escrever:

$$P_4 - P_3 = \frac{4}{3}(P_5 - P_4) \quad (17)$$

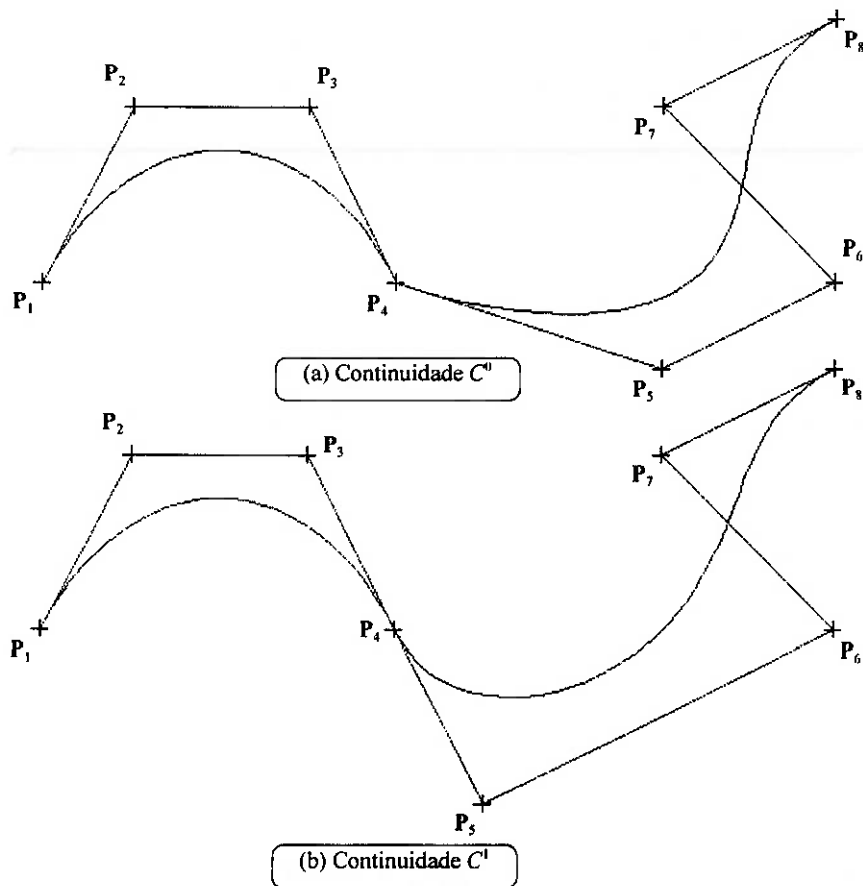


Figura 2.7: Ajuste de segmentos de curva de Bézier [Zeid91].

Uma característica muito útil é a propriedade da envoltória convexa. A envoltória convexa é definida como sendo o maior polígono convexo definido pelo conjunto de pontos de controle. É possível demonstrar que a envoltória convexa sempre contém a curva de Bézier [Farin88]. A Figura 2.8 ilustra alguns exemplos de envoltória convexa.

As curvas de Bézier, para algumas aplicações, possuem algumas desvantagens em relação a algumas outras representações. Primeiro, a curva não passa pelos pontos de controle, fato que pode ser inconveniente para alguns projetistas. Segundo, esta curva não possui controle local, mas somente controle de natureza global. Se um ponto de controle é mudado, toda a curva é modificada.

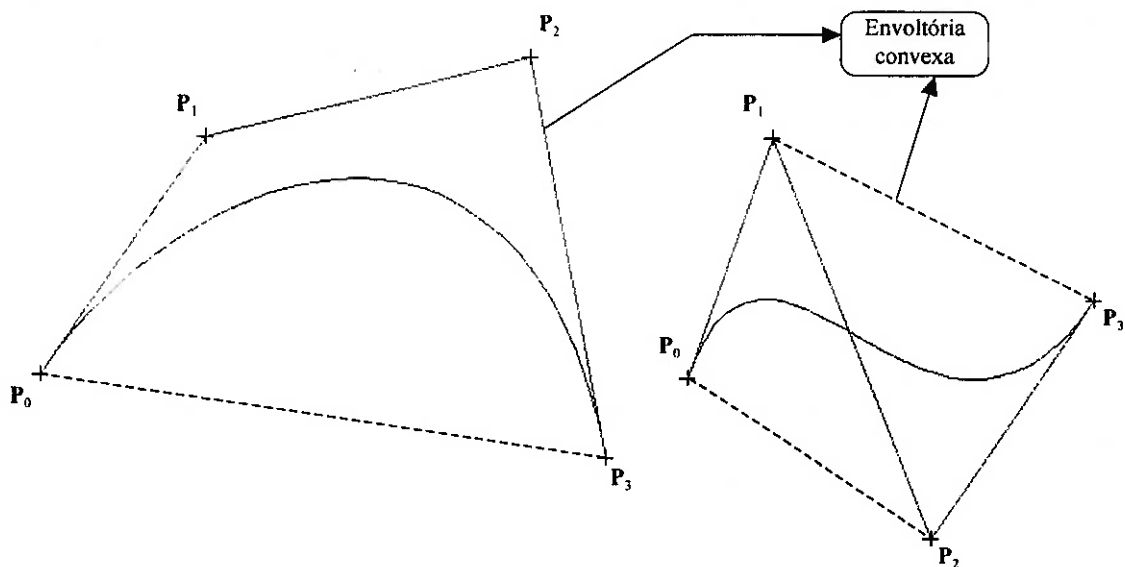


Figura 2.8: Envoltória convexa de uma curva de Bézier.

#### 2.4) Uma representação paramétrica de superfícies sintéticas: as Superfícies de Bézier

As superfícies de Bézier são superfícies paramétricas largamente utilizadas nas aplicações de CAD e na modelagem de sólidos em geral. Tais superfícies são geradas a partir de uma extrapolação bidimensional das curvas de Bézier apresentadas da seção anterior.

Assim, define-se uma superfície de Bézier da seguinte forma [Hosaka92]:

$$S(u, v) = \sum_{i=0}^m \sum_{j=0}^n B_i^m(u) \cdot B_j^n(v) \cdot P_{ij} \quad (18)$$

sendo  $P_{ij}$  são pontos do espaço que servem como pontos de controle da superfície e  $B_i^m(u)$  e  $B_j^n(v)$  são polinômios de Bernstein definidos em ( 8 ).

Os pontos de controle estão dispostos em uma rede retangular de  $n+1$  por  $m+1$ .



Assim, o vetor posição de um ponto sobre a superfície será dado por:

$$\mathbf{S}(u, v) = \begin{bmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{bmatrix} \quad (19)$$

sendo as coordenadas definidas por polinômios em  $u$  e  $v$ . O grau da superfície (e, portanto, dos polinômios que a descrevem em cada coordenada) é definido pelos valores de  $m$  e  $n$ . Doravante neste trabalho utilizar-se-ão superfícies de Bézier de 3º grau ( $m = n = 3$ ), a título de simplificação. Entretanto, como ficará claro, o método a ser implementado funciona independentemente do grau das superfícies.

Desta forma, uma superfície de Bézier de 3º grau fica definida por:

$$\mathbf{S}(u, v) = \sum_{i=0}^3 \sum_{j=0}^3 B_i^3(u) \cdot B_j^3(v) \cdot \mathbf{P}_{ij} \quad (20)$$

ou, em forma matricial [Yamaguchi88]:

$$\mathbf{S}(u, v) = \begin{bmatrix} u^3 \\ u^2 \\ u^1 \\ u^0 \end{bmatrix}^T \cdot \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \\ P_{30} & P_{31} & P_{32} & P_{33} \end{bmatrix} \cdot \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} v^3 \\ v^2 \\ v^1 \\ v^0 \end{bmatrix}$$

Uma das vantagens da forma matricial é a facilidade no cálculo das derivadas. Assim,  $S_u(u, v) = \frac{\partial \mathbf{S}(u, v)}{\partial u}$  é dada pela expressão abaixo:

$$S_u(u, v) = \begin{bmatrix} 3u^2 \\ 2u^1 \\ u^0 \\ 0 \end{bmatrix}^T \cdot \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \\ P_{30} & P_{31} & P_{32} & P_{33} \end{bmatrix} \cdot \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} v^3 \\ v^2 \\ v^1 \\ v^0 \end{bmatrix}$$

A Figura 2.9 mostra uma superfície de Bézier de grau 3 e seus respectivos pontos de controle. É interessante notar que os pontos da superfície são obtidos através dos parâmetros  $u$  e  $v$ , de modo que a mesma pode ser mapeada através do domínio paramétrico mostrado na Figura 2.10. Ou seja, para cada ponto deste domínio paramétrico  $(u, v)$  há um correspondente sobre a superfície no espaço tridimensional.

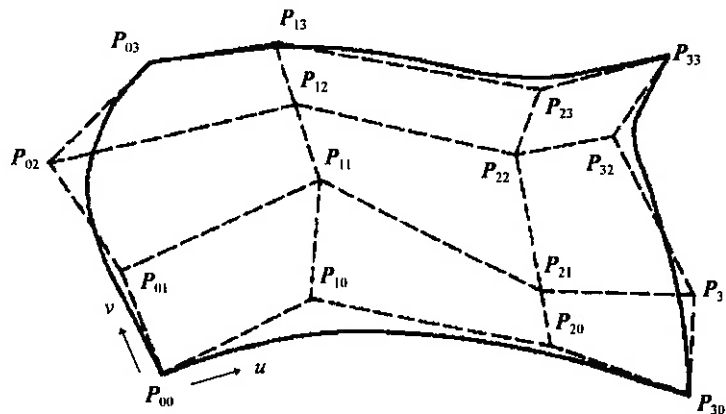


Figura 2.9: Exemplo de uma superfície de Bézier de grau 3 [Zeid91].

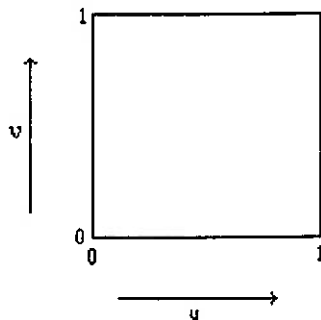
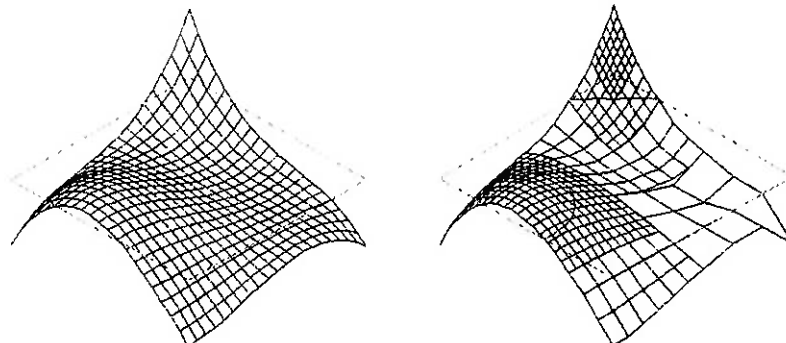


Figura 2.10: Exemplo do domínio paramétrico de uma superfície.

## 2.5) Exemplo de um Método de Determinação da intersecção entre superfícies estudado: O Método das Subdivisões Recursivas (ou Bézier-Bernstein)

Este método consiste na subdivisão recursiva do retalho (patch) de superfície por onde pode passar a curva de intersecção, decompondo a superfície em polígonos e aproximando a intersecção a partir dos segmentos em que os polígonos se interceptam. Esta subdivisão representa um refinamento da intersecção, sendo o nível deste refinamento definido pelo nível de recursão. Esta operação vai transformar, por exemplo, um retalho de superfície de Bézier com  $(n + 1)^2$  pontos de controle em quatro patches com  $(n + 1)^2$  pontos de controle cada [Lasser86]. Na Figura 2.11 mostra um exemplo deste processo.



*Figura 2.11: Exemplo de curvas de intersecção entre uma superfície de Bézier e um plano determinadas através do Método das Subdivisões Recursivas.*

Esta proposta tem como principal vantagem sua rapidez. Todavia, possui diversos problemas: ocupa larga quantidade de memória, pode perder pedaços da curva de intersecção e mesmo curvas completas dependendo no nível de recursão e tem necessidade de que sejam usados algoritmos especiais para processar os casos em que as subdivisões se tornam paralelas, impossibilitando a determinação de uma linha de intersecção [Pincinato96].

## **2.6) Exemplo de um Método de Determinação da intersecção entre superfícies estudado: o Algoritmo Geométrico de Caminhada com busca pelo primeiro ponto**

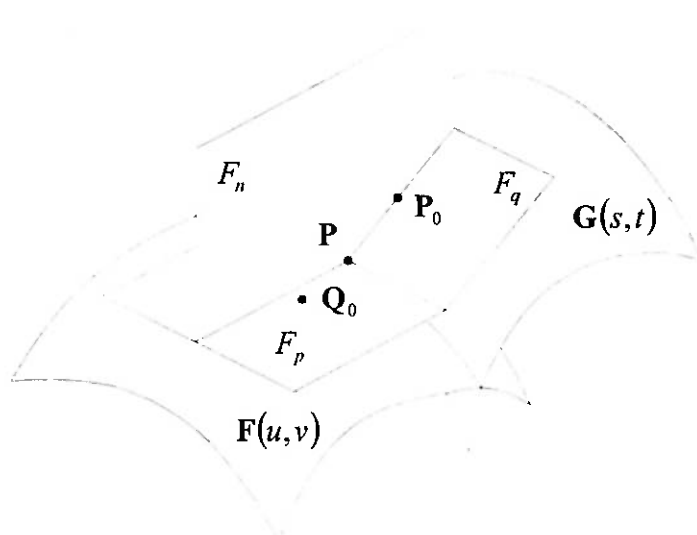
Este algoritmo utilizado para calcular a curva de intersecção entre uma ou mais superfícies paramétricas é uma proposta que consiste em três passos principais, descritos a seguir [Pincinato96;Hosaka92].

### **Primeira parte: Determinação de um ponto sobre cada uma das superfícies de Bézier.**

Devem ser escolhidos um par de pontos, um sobre cada superfície, os quais farão parte de um processo de refinamento que irá convergir para um ponto sobre uma

curva de intersecção entre as superfícies, como é mostrado a seguir.

**Segunda parte: Aproximação de um ponto qualquer sobre a superfície para um ponto sobre a curva de intersecção.**



- $G(s,t)$ : superfície paramétrica  $G$
- $F(u,v)$ : superfície paramétrica  $F$
- $P_0$ : aproximação inicial do ponto de intersecção sobre  $G(s,t)$
- $Q_0$ : aproximação inicial do ponto de intersecção sobre  $F(u,v)$
- $F_q$ : plano tangente a  $G(s,t)$  em  $P_0$
- $F_p$ : plano tangente a  $F(u,v)$  em  $Q_0$
- $F_n$ : plano auxiliar, perpendicular a  $F_q$  e  $F_p$  e passando por  $P_0$

Figura 2.12: Determinação de um ponto  $P$  sobre a curva de intersecção.

Consideremos duas superfícies paramétricas  $G(s,t)$  e  $F(u,v)$  ilustradas na Figura 2.12. Os pontos  $P_0$  e  $Q_0$  estão respectivamente sobre as superfícies  $G(s,t)$  e  $F(u,v)$ . Assim temos:

$$P_0 = G(s_0, t_0) \text{ e } Q_0 = F(u_0, v_0) \quad (21)$$

Vamos denominar o plano tangente à superfície  $G(s,t)$  no ponto  $P_0$  por  $F_p$  e seu vetor normal por  $N_p$ . Da mesma forma vamos denominar por  $F_q$  e  $N_q$  o plano tangente à superfície  $F(u,v)$  no ponto  $Q_0$  e seu vetor normal. Logo:

$$N_p = \frac{\mathbf{G}_s \times \mathbf{G}_t}{|\mathbf{G}_s \times \mathbf{G}_t|} \text{ e } N_q = \frac{\mathbf{F}_u \times \mathbf{F}_v}{|\mathbf{F}_u \times \mathbf{F}_v|} \quad (22)$$

onde:  $\mathbf{F}_u$  é a derivada de primeira ordem da superfície  $F$  na direção  $u$

$\mathbf{F}_v$  é a derivada de primeira ordem da superfície  $F$  na direção  $v$

$G_s$  é a derivada de primeira ordem da superfície  $G$  na direção  $s$

$G_t$  é a derivada de primeira ordem da superfície  $G$  na direção  $t$

Denominando por  $d_p$  e  $d_q$  as distâncias a partir da origem dos planos  $F_p$  e  $F_q$ , temos que:

$$d_p = \mathbf{N}_p \cdot \mathbf{G}(s_0, t_0) \text{ e } d_q = \mathbf{N}_q \cdot \mathbf{F}(u_0, v_0) \quad (23)$$

Vamos definir um plano  $F_n$  que passa por  $\mathbf{P}_0$  e é ortogonal aos planos  $F_p$  e  $F_q$ . Seu vetor normal será denominado por  $\mathbf{N}_n$  e sua distância à origem será  $d_n$ . Então:

$$\mathbf{N}_n = \frac{\mathbf{N}_p \times \mathbf{N}_q}{|\mathbf{N}_p \times \mathbf{N}_q|} \text{ e } d_n = \mathbf{N}_n \cdot \mathbf{G}(s_0, t_0) \quad (24)$$

A intersecção destes três planos determina um novo ponto  $\mathbf{P}$ .  $\Delta\mathbf{P}$  é definido como  $\mathbf{P} - \mathbf{P}_0$  e  $\Delta\mathbf{Q}$  como  $\mathbf{P} - \mathbf{Q}_0$ . Pode-se calcular  $\mathbf{P}$  da seguinte forma:

$$\mathbf{P} = \frac{d_p \cdot (\mathbf{N}_q \times \mathbf{N}_n) + d_q \cdot (\mathbf{N}_n \times \mathbf{N}_p) + d_n \cdot (\mathbf{N}_p \times \mathbf{N}_q)}{\mathbf{N}_n \cdot (\mathbf{N}_p \times \mathbf{N}_q)} \quad (25)$$

Vamos assumir que as seguintes relações são verdadeiras, para valores de  $\delta_w$ ,  $\delta_t$ ,  $\delta_u$  e  $\delta_v$  suficientemente pequenos:

$$\mathbf{G}_s \delta_s + \mathbf{G}_t \delta_t = \Delta\mathbf{P} \quad (26)$$

$$\mathbf{F}_u \delta_u + \mathbf{F}_v \delta_v = \Delta\mathbf{Q} \quad (27)$$

Os seguintes vetores são definidos:

$$\mathbf{G}'_s = \mathbf{G}_s \times \mathbf{N}_p \quad \mathbf{G}'_t = \mathbf{G}_t \times \mathbf{N}_p \quad (28)$$

$$\mathbf{F}'_u = \mathbf{F}_u \times \mathbf{N}_q \quad \mathbf{F}'_v = \mathbf{F}_v \times \mathbf{N}_q \quad (29)$$

Combinando (26) e (27) respectivamente com as equações (28) e (29) pode-se deduzir que:

$$\delta_s = \frac{(\mathbf{G}'_t \cdot \Delta \mathbf{P})}{(\mathbf{G}'_t \cdot \mathbf{G}_s)}, \delta_t = \frac{(\mathbf{G}'_s \cdot \Delta \mathbf{P})}{(\mathbf{G}'_s \cdot \mathbf{G}_t)}, \delta_u = \frac{(\mathbf{F}'_v \cdot \Delta \mathbf{Q})}{(\mathbf{F}'_v \cdot \mathbf{F}_u)}, \delta_v = \frac{(\mathbf{F}'_u \cdot \Delta \mathbf{Q})}{(\mathbf{F}'_u \cdot \mathbf{F}_v)} \quad (30)$$

Assim os novos pontos serão:

$$\mathbf{P}_0 = \mathbf{G}(s_0 + \delta_s, t_0 + \delta_t) \quad (31)$$

$$\mathbf{Q}_0 = \mathbf{F}(u_0 + \delta_u, v_0 + \delta_v) \quad (32)$$

Definindo  $\Delta PQ = \max\{|\Delta \mathbf{P}|, |\Delta \mathbf{Q}|\}$  repete-se o procedimento acima até que  $\Delta PQ$  seja menor que  $\Delta$ , que é a tolerância definida para o método.

**Terceira parte: Determinação do próximo ponto da curva de intersecção a partir de um ponto sobre a curva de intersecção.**

A determinação do próximo ponto da curva de intersecção é feita de maneira parecida com a aproximação de um ponto qualquer sobre a superfície para um ponto sobre a curva de intersecção. São calculados os pontos  $\mathbf{P}_0 = \mathbf{G}(s_0, t_0)$  e  $\mathbf{Q}_0 = \mathbf{F}(u_0, v_0)$  depois de feita a aproximação garantindo que esses dois pontos estejam próximos. Em seguida, calcula-se os planos  $F_p$ , tangente à superfície  $\mathbf{G}(s, t)$  em  $\mathbf{P}_0$  e  $F_q$ , tangente à superfície  $\mathbf{F}(u, v)$  em  $\mathbf{Q}_0$ . O vetor normal ao plano  $F_n$ , que passa por  $\mathbf{P}$  e é perpendicular aos planos  $F_p$  e  $F_q$  indica a direção a ser seguida. O sentido é determinado pelo sentido seguido anteriormente e devidamente armazenado para evitar que o traçado da curva tenha seu sentido invertido caso haja inversão do sinal do produto vetorial. O traçado continua até que a curva atinja uma das bordas de uma das superfícies.

Esta proposta apresenta dois problemas fundamentais: a escolha dos pontos iniciais no primeiro passo e curvas de intersecção que possuam pontos singulares. O primeiro problema é gerado pela necessidade da escolha de uma amostra de pontos em quantidade considerável em cada superfície para que se eleve a confiança na determinação de pontos iniciais para todas as curvas de intersecção. Ainda assim, mesmo

para uma amostra extensa de pontos escolhidos sobre as superfícies, é possível que não seja encontrado um ponto inicial para alguma curva de intersecção. Além disso, quando ambas as superfícies forem tangentes ao longo de curvas inteiras pertencentes à intersecção entre as superfícies, não será possível determinar nenhum ponto inicial. Pontos isolados de tangência entre as superfícies também podem ser perdidos.

Já o segundo problema dá-se quando do surgimento de singularidades matemáticas entre as equações da Caminhada. Ou seja, ao proceder-se com o traçado da curva de intersecção, atinge-se um ponto singular. Define-se ponto singular como aquele onde as duas superfícies possuem vetores normais nulos ou colineares. Isto acaba anulando  $N_n$  em ( 24 ), o que faz surgir um zero no denominador de ( 25 ). Ou seja, o traçado atinge um ponto além do qual o algoritmo não é capaz de prosseguir.

A proposta a ser apresentada neste trabalho busca fundamentalmente fornecer soluções que não tenham estas duas limitações. Neste sentido, serão propostos nos próximos capítulos um método eficiente para determinação dos pontos iniciais de cada curva e um algoritmo de Caminhada usando equações diferenciais para facilitar o tratamento dos pontos singulares.

## **2.7) Justificativa na escolha dos métodos implementados neste trabalho**

Além dos métodos apresentados nas seções 2.5 e 2.6 (a serem referidos como Subdivisões e [Hosaka92], respectivamente), também foram estudados os métodos apresentados em [Grandine97] (Subdivisão dos Domínios Paramétricos) e em [Hu97] (Análise dos Domínios Paramétricos) para determinação dos pontos iniciais, a serem utilizados em conjunto com a proposta da Caminhada com equações diferenciais definida em [Hu97] e em [Kriezis92]. Estas últimas propostas, escolhidas neste trabalho para serem desenvolvidas, implementadas e estendidas para lidar com superfícies NURBS, serão detalhadas ao longo do capítulo 3. Todavia, para justificar a escolha destes métodos

em detrimento dos anteriores, foi feita a Tabela 2.1, a qual enumera vantagens e desvantagens de cada proposta (algumas já mencionadas nas seções 2.5 e 2.6, outras a serem analisadas nas seções 3.1 e 3.2).

*Tabela 2.1: Comparação entre os métodos estudados.*

	[Hosaka92] (seção 2.6)	Subdivisões (seção 2.5)	[Grandine97] + Marching Com Equações diferenciais	[Hu97] + Marching Com Equações diferenciais
Determina todas as curvas em condições sem pontos singulares			✓	✓
Determina laços definidos por um ponto único			✓	✓
Problema com curvas de intersecção coincidentes com o ângulo de varredura $\theta$			✓	
Problema com superfícies tangentes ao longo de curvas a planos paralelos				✓
Problemas com curvas de intersecção contendo infinitos pontos singulares	✓	✓	✓	✓
Velocidade	4	1	2	3
Necessidade de fornecer um chute inicial	✓			



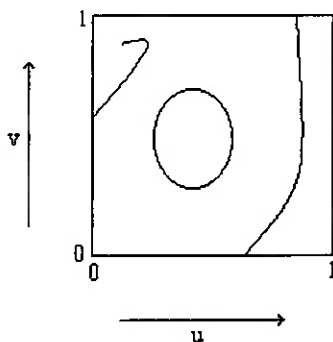
### 3) Estudo do Problema

Vamos definir o problema da determinação da curva de intersecção entre superfícies da seguinte forma: dadas  $F(u,v)$  e  $G(s,t)$  superfícies quaisquer parametrizadas no domínio  $[0,1]^2$ , deseja-se obter os pares  $(u,v)$  e  $(s,t)$  tal que

$$F(u,v) - G(s,t) = 0 \quad (33)$$

A expressão acima compreende, na verdade, um sistema de 3 equações (uma para cada coordenada do espaço) e 4 incógnitas, o que torna extremamente complexa uma solução direta e controlada para o mesmo. Para traçar as curvas de intersecção entre as superfícies, será utilizado o Método de Caminhada [Hu97], que a partir de um ponto inicial da curva de intersecção determina seus pontos seguintes, os quais satisfazem (33), através de procedimentos numérico-geométricos.

O conjunto solução de (33) pode consistir em nenhuma, uma ou mais curvas isoladas, como mostra a Figura 3.1. Para o Método da Caminhada deve-se encontrar ao menos um ponto inicial para cada curva deste conjunto solução. Este é o objetivo dos métodos apresentados nas seções 3.1 e 3.2.



*Figura 3.1: Exemplo de um conjunto solução da intersecção entre duas superfícies, formado por curvas disjuntas, vistas a partir do domínio paramétrico de uma das superfícies.*

### 3.1) Método da Subdivisão dos Domínios Paramétricos

Através do método apresentado por [Hu97], pode-se encontrar pontos iniciais para as curvas abertas (aquelas cujos pontos inicial e final no espaço paramétrico são distintos – é interessante notar que estes pontos podem corresponder ao mesmo ponto do espaço tridimensional) pertencentes ao conjunto solução do problema. Para tal iguala-se, em ( 33 ), cada parâmetro ( $u, v, s$  e  $t$ ) a zero e, depois, a um. Assim, após cada uma destas substituições, resta em ( 33 ) um sistema 3x3 que pode ser resolvido diretamente, conforme será detalhado em seção à frente. Feito isto, tem-se os pontos das extremidades de todas as curvas abertas do conjunto solução, uma vez que as mesmas sempre começam e terminam nas bordas dos domínios paramétricos de uma das superfícies do par em questão, como no exemplo da Figura 3.2.

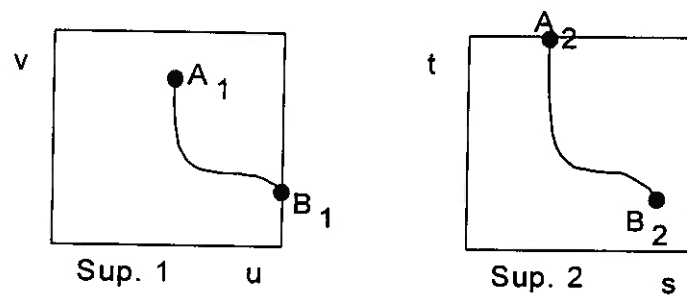


Figura 3.2: Exemplo de uma curva aberta de intersecção entre Sup. 1 e Sup. 2, vista a partir de cada domínio paramétrico, com os pontos das extremidades destacados. Notar que  $A_1$  e  $A_2$  representam o mesmo ponto no espaço 3D, bem como  $B_1$  e  $B_2$ .

Todavia, podem existir curvas de intersecção que sejam fechadas dentro do domínio paramétrico, de modo a não cruzarem as bordas dos mesmos. São os chamados laços de intersecção. Para traçar os laços, os mesmos devem ser quebrados em curvas abertas, identificando-se, então, os pontos das extremidades das mesmas. Mas para que possam ser quebrados, os laços devem ser, primeiro, detectados.

Em [Sederberg89], demonstra-se que se duas superfícies de continuidade ao menos  $C^1$  se interceptam ao longo de um laço, então existe uma reta perpendicular a

ambas superfícies e que atravessa o interior desta curva. O par formado pelos pontos, um de cada superfície, nos quais esta reta as intercepta é chamado par de *pontos com normais colineares*. Ou seja, o vetor normal de uma superfície no primeiro ponto é colinear ao da outra superfície no segundo ponto.

Definindo a normal sobre a superfície  $F(u, v)$  como

$$\mathbf{N}^F(u, v) = \frac{\mathbf{F}_u(u, v) \times \mathbf{F}_v(u, v)}{|\mathbf{F}_u(u, v) \times \mathbf{F}_v(u, v)|} \quad (34)$$

e a normal sobre a superfície  $G(s, t)$  como

$$\mathbf{N}^G(s, t) = \frac{\mathbf{G}_s(s, t) \times \mathbf{G}_t(s, t)}{|\mathbf{G}_s(s, t) \times \mathbf{G}_t(s, t)|} \quad (35)$$

Com objetivo de simplificação, a derivada parcial  $\mathbf{F}_u(u, v)$  da superfície  $F$  na direção  $u$  será doravante referida apenas por  $\mathbf{F}_u$ , o mesmo valendo para  $\mathbf{F}_v(u, v)$ ,  $\mathbf{G}_s(s, t)$  e  $\mathbf{G}_t(s, t)$ , e para as normais  $\mathbf{N}^F(u, v)$ ,  $\mathbf{N}^G(s, t)$ .

Associando as expressões ( 34 ) e ( 35 ) com a condição de colinearidade antes descrita, pode-se, então, determinar os pares de pontos com normais colineares entre as superfícies resolvendo-se as seguintes equações:

$$\mathbf{F}_u \circ (\mathbf{G}_s \times \mathbf{G}_t) = 0 \quad (36)$$

$$\mathbf{F}_v \circ (\mathbf{G}_s \times \mathbf{G}_t) = 0 \quad (37)$$

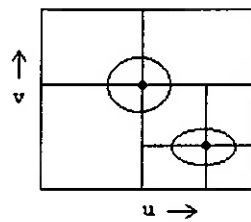
$$(\mathbf{F} - \mathbf{G}) \circ \mathbf{F}_u = 0 \quad (38)$$

$$(\mathbf{F} - \mathbf{G}) \circ \mathbf{F}_v = 0 \quad (39)$$

As equações ( 36 ) a ( 39 ) formam um sistema 4x4 que pode ser resolvido, fornecendo como resultado os pontos com normais colineares sobre cada domínio paramétrico. A partir de cada ponto encontrado, é possível subdividir sucessivamente o domínio paramétrico de modo a quebrar as curvas fechadas de intersecção em curvas abertas cujos pontos iniciais serão os pontos de cruzamento das mesmas com as bordas de cada novo subdomínio, conforme mostra o exemplo da Figura 3.3. Os pontos iniciais

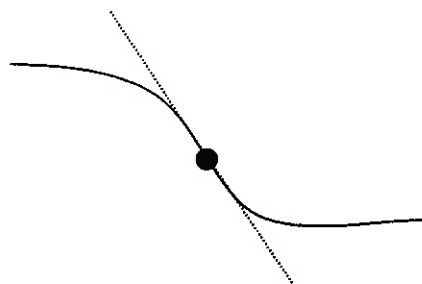
podem assim ser obtidos pelo método descrito anteriormente, ou seja, substituindo-se na equação ( 33 ) cada parâmetro pelo valor das bordas de cada novo subdomínio.

Na seção 3.4, discutiremos um método para solucionar sistemas de equações polinomiais não-lineares.



*Figura 3.3: domínio paramétrico subdividido através dos pontos com normais colineares assinalados.*

É interessante notar que pontos onde as superfícies são tangentes (e que, portanto, são pontos de intersecção) são também soluções para o sistema 4x4 formado pelas equações ( 36 ) a ( 39 ), já que em pontos de tangência as normais das superfícies são colineares. Outros pontos de intersecção também podem ser soluções para este sistema 4x4, como, por exemplo, pontos de inflexão (vide Figura 3.4).



*Figura 3.4: Exemplo de intersecção num ponto de inflexão.*

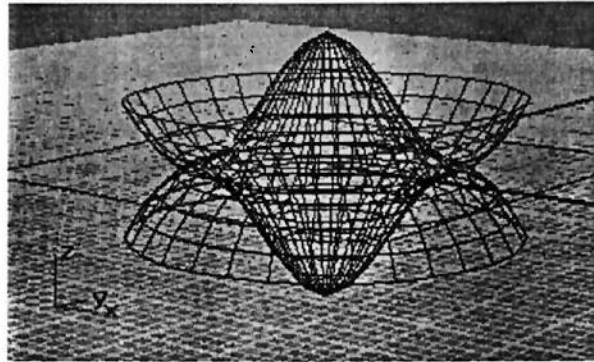
Desta maneira, algumas soluções do sistema 4x4 podem ser curvas de intersecção degeneradas em pontos. Neste caso, as superfícies se interceptam e são tangentes uma à outra no ponto de intersecção. Todavia, é possível haver situações em

que as superfícies são tangentes ao longo de toda a curva de intersecção. Todos os pontos destas curvas serão soluções para o sistema das equações ( 36 ) a ( 39 ). Isto significa que existirão infinitas soluções para este sistema 4x4. Se estas curvas são abertas, pode-se determinar os pontos de intersecção das extremidades das curvas, conforme foi descrito no início desta seção, perdendo-se, contudo, possíveis laços comuns de intersecção. No entanto, se existirem laços de intersecção ao longo dos quais as superfícies são tangentes (como na Figura 3.5), o passo de quebrá-los em curvas abertas também fica comprometido. Como o sistema 4x4 das equações ( 36 ) a ( 39 ) apresentaria infinitas soluções, estes laços seriam perdidos. Mais ainda, este problema pode ocorrer mesmo que as superfícies não sejam tangentes, bastando que elas tenham planos de tangência paralelos ao longo de laços em cada superfície (vide Figura 3.6). Por exemplo, pode-se imaginar um par de superfícies que sejam tangentes ao longo de um laço e que este laço pertença a um plano. Caso estas superfícies sejam deslocadas na direção do vetor normal a este plano, o sistema de ( 36 ) a ( 39 ) continuaria a apresentar infinitas soluções (ainda que as superfícies sequer tivessem qualquer ponto de intersecção).

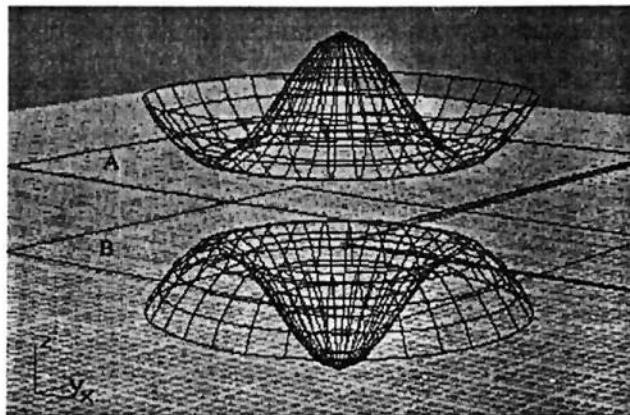
Uma possível solução para alguns problemas acima descritos, proposta em [Hu97], seria resolver um sistema de 5 equações e 4 incógnitas composto por ( 33 ), ( 36 ) e ( 37 ), o qual forneceria como solução somente os pontos que, além de possuírem normais colineares, também fossem pontos de intersecção. Contudo, possíveis laços de intersecção que não fossem fruto de tangência entre as superfícies poderiam ser perdidos, já que os demais pontos colineares normais não seriam determinados. Além disso, o emprego de tal técnica pressupõe saber-se, a priori, se as superfícies se interceptam ao longo de laços de tangência, o que representa tarefa intrincada.

É importante observar que, na ausência de curvas completas com pontos singulares, determinaremos não só, a partir de um ponto com normal colinear de uma superfície, pontos que pertencem ao laço de intersecção, mas também pontos que não obrigatoriamente pertencem a este laço, como mostra a Figura 3.7. No caso (a) da mesma, as superfícies apresentam dois pares de pontos com normais colineares para um laço. A partir destes, foram determinados 6 pontos pertencentes ao laço de intersecção e 2

pertencentes à curva aberta de intersecção. Já no caso (b), apesar de ter sido identificado um par de pontos com normais colineares, não existe laço de intersecção. Ainda assim, foram determinados dois pontos pertencentes à curva aberta de intersecção.



*Figura 3.5: Exemplo de superfícies tangentes ao longo de um laço de intersecção.*



*Figura 3.6: A superfície superior é tangente ao plano A ao longo de um laço de intersecção, o mesmo ocorrendo com a superfície inferior e o plano B, sendo A e B paralelos.*

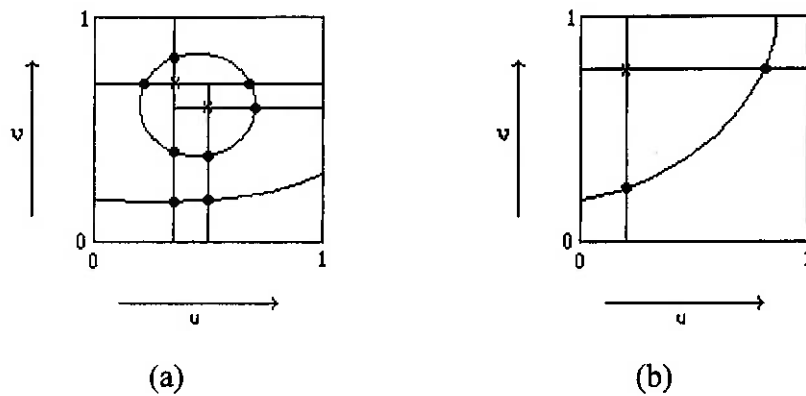


Figura 3.7: Exemplos de casos passíveis de ocorrer na aplicação do método.

### 3.2) Método da Análise dos Domínios Paramétricos

Esta proposta, apresentada em [Grandine97], também tem por objetivo determinar pontos iniciais para as curvas de intersecção, a serem usados no método da Caminhada, a ser detalhado na seção 3.5. Assim, dadas  $F(u,v)$  e  $G(s,t)$  superfícies parametrizadas no domínio  $[0,1]^2$ , deseja-se obter o conjunto  $u, v, s$  e  $t$  tal que:

$$F(u,v) - G(s,t) = 0 \quad (40)$$

ou seja,

$$F(u(\tau),v(\tau)) - G(s(\tau),t(\tau)) = 0 \quad (41)$$

Para determinar os pontos das curvas de intersecção que cruzam as bordas dos domínios paramétricos de  $F$  e  $G$  iguala-se cada parâmetro a zero ou um em (40) e resolve-se o sistema 3x3 restante. Para  $u = 0, 1$  e  $v = 0, 1$ , são obtidos pontos iniciais das curvas de intersecção que começam ou terminam nas bordas do domínio paramétrico de  $F$  e que correspondem a pontos quaisquer no domínio de  $G$ . Já para os casos  $s = 0, 1$  e  $t = 0, 1$  são obtidos pontos iniciais de curvas que podem começar em qualquer ponto do domínio de  $F$  mas que estão necessariamente sobre as bordas do domínio de  $G$ . Desta forma, os casos para as bordas de  $F$  e  $G$  devem ser processados separadamente.

Assim, supondo que todos os zeros para as bordas do domínio paramétrico de  $F$  tenham sido determinados ( $u = 0, 1$  e  $v = 0, 1$  em (40)), obtendo-se o resultado da Figura 3.8.

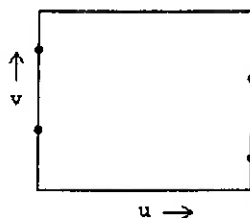


Figura 3.8: Pontos iniciais das curvas de intersecção sobre as bordas de um domínio paramétrico.



Então as possíveis topologias para as curvas de intersecção que cruzam as bordas deste domínio estão na Figura 3.8. Além desses casos, também pode haver curvas de intersecção que não cruzam nenhuma borda do domínio paramétrico. Tais curvas são chamadas de laços de intersecção, e podem ser em qualquer quantidade e estar em qualquer posição no interior do espaço paramétrico.

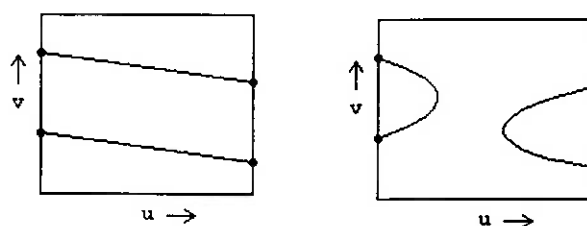


Figura 3.8: Possíveis topologias para as curvas de intersecção a partir dos pontos iniciais da Figura 3.6.

Para determinar tais laços, faz-se uma varredura (no espaço paramétrico) com linhas paralelas inclinadas de  $\theta$  em relação à direção do parâmetro  $v$ , conforme mostra a Figura 3.9.

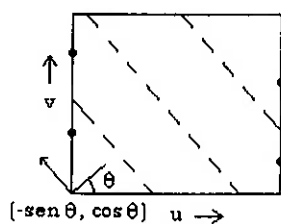


Figura 3.9: Linhas paralelas (pontilhadas) para varredura do espaço paramétrico.

Desta forma, o vetor  $(-\text{sen } \theta, \text{cos } \theta)$  será paralelo às linhas de varredura, sendo  $0 \leq \theta \leq \pi/2$ . Isto serve para determinar os pontos em que as curvas de solução fazem o retorno em relação às bordas. Chama-se estes pontos de *pontos de virada*, e eles caracterizam-se pelo fato de que a tangente à curva neste ponto é paralela à direção das linhas. No exemplo das curvas de intersecção da Figura 3.10, mostra-se como as linhas de varredura irão encontrar os pontos de virada tanto das curvas abertas quanto dos laços

de intersecção, detectando assim a presença destes últimos e fornecendo pontos iniciais para os mesmos.

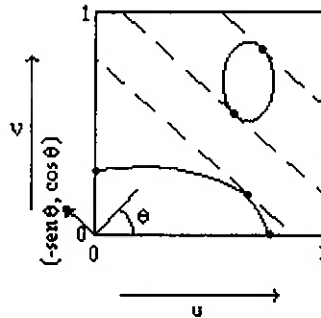


Figura 3.10: Exemplo de determinação dos pontos de virada das curvas de intersecção através das linhas de varredura.

Um ponto de virada, ou seja, o ponto em que a curva de intersecção fica tangente à linha de varredura, é caracterizado por:

$$u'(\tau) \cdot \cos \theta + v'(\tau) \cdot \sin \theta = 0 \quad (42)$$

ou seja,  $(u'(\tau), v'(\tau))$  é perpendicular a  $(\cos \theta, \sin \theta)$ .

Diferenciando a equação ( 41 ):

$$F_u u'(\tau) + F_v v'(\tau) - G_s s'(\tau) - G_t t'(\tau) = 0 \quad (43)$$

Daí, tem-se que o vetor  $(u'(\tau), v'(\tau), s'(\tau), t'(\tau))$  está no espaço nulo da matriz  $3 \times 4$   $[F_u \ F_v \ -G_s \ -G_t]$ . O espaço nulo desta matriz é dado por um múltiplo do seguinte vetor:

$$\begin{pmatrix} \det[F_v \ G_s \ G_t] \\ -\det[F_u \ G_s \ G_t] \\ -\det[F_u \ F_v \ G_t] \\ \det[F_u \ F_v \ G_s] \end{pmatrix} \quad (44)$$

Assim, de ( 42 ), ( 43 ) e ( 44 ), tem-se que os pontos de virada são dados pelas soluções do seguinte sistema  $4 \times 4$ :

$$\begin{aligned} \mathbf{F}(u, v) - \mathbf{G}(s, t) &= 0 \\ (\mathbf{F}_u \sin \theta - \mathbf{F}_v \cos \theta) \circ \mathbf{G}_s \times \mathbf{G}_t &= 0 \end{aligned} \quad (45)$$

Nota-se que o conjunto solução deste sistema deve conter também todos os pontos em que as curvas de intersecção se encontram, uma vez que as mesmas só se cruzam em *ponto críticos*, nos quais o gradiente da função desaparece, sendo uma solução para o sistema acima.

Em última análise, a idéia principal deste método é encontrar pontos iniciais para cada curva do conjunto solução do problema de intersecção. Com isto, simplifica-se a solução em si, posto que se adiciona uma equação à expressão original ( 40 ) do problema, com 3 equações e 4 incógnitas, originando um sistema 4x4 ( 45 ). É interessante notar que tal sistema serve fundamentalmente para tratar dos chamados laços de intersecção, posto que para encontrar os pontos iniciais das curvas abertas do conjunto solução bastaria determinar os pontos em que tais curvas cruzam as bordas dos domínios paramétricos das superfícies em questão, ou seja, as soluções da expressão ( 40 ) com cada variável recebendo os valores zero e um (o que resulta em diversos sistemas 3x3).

Todavia, como os laços de intersecção não cruzam as bordas dos domínios paramétricos, é necessária a estratégia apresentada para determinar pontos iniciais sobre os mesmos, quebrando, com isto, cada laço em 2 curvas abertas, cujos extremos são definidos justamente por este par de pontos de virada.

Assim, nota-se que devem ser escolhidos métodos numéricos adequados para resolver os sistemas de polinômios cujas soluções irão representar os pontos iniciais desejados, os quais serão classificados em listas que definem a relação entre eles, possibilitando uma análise topológica completa das curvas de intersecção vista do domínio paramétrico, o que já representa uma vantagem na manipulação das mesmas. Ou seja, vale ressaltar que este método, da mesma forma que o apresentado na seção 3.1, também irá envolver a solução de sistemas de equações não-lineares. Para realizar tal tarefa, será utilizado o método do poliedro projetado, o qual será discutido na seção 3.4.

Uma dificuldade na utilização do método do poliedro projetado é a transformação dos polinômios dos sistemas para a base de Bernstein. Um procedimento com este objetivo será detalhado na seção 3.3.

Contudo, há um caso especial o qual aumenta significativamente a complexidade do algoritmo apresentado, que ocorre quando as superfícies são tangentes ao longo de curvas inteiras do conjunto solução, e não apenas em pontos isolados. Quando duas superfícies são tangentes em determinado ponto, o vetor normal de ambas no mesmo é paralelo. Definindo a normal sobre a superfície  $F$  como

$$\mathbf{N}^F = \frac{\mathbf{F}_u \times \mathbf{F}_v}{|\mathbf{F}_u \times \mathbf{F}_v|}$$

e a normal sobre a superfície  $G$  como

$$\mathbf{N}^G = \frac{\mathbf{G}_s \times \mathbf{G}_t}{|\mathbf{G}_s \times \mathbf{G}_t|}$$

então temos que se  $\mathbf{N}^F$  for paralelo a  $\mathbf{N}^G$ , então os vetores  $\mathbf{F}_u$ ,  $\mathbf{F}_v$ ,  $\mathbf{G}_s$  e  $\mathbf{G}_t$  são coplanares. Desta maneira, também serão coplanares os três vetores que formam o produto misto da segunda equação do sistema ( 45 ), de forma que o resultado deste produto apresenta valor nulo. Ou seja, também serão soluções para o sistema ( 45 ) os pontos de tangência entre as superfícies. Estes pontos, quando ocorrem isolados, podem ser bem identificados e tratados, representando tão somente um acréscimo finito no número de soluções do sistema ( 45 ). Entretanto, se houver curvas ao longo das quais as superfícies sejam tangentes, então cada ponto de cada uma destas curvas será solução para o sistema ( 45 ), de forma que o número de soluções para o mesmo torna-se infinito.

Outra questão delicada é a escolha de  $\theta$  para as linhas de varredura, pois se uma curva intersecção estiver coincidente com a linha de varredura, então o problema torna-se extremamente complexo, pois o sistema ( 45 ) passa a ter infinitas soluções.

Esta última limitação é adicional em relação ao método apresentado na seção 3.1. Contudo, esta proposta apresenta também algumas vantagens sobre aquela. A começar do problema da tangência, que neste método se limita a laços de intersecção,

enquanto naquele estende-se a planos paralelos de tangência ao longo de laços para cada superfície. Além disso, o grau máximo das equações do sistema definido por ( 45 ) é menor que a do sistema ( 36 ) a ( 39 ), o que facilita a priori o procedimento de solução numérica através de uma diminuição bastante significativa no seu tempo de processamento.

### 3.3) Transformando um Polinômio para a Base de Bernstein

Conforme observado em [Sherbrooke93], ao utilizar o método do poliedro projetado para solucionar sistemas polinomiais não-lineares é necessário que todo polinômio do sistema seja convertido para um polinômio equivalente em base de Bernstein. Uma proposta neste sentido é apresentada a seguir.

Conforme visto na seção 2.3, a base de Bernstein é usada como função-base para a representação de curvas e superfícies de Bézier. Nesta seção 3.3.1 e no início da 3.3.2, apresentam-se inicialmente algumas abordagens anteriores para a conversão de polinômios de uma e duas variáveis. Finalmente, uma abordagem alternativa para a conversão de polinômios com duas variáveis é proposta na seção 3.3.2, representando uma contribuição original deste trabalho, e, seguindo a mesma lógica, um algoritmo genérico para a conversão de polinômios com múltiplas variáveis é apresentado na seção 3.3.3.

#### 3.3.1) Convertendo Polinômios com uma Variável para a Base de Bernstein [Hoschek93]

Supondo que se deseja converter o polinômio de grau  $m$  abaixo, parametrizado por  $u$ ,  $0 \leq u \leq 1$ , para a base de Bernstein:

$$P(u) = \sum_{k=0}^m a_k \cdot u^k \quad (46)$$

Um polinômio equivalente escrito na base de Bernstein segue a formato abaixo:

$$P(u) = \sum_{i=0}^m w_i \cdot B_i^m(u) = \sum_{i=0}^m w_i \cdot \binom{m}{i} \cdot (1-u)^{m-i} \cdot u^i \quad (47)$$

onde  $w_i$  é o peso associado com o polinômios de Bernstein  $B_i^m(u)$ .

Desta maneira, é razoável dizer que converter um polinômio na base de potências para a base de Bernstein significa, em outras palavras, determinar os pesos  $w_i$  na expressão ( 47 ) de modo que a igualdade ( 46 ) = ( 47 ) seja verdadeira.

Todavia, usando-se o binômio de Newton, sabe-se que:

$$(1-u)^{m-i} = \sum_{j=0}^{m-i} \binom{m-i}{j} \cdot (-1)^j \cdot u^j \quad (48)$$

Substituindo ( 48 ) em ( 47 ), obtém-se:

$$P(u) = \sum_{i=0}^m \sum_{j=0}^{m-i} w_i \cdot \binom{m}{i} \cdot \binom{m-i}{j} \cdot (-1)^j \cdot u^{i+j} \quad (49)$$

Fazendo  $j = g - i$  em ( 49 ), tem-se:

$$P(u) = \sum_{i=0}^m \sum_{g=i}^m w_i \cdot \binom{m}{i} \cdot \binom{m-i}{g-i} \cdot (-1)^{g-i} \cdot u^g \quad (50)$$

Como  $\binom{m-i}{g-i} = \frac{\binom{m}{g} \binom{g}{i}}{\binom{m}{i}}$ , reordenando o somatório a expressão anterior fica:

$$P(u) = \sum_{g=0}^m \sum_{i=0}^g w_i \cdot \binom{m}{g} \cdot \binom{g}{i} \cdot (-1)^{g-i} \cdot u^g \quad (51)$$

Com isto, podem ser comparados os elementos de ( 46 ) e ( 51 ) por potências de  $u$ , o que resulta em:

$$a_g = \sum_{i=0}^g w_i \cdot \binom{m}{g} \cdot \binom{g}{i} \cdot (-1)^{g-i} \quad (52)$$

Isto significa que é possível determinar cada peso  $w_g$  em ( 47 ), com  $0 \leq g \leq m$ , seguindo um procedimento que começa com a obtenção de  $w_0$ , a partir do qual  $w_1$  é calculado, e assim por diante até que a determinação do último peso  $w_n$ , que irá depender de todos os pesos anteriormente calculados. Este procedimento é representado pela seguinte expressão:

$$w_g = \frac{a_g}{\binom{m}{g}} - \sum_{i=0}^{g-1} w_i \cdot \binom{g}{i} \cdot (-1)^{g-i} \quad (53)$$

Representando os pesos  $w_g$  dentro de um vetor  $\mathbf{W}$ , é possível visualizar como seu elemento genérico  $w_g$  é obtido a partir de uma combinação linear entre o termo  $a_g$  de (46) e os termos do subconjunto  $A$  mostrado na Figura 3.11.

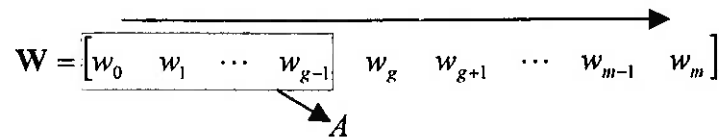


Figura 3.11: Vetor de pesos  $\mathbf{W}$ , indicando que o valor do peso  $w_g$  depende dos valores dos pesos  $w_0$  até  $w_{g-1}$ , como exposto pela expressão (53).

Esta seqüência inspira o seguinte algoritmo para determinar os pesos  $w$ :

**Dados**

$a(i)$ : função que retorna o coeficiente linear do monômio de grau  $i$  do parâmetro  $u$ ;  
 $\text{comb}(i, j)$ : função que retorna o coeficiente binomial  $\binom{i}{j}$ ;  
 $\text{powe}(i, j)$ : função que retorna  $i$  elevado a  $j$ ;  
 $m$ : grau máximo do parâmetro  $u$  no polinômio;  
 $g, i$ : variáveis auxiliares para os índices dos somatórios;  
 $W[0..m]$ : vetor de pesos  $w$ , inicializado com zeros.

```

For g = 0 until m
  For i = 0 until g
    if ((i = g))
      then
        W[g] = W[g] + a(g) / (comb(m, g));
    else
        W[g] = W[g] - W[i] * comb(g, i) * powe((-1), (g-i));
  
```

### 3.3.2) Convertendo Polinômios de duas Variáveis para a Base de Bernstein

Para converter o seguinte polinômio de duas variáveis:

$$P(u_1, u_2) = \sum_{g_1=0}^{d_1} \sum_{g_2=0}^{d_2} a_{g_1 g_2} \cdot u_1^{g_1} \cdot u_2^{g_2} \quad (54)$$



[Hoschek93] propôs uma abordagem matricial. Reescrevendo ( 54 ) desta maneira, tem-se:

$$P(u_1, u_2) = \mathbf{u}_1^T \mathbf{A} \mathbf{u}_2 \quad (55)$$

onde  $\mathbf{u}_1 = (1, u_1, \dots, u_1^{d_1})^T$ ,  $\mathbf{u}_2$  é definido similarmente, e  $\mathbf{A} = \begin{pmatrix} a_{00} & a_{01} & \dots & a_{0d_2} \\ \vdots & & & \vdots \\ a_{d_1 0} & a_{d_1 1} & \dots & a_{d_1 d_2} \end{pmatrix}$ .

Um polinômio equivalente a ( 54 ) escrito na base de Bernstein seria:

$$P(u_1, u_2) = \sum_{i_1=0}^{d_1} \sum_{i_2=0}^{d_2} w_{i_1 i_2} \cdot B_{i_1}^{d_1}(u_1) \cdot B_{i_2}^{d_2}(u_2) \quad (56)$$

ou, em termos de matrizes,

$$P(u_1, u_2) = \mathbf{u}_1^T [\mathbf{BU1}] \mathbf{W} [\mathbf{BU2}]^T \mathbf{u}_2 \quad (57)$$

onde  $\mathbf{W}$  é a matriz dos pesos  $w_{i_1 i_2}$ , os quais se deseja determinar, e  $[\mathbf{BU1}]$  e  $[\mathbf{BU2}]$  são as matrizes de coeficientes dos polinômios de Bernstein em termos dos monômios de  $u_1$  e  $u_2$ , respectivamente. Comparando ( 55 ) e ( 57 ), obtém-se:

$$\mathbf{A} = [\mathbf{BU1}] \mathbf{W} [\mathbf{BU2}]^T \quad (58)$$

ou,

$$\mathbf{W} = [\mathbf{BU1}]^{-1} \mathbf{A} ([\mathbf{BU2}]^T)^{-1} \quad (59)$$

Assim, para determinar a matriz  $\mathbf{W}$  deve-se antes executar um procedimento para montar cada matriz de coeficientes  $[\mathbf{BU1}]$  e  $[\mathbf{BU2}]$ , os quais variam de acordo com os graus  $d_1$  e  $d_2$  respectivamente, para, finalmente, poder computar o produto destas com a matriz  $\mathbf{A}$ .

Todavia, um método menos custoso computacionalmente e mais geral pode ser adotado se for estendida a idéia da seção anterior para uma abordagem bidimensional. Usando a mesma lógica da seqüência de ( 46 ) até ( 51 ), obtém-se a seguinte expressão ( 56 ):

$$P(u_1, u_2) = \sum_{g_1=0}^{d_1} \sum_{g_2=0}^{d_2} \sum_{i_1=0}^{g_1} \sum_{i_2=0}^{g_2} w_{i_1 i_2} \cdot \binom{d_1}{g_1} \cdot \binom{d_2}{g_2} \cdot \binom{g_1}{i_1} \cdot \binom{g_2}{i_2} \cdot (-1)^{g_1+g_2-i_1-i_2} \cdot u_1^{g_1} \cdot u_2^{g_2} \quad (60)$$

Desta forma, para determinar cada peso  $w_{g_1 g_2}$  em ( 56 ), com  $0 \leq g_1 \leq d_1$  e  $0 \leq g_2 \leq d_2$ , uma vez encontrados os pesos  $w_{00}, w_{01}, \dots, w_{g_1-1, g_2-1}$ , pode-se usar:

$$w_{g_1 g_2} = \frac{a_{g_1 g_2}}{\binom{d_1}{g_1} \cdot \binom{d_2}{g_2}} - \sum_{i_1=0}^{g_1-1} \sum_{i_2=0}^{g_2-1} w_{i_1 i_2} \cdot \binom{g_1}{i_1} \cdot \binom{g_2}{i_2} \cdot (-1)^{g_1+g_2-i_1-i_2} \quad (61)$$

Representando os pesos  $w_{g_1 g_2}$  dentro de um vetor  $\mathbf{W}$ , é possível visualizar que seu elemento genérico  $w_{g_1 g_2}$  é obtido através da combinação linear entre o termo  $a_{g_1 g_2}$  de ( 54 ) e os termos do subconjunto  $A$  mostrado na Figura 3.12.

$$\mathbf{W} = \begin{bmatrix} w_{00} & w_{01} & \dots & w_{0(g_2-1)} & w_{0g_2} & w_{0(g_2+1)} & \dots & w_{0(d_2-1)} & w_{0d_2} \\ w_{10} & w_{11} & \dots & w_{1(g_2-1)} & w_{1g_2} & w_{1(g_2+1)} & \dots & w_{1(d_2-1)} & w_{1d_2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{(g_1-1)0} & w_{(g_1-1)1} & \dots & w_{(g_1-1)(g_2-1)} & w_{(g_1-1)g_2} & w_{(g_1-1)(g_2+1)} & \dots & w_{(g_1-1)(d_2-1)} & w_{(g_1-1)d_2} \\ w_{g_1 0} & w_{g_1 1} & \dots & w_{g_1(g_2-1)} & w_{g_1 g_2} & w_{g_1(g_2+1)} & \dots & w_{g_1(d_2-1)} & w_{g_1 d_2} \\ w_{(g_1+1)0} & w_{(g_1+1)1} & \dots & w_{(g_1+1)(g_2-1)} & w_{(g_1+1)g_2} & w_{(g_1+1)(g_2+1)} & \dots & w_{(g_1+1)(d_2-1)} & w_{(g_1+1)d_2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{(d_1-1)00} & w_{(d_1-1)1} & \dots & w_{(d_1-1)(g_2-1)} & w_{(d_1-1)g_2} & w_{(d_1-1)(g_2+1)} & \dots & w_{(d_1-1)(d_2-1)} & w_{(d_1-1)d_2} \\ w_{d_1 0} & w_{d_1 1} & \dots & w_{d_1(g_2-1)} & w_{d_1 g_2} & w_{d_1(g_2+1)} & \dots & w_{d_1(d_2-1)} & w_{d_1 d_2} \end{bmatrix}$$

Figura 3.12: Matriz de pesos  $\mathbf{W}$ .

Assim, dados  $a_{g_1 g_2}$ , coeficiente do termo  $u_1^{g_1} u_2^{g_2}$ , e os pesos  $w_{i_1 i_2}$  do subconjunto  $A$  acima, obtém-se o peso  $w_{g_1 g_2}$  comparando ( 54 ) e ( 60 ) em termos do produto dos monômios de  $u_1$  e  $u_2$ , extrapolando para o caso bidimensional o procedimento anteriormente apresentado para uma variável.

Neste caso, um algoritmo para determinar os pesos  $w$  poderia ser o seguinte:



Com a mesma lógica apresentada nos casos anteriores e extrapolando aquele procedimento para o caso  $n$ -dimensional, é possível desenvolver um algoritmo para cada caso quando necessário, a partir da seguinte expressão para um elemento genérico da matriz de pesos  $W$ :

$$W_{g_1 g_2 \dots g_n} = \frac{a_{g_1 g_2 \dots g_n}}{\binom{d_1}{g_1} \cdot \binom{d_2}{g_2} \dots \binom{d_n}{g_n}} - \sum_{i_1=0}^{g_1-1} \sum_{i_2=0}^{g_2-1} \dots \sum_{i_n=0}^{g_n-1} W_{i_1 i_2 \dots i_n} \cdot \binom{g_1}{i_1} \cdot \binom{g_2}{i_2} \dots \binom{g_n}{i_n} \cdot (-1)^{g_1+g_2+\dots+g_n-i_1-i_2-\dots-i_n} \quad (64)$$

Um algoritmo genérico e recursivo para determinar os pesos  $w$  seria:

**Dados**

$n$ : o número de variáveis no polinômio;  
 $g[1..n], i[1..n]$ : vetores de variáveis auxiliares para os índices dos somatórios;  
 $d[1..n]$ : vetor com o grau máximo de cada variável no polinômio;  
 $W[0..d[1], 0..d[2], \dots, 0..d[n]]$ : matriz de pesos  $w$ ;

$get\_a((array) g, n)$ : função que retorna o coeficiente linear do termo de graus  $g[1], g[2], \dots, g[n]$  das variáveis  $u_1, u_2, \dots, u_n$ ;

$comb((array) i, (array) j, n)$ : função que retorna o produto dos coeficiente binomiais  $\binom{i[1]}{j[1]} \cdot \binom{i[2]}{j[2]} \dots \binom{i[n]}{j[n]}$ ;

$powe(i, j)$ : função que retorna  $i$  elevado  $j$ ;

$sum((array) i, n)$ : função que retorna a soma  $i[1]+i[2]+\dots+i[n]$ ;

$comp((array) i, (array) j, n)$ : função booleana que retorna TRUE se  $i[k]=j[k]$  para qualquer inteiro  $k$  entre 0 e  $n$ , e FALSE caso contrário;

$set\_W((array) g, n, value)$ : função que atribui o valor  $value$  para o elemento  $W[g[1], g[2], \dots, g[n]]$  da matriz  $W$ ;

$get\_W((array) g, n)$ : função que retorna o elemento  $W[g[1], g[2], \dots, g[n]]$  da matriz  $W$ ;

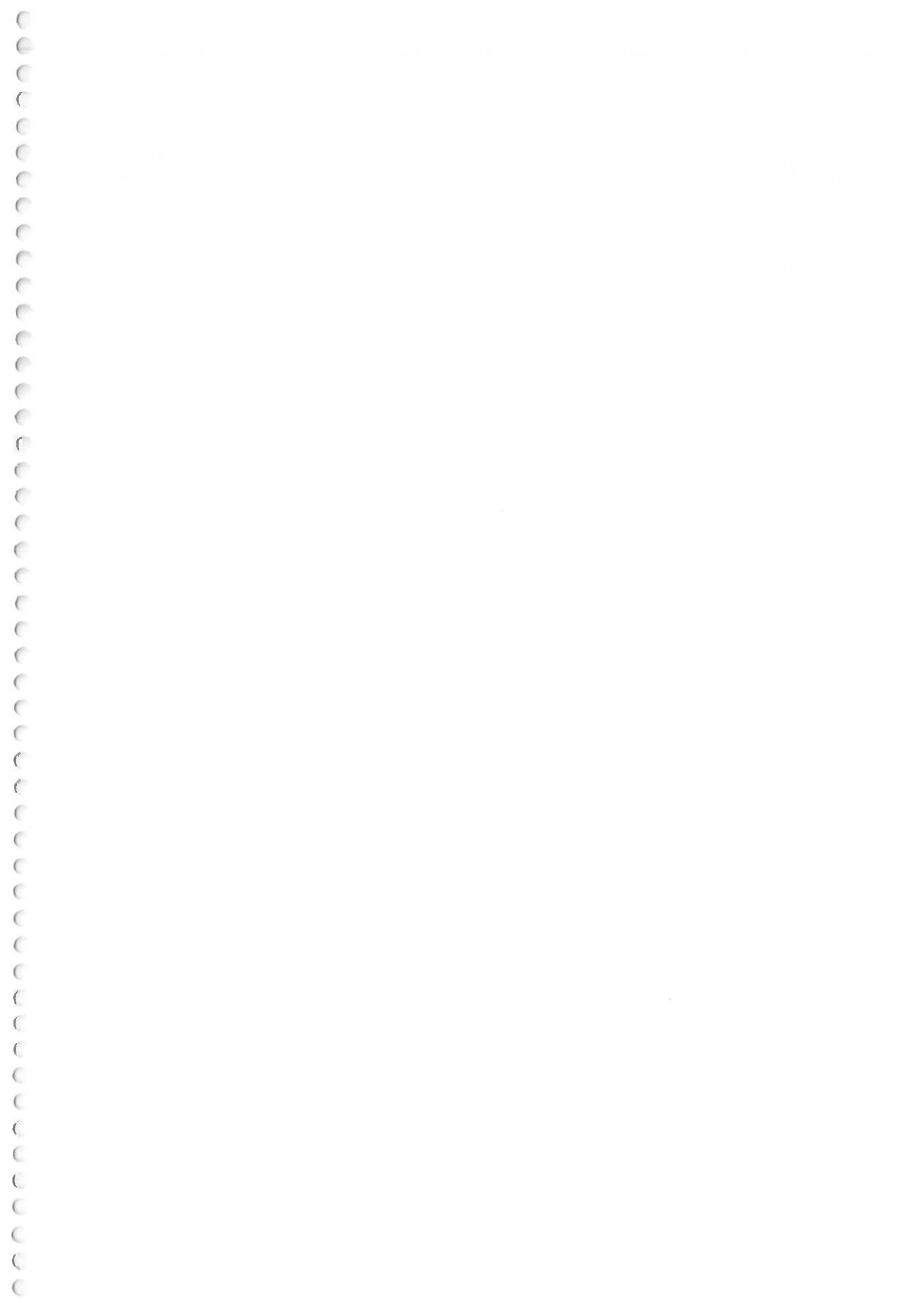
Compute(level)

```

For g[level] = 0 until d[level]
  For i[level] = 0 until g[level]
    if level = n
      then
        if comp(g, i, n)
          then
            set_W(g, n, (get_W(g, n) + get_a(g, n) / comb(d, g, n)));
          else
            set_W(g, n, (get_W(g, n) - get_W(i, n) * comb(g, i, n) *
              powe((-1), sum(g, n) - sum(i, n)));
        else
          Compute(level+1);

```

Compute(1);



### 3.4) Solucionando Sistemas Polinomiais Não-Lineares

Solucionar os sistemas polinomiais apresentados nas seções 3.1 e 3.2 significa, em termos algébricos, determinar todas as n-uplas  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  tal que

$$f_1(\mathbf{x}) = f_2(\mathbf{x}) = \dots = f_n(\mathbf{x}) = 0 \quad (65)$$

sendo  $x_i \in [0,1]^{\dagger}$  e  $f_k$  um polinômio com  $n$  variáveis.

A seguir, apresenta-se um método para a solução destes sistemas.

#### 3.4.1) Método do Poliedro Projetado

O algoritmo do poliedro projetado está apresentado integralmente em [Sherbrooke93;Zhou93], sendo que nesta seção ele será descrito de forma resumida, com um exemplo de aplicação ao final. O primeiro passo a ser realizado neste método consiste em transformar cada equação  $f_k$  para um polinômio em Bernstein (vide seção 3.3), ou seja:

$$f_k(\mathbf{x}) = \sum_{i_1=0}^{d_1^{(k)}} \sum_{i_2=0}^{d_2^{(k)}} \dots \sum_{i_n=0}^{d_n^{(k)}} [w_{i_1 i_2 \dots i_n}^{(k)}] \cdot B_{i_1}^{d_1^{(k)}}(x_1) \cdot B_{i_2}^{d_2^{(k)}}(x_2) \dots B_{i_n}^{d_n^{(k)}}(x_n) \quad (66)$$

onde  $d_i^{(k)}$  é o grau da variável  $x_i$  em  $f_k$ .

Definindo-se um multi-índice  $I$  como uma n-upla ordenada de inteiros não-negativos  $(i_1, i_2, \dots, i_n)$ , usando-se a notação  $\sum_I^D$  na qual, por exemplo, se  $D = (1,1)$ , então  $\sum_I^D w_I \equiv w_{00} + w_{01} + w_{10} + w_{11}$ , e definindo-se também um multinômio de Bernstein como sendo

$$B_I^D(\mathbf{u}) = B_{i_1}^{d_1}(u_1) \cdot B_{i_2}^{d_2}(u_2) \dots B_{i_n}^{d_n}(u_n) \quad (67)$$

então pode-se escrever (66) de maneira mais simplificada:

---

<sup>†</sup> ou qualquer outro domínio finito  $[a_i, b_i]$ , o qual deve, então, ser transportado para o domínio  $[0,1]$  através de uma transformação de variáveis  $x_i' = a_i + x_i \cdot (b_i - a_i)$  em todas as equações  $f_k$ .

$$f_k(\mathbf{x}) = \sum_I^{D^{(k)}} w_I^{(k)} \cdot B_I^{D^{(k)}}(\mathbf{x}) \quad (68)$$

Logo, fazer a transformação de um polinômio para o polinômio equivalente em base de Bernstein consiste em obter a matriz (cuja dimensão é o número de variáveis da equação) formada pelos pesos  $w_I$  para cada equação  $f_k$ . Na seção 3.3 foi mostrado como determinar esta matriz.

Feito isto, temos então todas as equações  $f_k$  na forma exibida em (68). Agora, vamos redefinir o problema da solução do sistema de equações de (65) como a tarefa de determinar a intersecção do grafo<sup>‡</sup> de cada  $f_k$  (cada grafo sendo uma hipersuperfície em  $\mathfrak{R}^{n+1}$ ) e o hiperplano  $x_{n+1} = 0$ . Cada grafo de  $f_k$  é dado por

$$\mathbf{F}_k(\mathbf{x}) = (x_1, x_2, \dots, x_n, f_k(\mathbf{x})) = (\mathbf{x}, f_k(\mathbf{x})) \quad (69)$$

de modo que o sistema de (65) passa a ser descrito por:

$$\mathbf{F}_1(\mathbf{x}) = \mathbf{F}_2(\mathbf{x}) = \dots = \mathbf{F}_n(\mathbf{x}) = (\mathbf{x}, 0) \quad (70)$$

Pode ser demonstrado (vide Apêndice A1.1) que

$$\sum_{i=0}^n \frac{i}{n} B_i^n(u) = u \quad (71)$$

ou seja, o monômio  $u$  pode ser expresso como uma soma ponderada dos polinômios de Bernstein com coeficientes igualmente espaçados no intervalo  $[0,1]$ . Então, usando (67), (71) e a propriedade (11), é possível obter uma expressão (vide Apêndice A1.2) equivalente a cada  $x_j$  em (69):

$$x_j = \sum_I^{D^{(k)}} \frac{i_j}{d_j^{(k)}} B_I^{D^{(k)}}(\mathbf{x}) \quad (72)$$

a qual, usada em (69), dá origem a:

$$\mathbf{F}_k(\mathbf{x}) = \sum_I^{D^{(k)}} \mathbf{v}_I^{(k)} \cdot B_I^{D^{(k)}}(\mathbf{x}) \quad (73)$$

onde

<sup>‡</sup> Seja  $f: \mathfrak{R}^n \rightarrow \mathfrak{R}$  uma função de  $\mathbf{x}$ . Então, o grafo de  $f$  é a função  $F: \mathfrak{R}^n \rightarrow \mathfrak{R}^{n+1}$  definida por:  $F(\mathbf{x}) = (\mathbf{x}, f(\mathbf{x}))$ .

$$\mathbf{v}_I^{(k)} = \left( \frac{i_1}{d_1^{(k)}}, \frac{i_2}{d_2^{(k)}}, \dots, \frac{i_n}{d_n^{(k)}}, w_I^{(k)} \right) \quad (74)$$

sendo  $\mathbf{v}_I^{(k)}$  os chamados pontos de controle de  $\mathbf{F}_k$ .

Ou seja, até este ponto, tudo o que foi feito consiste em reescrever o problema em termos de hipersuperfícies paramétricas  $\mathbf{F}_k$ , de modo a obter os pontos de controle de (74), o que irá permitir o uso da propriedade do convex-hull da base de Bernstein, como será visto a seguir.

Como dito em (65), as  $n$  variáveis  $x_i$  a serem determinadas mapeiam um box do espaço  $n$ -dimensional definido por:

$$S = [0,1] \times [0,1] \times \dots \times [0,1] = [0,1]^n \quad (75)$$

dentro do qual estão contidas as soluções desejadas para o sistema. Estando o mesmo reescrito na forma de (70), pode-se, então, utilizar as propriedades da base de Bernstein para eliminar recursivamente partes do box definido em (75) que não possuam soluções para o sistema. Inicialmente, deve-se definir duas propriedades dos multinômios de Bernstein, obtidas por extrapolação imediata de (10) e (11), respectivamente:

$$B_I^M(\mathbf{u}) \geq 0 \quad \forall \mathbf{u} \in [0,1]^n \quad (76)$$

e

$$\sum_I^{D^{(k)}} B_I^{D^{(k)}}(\mathbf{u}) = 1 \quad (77)$$

Estas duas equações levam a uma propriedade do convex-hull da base de Bernstein:

Teorema 1: Seja  $\mathbf{v}_I^{(k)}$  um conjunto de pontos de controle de  $\mathbf{F}_k$ ,  $C(\mathbf{v}_I^{(k)})$  o convex-hull deste conjunto e  $\mathbf{x} \in [0, 1]^n$ . Então,  $\forall k$  entre 1 e  $n$  inclusive,

$$\mathbf{F}_k(\mathbf{x}) \in C(\mathbf{v}_I^{(k)}) \quad (78)$$



pois, de ( 73 ), tem-se que  $F_k$  é uma combinação linear dos pontos de controle  $v_i^{(k)}$ , com coeficientes  $B_i^{D^{(k)}}(\mathbf{x})$ . Contudo, por ( 76 ) e ( 77 ), esses coeficientes são positivos e sua soma resulta em 1, satisfazendo a definição do convex-hull<sup>§</sup>.

Como todo ponto da hipersuperfície  $F_k$  está no interior do convex-hull de seus pontos de controle, a equação ( 70 ) leva ao seguinte teorema:

Teorema 2: Seja  $\mathbf{x} \in [0, 1]^n$  uma solução do sistema, ou seja, uma raiz comum a todos os  $f_k$ , e  $X_{n+1}$  o conjunto de pontos de  $\mathfrak{R}^{n+1}$  que estão sobre o hiperplano  $x_{n+1} = 0$ . Então,

$$(\mathbf{x}, 0) \in \bigcap_{k=1}^n C(v_i^{(k)}) \cap X_{n+1} \quad (79)$$

Daí, segue o corolário:

Corolário 1: Se  $(\mathbf{x}, 0)$  não satisfaz ( 79 ), então  $\mathbf{x}$  não é uma raiz comum para todos os  $f_k$ .

Baseado nisto, pode-se afirmar que a tarefa de solucionar o sistema de equações polinomiais foi transformada na de determinar a intersecção dos convex-hulls formados pelos pontos de controle de cada hipersuperfície  $F_k$ , na dimensão  $n + 1$ , sendo  $n$  o número de variáveis do sistema inicial. Contudo, é perfeitamente possível transformar este problema de dimensão  $(n + 1)$  em  $n$  problemas bidimensionais projetando os pontos de controle de cada  $F_k$  em cada plano formado pela coordenada relativa a uma variável e pela última coordenada, relativa à função original  $f_k$  (ou seja, um plano para cada variável do sistema inicial). Desta forma, os pontos de controle de ( 74 ) serão projetados em  $n$  planos, sendo mapeados num plano  $j$  como

---

<sup>§</sup> Supondo que  $P$  seja um conjunto de  $m$  pontos de  $\mathfrak{R}^n$ , o convex-hull de  $P$ ,  $C(P)$ , denota o conjunto de todos os  $\mathbf{x}$  tal que  $\mathbf{x} = \sum_{i=1}^m c_i \cdot \mathbf{p}_i$ , onde, para cada  $i$ ,  $\mathbf{p}_i \in P$  e  $c_i \geq 0$ , e  $\sum_{i=1}^m c_i = 1$ .

$$\mathbf{v}_{j,l}^{(k)} = \left( \frac{i_j}{d_j^{(k)}}, w_l^{(k)} \right) \quad (80)$$

A partir de então, forma-se, em cada plano, o convex-hull bidimensional dos pontos de controle de cada  $F_k$ . Faz-se, para cada plano, a intersecção de cada convex-hull com o segmento  $[0,1]$  (relativo à faixa da variável deste plano no box de solução de (75)) da abcissa do plano, sendo o resultado um segmento, um ponto ou vazio (pois os polígonos em questão são convexos). E, novamente para cada plano, faz-se a intersecção entre estes segmentos, o que pode resultar num segmento, num ponto, ou vazio. Se for vazio, não há valores para a variável deste plano os quais sejam soluções. Se o resultado for um ponto, então o mesmo é um valor de solução para aquela variável. Já para o caso de resultar um segmento, então pode haver, no interior do mesmo, um ou mais pontos que representem soluções para a variável.

Isto significa que, para cada variável  $x_i$ , teremos possivelmente reduzido a faixa de procura de  $[0,1]$  (conforme (75)) para o segmento encontrado, digamos,  $[a_i, b_i]$ . Tem-se, com isto, um novo box solução o qual irá conter as soluções procuradas:

$$S = [a_1, b_1] \times [a_2, b_2] \times \dots \times [a_n, b_n] \quad (81)$$

Neste ponto, para cada  $k$ , define-se uma nova função  $f_k'$  por:

$$f_k'(\mathbf{x}) = f_k(a_1 + (b_1 - a_1) \cdot x_1, a_2 + (b_2 - a_2) \cdot x_2, \dots, a_n + (b_n - a_n) \cdot x_n) \quad (82)$$

a qual vai ser mapeada pelo domínio  $[0,1]^n$  da mesma maneira que  $f_k$  era pelo domínio formado pelos  $[a_i, b_i]$  encontrados.

Repete-se todo o processo (determinar os pontos de controle dos grafos das novas funções  $f_k$ , projetá-los nos planos das variáveis e fazer as intersecções dos convex-hulls com a abcissa), até que se encontrem boxes de solução como o de (81) no qual todos os seus lados sejam suficientemente pequenos para serem considerados raízes isoladas das variáveis. Ou seja, em  $S$  (81),  $\forall i$  tal que  $1 \leq i \leq n$ ,  $(b_i - a_i) \leq \text{tolerância}$ . Esta *tolerância* irá definir a precisão da solução encontrada para o sistema (75).

Todavia, ao se realizar a intersecção dos convex-hulls, pode ocorrer do valor de  $(b_i - a_i)$ , em algumas variáveis de  $S$  (em ( 81 )), não cair para valores significativamente menores que 1. Isto ocorre sobretudo quando há mais de uma solução para o sistema inicial. Neste caso, deve-se subdividir em dois o segmento da variável que não sofreu redução considerável, tratando-se, a partir de então, os dois sub-boxes resultantes como problemas independentes. Por exemplo, se foi obtido  $S = [a_1, b_1] \times [a_2, b_2] \times \dots \times [a_n, b_n]$  e  $(b_2 - a_2) > \textit{critério}$ , então serão considerados dois sub-boxes para a próxima iteração:  $S = [a_1, b_1] \times \left[ a_2, \frac{(b_2 - a_2)}{2} \right] \times \dots \times [a_n, b_n]$  e  $S = [a_1, b_1] \times \left[ \frac{(b_2 - a_2)}{2}, a_2 \right] \times \dots \times [a_n, b_n]$ . É interessante notar que tal subdivisão será feita automaticamente até que se isole um sub-box que contenha uma só solução (ou que o sub-box tenha todos os seus lados menores que *tolerância*, mesmo que existam em seu interior mais de uma solução, as quais seriam todas consideradas como uma só). O valor ótimo de *critério* para uma melhor eficiência do algoritmo em relação a tempo de processamento varia de acordo com o número de equações do sistema e o grau máximo das variáveis nestas equações. Conforme sugerido em [Sherbrooke93], foi adotado  $\textit{critério} = 0.9$ .

### 3.4.2) Exemplo Prático

A título de exemplo, será resolvido agora um sistema extremamente simples de modo a ilustrar o funcionamento do método. Tem-se, então, como tarefa, encontrar as soluções para o seguinte sistema:

$$3x - 1 = 0$$

$$y - x = 0$$

ou seja, temos duas funções:

$$f_1 = 3x - 1$$

$$f_2 = y - x$$

sendo que o box de solução é definido por  $S = [0,1] \times [0,1]$ . Utilizando-se ( 63 ), pode-se determinar os pontos de controle das hipersuperfícies  $F_1$  e  $F_2$ , na forma de ( 74 ). Inicialmente, vamos determinar, através de ( 63 ), como seria um polinômio  $f$  de duas variáveis, com grau máximo igual a 1, em termos dos pontos de controle da base de Bernstein:

$$f = w_{00} \cdot B_0^1(x) \cdot B_0^1(y) + w_{01} \cdot B_0^1(x) \cdot B_1^1(y) + w_{10} \cdot B_1^1(x) \cdot B_0^1(y) + w_{11} \cdot B_1^1(x) \cdot B_1^1(y)$$

ou seja,

$$f = (w_{00}) \cdot x^0 \cdot y^0 + (w_{01} - w_{00}) \cdot x^0 \cdot y^1 + (w_{10} - w_{00}) \cdot x^1 \cdot y^0 + (w_{11} - w_{10} - w_{01} + w_{00}) \cdot x^1 \cdot y^1$$

Comparando-se termo a termo em ordem crescente  $f$  com  $f_1$ , chega-se a:

$$w_{00} = -1$$

$$(w_{01} - w_{00}) = 0 \Rightarrow w_{01} = -1$$

$$(w_{10} - w_{00}) = 3 \Rightarrow w_{10} = 2$$

$$(w_{11} - w_{10} - w_{01} + w_{00}) = 0 \Rightarrow w_{11} = 2$$

como pontos de controle de  $f_1$  em base de Bernstein.

Analogamente para  $f_2$ , são obtidos:

$$w_{00} = 0$$

$$(w_{01} - w_{00}) = 1 \Rightarrow w_{01} = 1$$

$$(w_{10} - w_{00}) = -1 \Rightarrow w_{10} = -1$$

$$(w_{11} - w_{10} - w_{01} + w_{00}) = 0 \Rightarrow w_{11} = 0$$

como pontos de controle de  $f_2$  em base de Bernstein.

Assim, encontra-se os seguintes pontos de controle para as hipersuperfícies (usando ( 74 )):

Para  $F_1$ :

$$\mathbf{v}_{00}^{(1)} = (0,0,-1)$$

$$\mathbf{v}_{01}^{(1)} = (0,1,-1)$$

$$\mathbf{v}_{10}^{(1)} = (1,0,2)$$

$$\mathbf{v}_{11}^{(1)} = (1,1,2)$$

Para  $F_2$ :

$$\mathbf{v}_{00}^{(2)} = (0,0,0)$$

$$\mathbf{v}_{01}^{(2)} = (0,1,1)$$

$$\mathbf{v}_{10}^{(2)} = (1,0,-1)$$

$$\mathbf{v}_{11}^{(2)} = (1,1,0)$$

Projetando os pontos de controle de  $F_1$  no plano relativo à variável  $x$  conforme pode ser visto na Figura 3.13, obtém-se o segmento resultante para  $x$  como sendo  $[1/3, 1/3]$ .

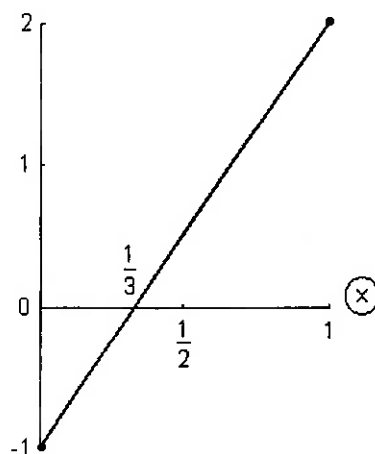


Figura 3.13: *Projeção dos pontos de controle de  $F_1$  para a variável  $x$ .*

Já fazendo a projeção dos pontos de controle de  $F_2$  em relação ao plano da variável  $x$  conforme mostrado na Figura 3.14, tem-se que o segmento resultante é o  $[0,1]$ . Fazendo  $[1/3, 1/3] \cap [0,1] = [1/3, 1/3]$ , chega-se ao novo valor da variável  $x$  no box de solução  $S$  ( $x$  foi determinado).

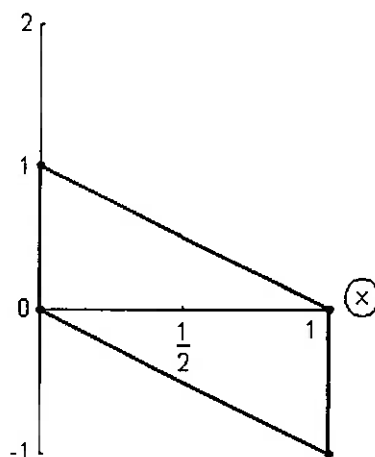


Figura 3.14: *Projeção dos pontos de controle de  $F_2$  para a variável  $x$ .*

Já projetando  $F_1$  e  $F_2$  para  $y$ , resulta nos gráficos das Figuras 3.15 e 3.16, respectivamente.

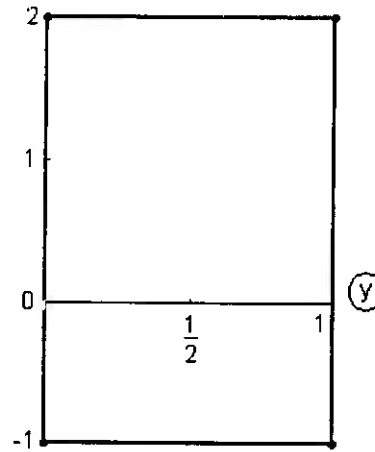


Figura 3.15: Projeção dos pontos de controle de  $F_1$  para a variável  $y$ .

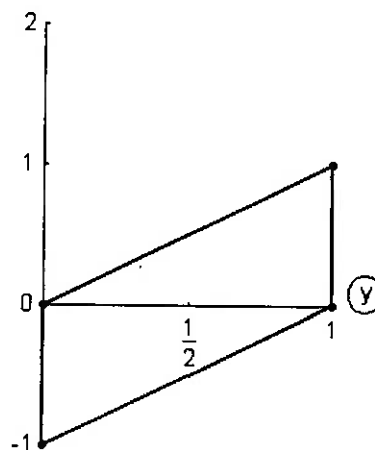


Figura 3.16: Projeção dos pontos de controle de  $F_2$  para a variável  $y$ .

Ou seja, a faixa de procura para  $y$  permanece sendo  $[0,1]$ . Temos, por conseguinte, um novo box de solução:  $S = [1/3, 1/3] \times [0,1]$ . Como o lado do box relativo à variável  $y$  não decresceu em comprimento para um valor menor que *critério* (ou 0.9), divide-se o box resultante em dois, a serem considerados como dois problemas independentes na próxima iteração:  $S = [1/3, 1/3] \times [0, 1/2]$  e  $S = [1/3, 1/3] \times [1/2, 1]$ .

Vamos considerar então o primeiro sub-box,  $S = [1/3, 1/3] \times [0, 1/2]$ . Devemos realizar a transformação indicada por ( 82 ) para que as funções  $f_1$  e  $f_2$  sejam mapeadas novamente por um domínio  $[0, 1] \times [0, 1]$ , mas com um comportamento equivalente ao do interior deste sub-box. Assim, as transformações serão:  $x = 1/3$  e  $y = 1/2 y'$ , o que resulta nas seguintes funções:

$$f_1' = 0$$

$$f_2' = \frac{1}{2} y' - \frac{1}{3}$$

de modo que, como a primeira equação já está determinada, só se faz necessário utilizar o grafo de  $f_2$ . Os pontos de controle de  $F_2$  serão:

$$v_{00}^{(2)} = (0, 0, -1/3)$$

$$v_{01}^{(2)} = (0, 1, 1/6)$$

$$v_{10}^{(2)} = (1, 0, -1/3)$$

$$v_{11}^{(2)} = (1, 1, 1/6)$$

Como a variável  $x$  já está determinada, só interessa realizar a projeção dos pontos de controle no plano de  $y'$ , o que resulta no gráfico da Figura 3.17, cuja intersecção com a abcissa ocorre no ponto  $2/3$ .

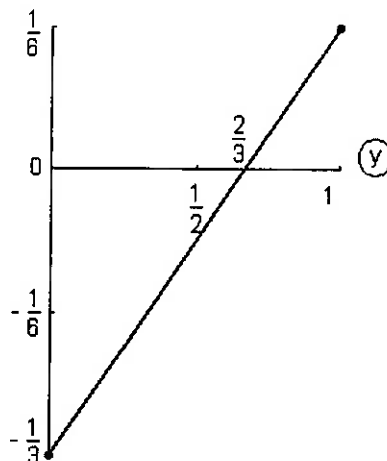


Figura 3.17: Projeção dos pontos de controle de  $F_2$  para a variável  $y'$ .

Ou seja, o segmento resultante para  $y'$  é  $[2/3, 2/3]$ . Fazendo, então, a transformação inversa em  $y'$ , posto que  $y = \frac{1}{2} y'$ , obtém-se o valor para a variável  $y$  original. Com isto, foi determinada a solução  $x = 1/3$  e  $y = 1/3$  para o sistema inicialmente proposto.

Resta ainda vasculhar o outro sub-box,  $S = [1/3, 1/3] \times [1/2, 1]$ . Realizando a transformação de coordenadas, chega-se a:

$$\begin{aligned}f_1' &= 0 \\f_2' &= \frac{1}{2} y' + \frac{1}{6}\end{aligned}$$

e os pontos de controle para  $F_2$  serão:

$$v_{00}^{(2)} = (0, 0, 1/6)$$

$$v_{01}^{(2)} = (0, 1, 2/3)$$

$$v_{10}^{(2)} = (1, 0, 1/6)$$

$$v_{11}^{(2)} = (1, 1, 2/3)$$

sendo imediato observar que a projeção dos mesmos no plano de  $y'$  não cruza a abcissa (posto que os valores da última coordenada são todos positivos), de forma que não há solução para o sistema neste sub-box, como era de se esperar.



### 3.5) Determinando a Curva de Intersecção a partir de Pontos Singulares da mesma através do Método da Caminhada com equações diferenciais

Uma vez obtidos os pontos iniciais do conjunto solução, deseja-se traçar cada curva  $C(w)$ , parametrizada pelo comprimento de arco, através do método de Caminhada proposto por [Hu97].

A direção de caminhada (marching) é dada pela tangente de  $C(w)$ , que é perpendicular à normal de ambas as superfícies no ponto em questão:

$$C' = \frac{dC}{dw} = \frac{N^F \times N^G}{|N^F \times N^G|} \quad (83)$$

Diferenciando as superfícies em relação ao parâmetro  $w$ , obtém-se:

$$\begin{aligned} \frac{dC}{dw} &= \frac{du}{dw} \cdot F_u + \frac{dv}{dw} \cdot F_v \\ \frac{dC}{dw} &= \frac{ds}{dw} \cdot F_s + \frac{dt}{dw} \cdot F_t \end{aligned} \quad (84)$$

Manipulando (vide Apêndice A2.1):

$$\begin{aligned} \frac{du}{dw} &= -\frac{Det[F_v \ C' \ N^F]}{|F_u \times F_v|}, & \frac{dv}{dw} &= \frac{Det[F_u \ C' \ N^F]}{|F_u \times F_v|} \\ \frac{ds}{dw} &= -\frac{Det[G_t \ C' \ N^G]}{|F_u \times F_v|}, & \frac{dt}{dw} &= \frac{Det[G_s \ C' \ N^G]}{|F_u \times F_v|} \end{aligned} \quad (85)$$

que são as equações a partir das quais os pontos da curva de intersecção podem ser obtidos sucessivamente por algum método numérico de integração do problema do valor inicial para este sistema de equações diferenciais ordinárias (85). Para realizar tal tarefa, utiliza-se o método Runge-Kutta de 4ª ordem (vide Apêndice 3).

Este algoritmo de caminhada possui os seguintes passos:

1. Obtém pontos iniciais para o algoritmo, ou seja, pelo menos um ponto para cada curva de intersecção. Isto é feito através do procedimento apresentado na seção 3.1 ou do método da seção 3.2.
2. Para cada um destes pontos, obtém-se as derivadas dos parâmetros das superfícies em relação ao comprimento de arco da curva de intersecção através de ( 85 ).
3. Integra-se este sistema de equações diferenciais ordinárias através de algum método de integração numérica (no nosso caso, utilizamos o Runge-Kutta de 4ª ordem – Apêndice 3), obtendo-se, assim, o próximo ponto da curva de intersecção.
4. Repete-se os passos 2 e 3 até que o traçado atinja as bordas do subdomínio a que a curva pertence, no caso do método da seção 3.1, ou até que se atinja um ponto de virada ou de borda, no caso do método da seção 3.2.

### 3.5.1) Casos especiais: Situações de Indeterminação e a função Distância Orientada

Contudo, há diversos casos especiais que podem ser encontrados ao se traçar as curvas, os quais resultam na indeterminação da equação ( 83 ), quando, via de regra, ambas superfícies possuem normais paralelas no ponto da curva em questão. Para tratar estes problemas, pode-se utilizar a função distância orientada como método alternativo para obter os valores das derivadas normalmente determinadas por ( 85 ).

A função real distância orientada  $\phi$  entre a superfície  $G$  e um ponto movendo-se sobre a superfície  $F$ , apresentada em [Wang91; Kriezis92], é definida por:

$$\phi(u, v) = \mathbf{N}^G[\mathbf{g}(\mathbf{F}(u, v))] \circ [\mathbf{F}(u, v) - \mathbf{G}(\mathbf{g}(\mathbf{F}(u, v)))] \quad (86)$$

onde  $\mathbf{g}$  é a função  $\mathbb{R}^3 \rightarrow [0,1]^2$  que obtém os valores dos parâmetros  $s$  e  $t$  de  $G$  que definem o ponto sobre esta superfície o qual estará mais próximo de um dado ponto em  $\mathbb{R}^3$ ,  $\mathbf{G}(\mathbf{g}(\mathbf{F}(u, v)))$  é um ponto sobre a superfície  $G$  o qual está mais próximo ao ponto definido por  $\mathbf{F}(u, v)$ , e  $\mathbf{N}^G[\mathbf{g}(\mathbf{F}(u, v))]$  é o vetor normal unitário na superfície  $G$  naquele mesmo ponto. Assim,  $|\phi(u, v)|$  denota a distância euclidiana real entre o ponto  $\mathbf{F}(u, v)$  e

$\mathbf{G}(\mathbf{g}(\mathbf{F}(u,v)))$ , desde que os vetores  $\mathbf{N}^G[\mathbf{g}(\mathbf{F}(u,v))]$  e  $[\mathbf{F}(u,v) - \mathbf{G}(\mathbf{g}(\mathbf{F}(u,v)))]$  sejam colineares (*condição de colinearidade*). Neste caso,  $\mathbf{G}(\mathbf{g}(\mathbf{F}(u,v)))$  é a projeção ortogonal do ponto  $\mathbf{F}(u,v)$  sobre a superfície  $\mathbf{G}$ , de modo que as derivadas de primeira e segunda ordem de  $\phi$  podem ser determinadas explicitamente. Daí, a intersecção entre as superfícies é equivalente ao conjunto de pontos que satisfazem a equação implícita  $\phi(u,v) = 0$ .

Utilizando as derivadas de primeira e segunda ordem de  $\phi$ , desenvolvem-se equações para obtenção dos valores das derivadas do sistema ( 85 ) nos pontos em que há indeterminação. Ou seja, nos pontos em que os vetores normais das duas superfícies são colineares.

Neste caso, quando durante a aplicação do método Marching através das equações de ( 85 ) o algoritmo obtém um ponto que define uma indeterminação em ( 83 ) para o cálculo do próximo ponto, basta resolver a seguinte equação:

$$\frac{d^2\phi}{dw^2} = \begin{bmatrix} du & dv \\ dw & dw \end{bmatrix} \cdot \mathbf{H} \cdot \begin{bmatrix} du & dv \\ dw & dw \end{bmatrix}^T = 0 \quad (87)$$

onde  $\mathbf{H}$  é o Hessiano de  $\phi$ , ou seja

$$\mathbf{H} = \begin{bmatrix} \phi_{uu} & \phi_{uv} \\ \phi_{uv} & \phi_{vv} \end{bmatrix} \quad (88)$$

Os valores da segunda derivada de  $\phi$  são (vide Apêndice A2.2):

$$\phi_{uu} = \mathbf{N}_u^G \circ \mathbf{F}_u + \mathbf{N}^G \circ \mathbf{F}_{uu} \quad (89)$$

$$\phi_{uv} = \mathbf{N}_v^G \circ \mathbf{F}_u + \mathbf{N}^G \circ \mathbf{F}_{uv} \quad (90)$$

$$\phi_{vv} = \mathbf{N}_v^G \circ \mathbf{F}_v + \mathbf{N}^G \circ \mathbf{F}_{vv} \quad (91)$$

As derivadas parciais das normais com parâmetros trocados acima ficam (vide Apêndice A2.4):

$$\mathbf{N}_u^G = \begin{bmatrix} \mathbf{N}_s^G & \mathbf{N}_t^G \end{bmatrix} \mathbf{L}^{-1} \begin{bmatrix} \mathbf{F}_u \circ \mathbf{G}_s & \mathbf{F}_u \circ \mathbf{G}_t \end{bmatrix}^T \quad (92)$$

$$\mathbf{N}_v^G = \begin{bmatrix} \mathbf{N}_s^G & \mathbf{N}_t^G \end{bmatrix} \mathbf{L}^{-1} \begin{bmatrix} \mathbf{F}_v \circ \mathbf{G}_s & \mathbf{F}_v \circ \mathbf{G}_t \end{bmatrix}^T \quad (93)$$

onde  $\mathbf{L}$  é o tensor de projeção ortogonal da superfície  $\mathbf{G}(s,t)$  dado por

$$L_{ij} = \mathbf{G}_i \circ \mathbf{G}_j - (\mathbf{F} - \mathbf{G}) \circ \mathbf{G}_{ij} \text{ com } i, j = s, t. \quad (94)$$

As derivadas parciais diretas das normais ficam (vide Apêndice A2.4):

$$\mathbf{N}_s^G = \frac{\left( \mathbf{G}_{ss} \times \mathbf{G}_t + \mathbf{G}_s \times \mathbf{G}_{st} - \frac{\partial}{\partial s} (\mathbf{G}_s \times \mathbf{G}_t) \right) \cdot \mathbf{N}^G}{|\mathbf{G}_s \times \mathbf{G}_t|} \quad (95)$$

$$\mathbf{N}_t^G = \frac{\left( \mathbf{G}_{st} \times \mathbf{G}_t + \mathbf{G}_s \times \mathbf{G}_{tt} - \frac{\partial}{\partial t} (\mathbf{G}_s \times \mathbf{G}_t) \right) \cdot \mathbf{N}^G}{|\mathbf{G}_s \times \mathbf{G}_t|} \quad (96)$$

Analisando-se o sinal do determinante de  $\mathbf{H}$ , chega-se a três possibilidades:

- i) se  $\det[\mathbf{H}] > 0$  então indica um ponto de tangência isolado entre as superfícies;
- ii) se  $\det[\mathbf{H}] = 0$  então indica um *cuspid point* (Figura 3.18 (a));
- iii) se  $\det[\mathbf{H}] < 0$  então indica um *ponto de sela* (Figura 3.18 (b)).

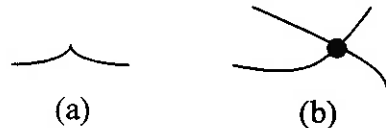


Figura 3.18: Exemplos de *cuspid point* e *saddle point* (ponto de sela).

Daí, uma solução geral de (87) para o caso de  $\det[\mathbf{H}] \leq 0$  será

$$\begin{bmatrix} \frac{du}{dw} & \frac{dv}{dw} \end{bmatrix}^T = \lambda_1 \cdot \begin{bmatrix} -\phi_{uv} \pm \sqrt{\det[\mathbf{H}]} & \phi_{uu} \end{bmatrix}^T \quad (97)$$

sendo que para obter  $\lambda_1$  impõe-se que  $\begin{bmatrix} \frac{du}{dw} & \frac{dv}{dw} \end{bmatrix} = 1$ .

Em seguida, determina-se o valor das derivadas de  $s$  e  $t$  através de (vide Apêndice A2.3):

$$\begin{bmatrix} \frac{ds}{dw} & \frac{dt}{dw} \end{bmatrix}^T = \mathbf{K}^{-1} \mathbf{E} \cdot \begin{bmatrix} \frac{du}{dw} & \frac{dv}{dw} \end{bmatrix}^T \quad (98)$$

onde  $\mathbf{K}$  é o primeiro tensor fundamental de  $\mathbf{G}(s,t)$ , dado por

$$K_{ij} = \mathbf{G}_i \circ \mathbf{G}_j \quad \text{com } i, j = s, t \quad (99)$$

e  $\mathbf{E}$  vale

$$E_{ab} = \mathbf{F}_b \circ \mathbf{G}_a \quad \text{com } b = u, v \text{ e } a = s, t. \quad (100)$$

Há, também, o caso, mencionado no fim da seção 3.1, em que as superfícies são tangentes ao longo de uma curva de intersecção aberta. Nesta situação as equações de ( 85 ) não podem ser utilizadas, posto que as mesmas são indeterminadas ao longo de toda a curva.

Ou seja,  $\det[\mathbf{H}] = 0$  ao longo da curva inteira. Com isto, chega-se a

$$\begin{bmatrix} \frac{du}{dw} & \frac{dv}{dw} \end{bmatrix}^T = \lambda_1 \cdot \begin{bmatrix} -\phi_{uv} & \phi_{uu} \end{bmatrix}^T \quad (101)$$

ou

$$\begin{bmatrix} \frac{du}{dw} & \frac{dv}{dw} \end{bmatrix}^T = \lambda_1 \cdot \begin{bmatrix} \phi_{vv} & -\phi_{uv} \end{bmatrix}^T \quad (102)$$

Determinando-se  $\lambda_1$  (impondo que  $\left[ \begin{bmatrix} \frac{du}{dw} & \frac{dv}{dw} \end{bmatrix} \right] = 1$ ) e o valor das derivadas de  $s$  e  $t$ , da mesma maneira feita em ( 98 ), tem-se as derivadas para serem usadas na Caminhada.

### 3.6) Implementação e Exemplos Com Superfícies de Bézier

Tudo o que foi exposto nas seções 3.1, 3.2, 3.3, 3.4 e 3.6 foi implementado e testado. Tal tarefa foi realizada em linguagem C, inicialmente com o auxílio de um compilador Borland C e, posteriormente, com um compilador Visual C++, para usufruir das vantagens da memória virtual do ambiente Windows. Os testes foram realizados num Pentium 200 MHz e num Pentium II 300 MHz.

O programa implementado lê de um arquivo os pontos de controle das superfícies de Bézier cuja intersecção se deseja determinar. A partir de então, monta-se as equações desejadas. Todas as equações polinomiais envolvidas são armazenadas em listas ligadas cujo conteúdo é o da Figura 3.19 que mostra um elemento da lista ligada o qual irá representar o termo genérico

$\dots + \text{coeficiente\_linear} \times \text{Variavel\_1}^{\text{Expoente\_1}} \times \text{Variavel\_2}^{\text{Expoente\_2}} \times \dots \times \text{Variavel\_n}^{\text{Expoente\_n}} + \dots$  de uma equação polinomial qualquer.

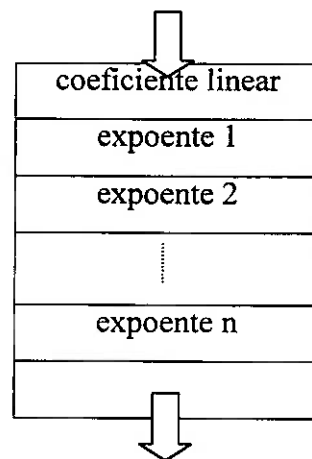


Figura 3.19: Elemento da lista ligada que armazena os polinômios no sistema.

As equações convenientes (( 36 ) a ( 39 ) para o Método da Subdivisão ou o sistema ( 45 ) para o Método da Análise) formam um sistema 4x4 utilizado para a detecção dos laços de intersecção. Nas rotinas implementadas, tal sistema foi solucionado através do método do poliedro projetado com *tolerância* 0.0001. Já os sistemas 3x3 resultantes das substituições dos valores das bordas também foram resolvidos com

auxílio do método do poliedro projetado, mas com *tolerância*  $10^{-6}$ . Tendo-se obtido os pontos iniciais para as curvas, basta aplicar o método da Caminhada para traçar as mesmas.

Foram resolvidos diversos exemplos para teste, alguns dos quais seguem abaixo. Através deles, fica clara a maior eficiência do Método da Análise dos Domínios Paramétricos sobre o Método da Subdivisão dos Domínios Paramétricos. Percebe-se que o tempo de processamento do primeiro é bastante menor que o do segundo, devido sobretudo ao grau inferior das equações polinomiais dos sistemas envolvidos no Método da Análise. Além disso, o Método da Subdivisão acaba por gerar um número bastante redundante de pontos iniciais, o que é causado pelo fato de, dependendo do grau das superfícies envolvidas, haver mais de um par de pontos com normais colineares no interior de um laço de intersecção, gerando uma quantidade desnecessariamente alta de pontos iniciais para tal laço.

O primeiro exemplo é relativo às superfícies de Bézier apresentadas na Figura 3.20. As Figuras 3.21 e 3.22 apresentam os pontos iniciais obtidos através dos Métodos da Subdivisão e da Análise, respectivamente. A comparação dos tempos de execução de cada método comprovam o que já era previsto teoricamente: o Método da Análise, por envolver equações polinomiais de menor grau no sistema a ser resolvido numericamente, é significativamente mais rápido que o Método das Subdivisões. Na Figura 3.23 pode-se ver o traçado das curvas de intersecção obtidas através do Método da Caminhada.

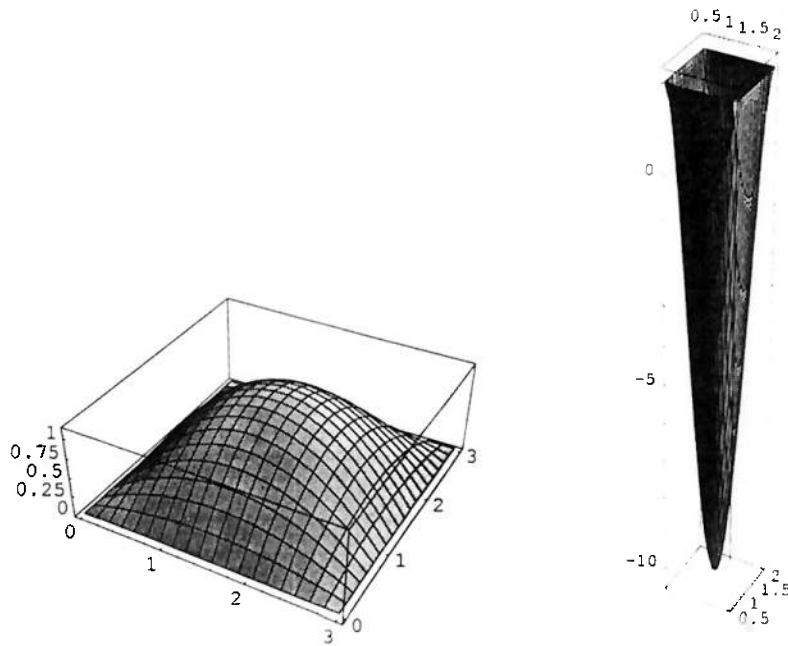


Figura 3.20: Superfícies de Bézier cuja intersecção se deseja determinar.

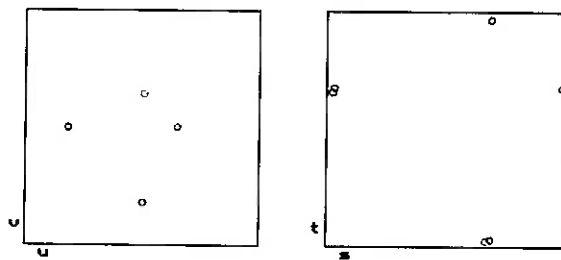
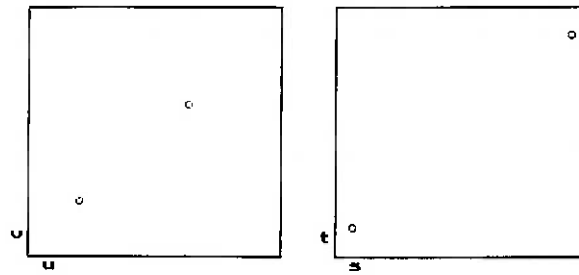


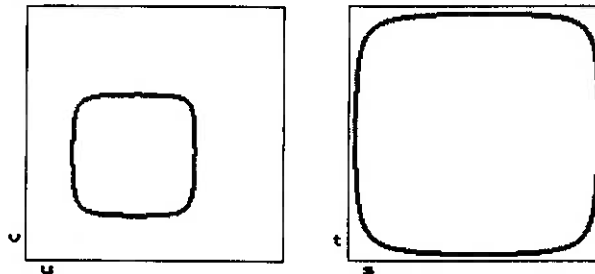
Figura 3.21: Pontos iniciais relativos à Figura 3.20 obtidos através do Método da Subdivisão dos Domínios Paramétricos. Tempo para determinação destes pontos:

59.130s.





*Figura 3.22: Pontos iniciais relativos à Figura 2.20 obtidos através do Método da Análise dos Domínios Paramétricos. Tempo para determinação destes pontos: 4.610s.*



*Figura 3.23: Curvas de intersecção entre as superfícies da Figura 3.20, traçadas através do Método da Caminhada com equações diferenciais.*

O segundo exemplo é relativo às superfícies de Bézier apresentadas na Figura 3.24. As Figuras 3.25 e 3.26 apresentam os pontos iniciais obtidos através dos Métodos da Subdivisão e da Análise, respectivamente. Aqui, o Método da Análise apresenta a outra de suas vantagens: ele é menos redundante, ou seja, encontra um número menor de pontos iniciais para as curvas, de modo que seu tempo de execução fica ainda mais breve quando comparado ao tempo de execução do Método da Subdivisão. Na Figura 3.27 pode-se ver o traçado das curvas de intersecção obtidas.

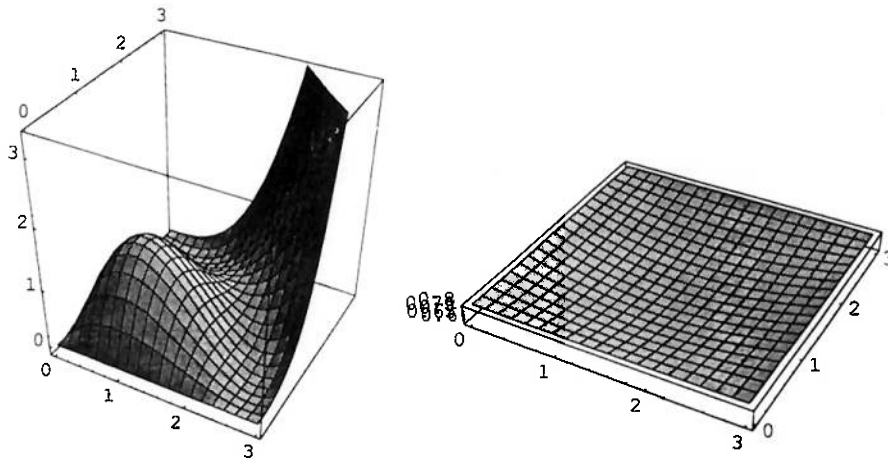


Figura 3.24: Superfícies de Bézier cuja intersecção se deseja determinar.

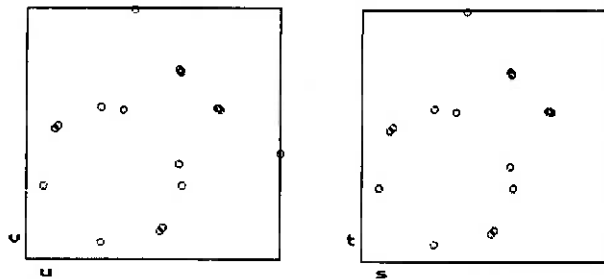


Figura 3.25: Pontos iniciais relativos à Figura 3.24 obtidos através do Método da Subdivisão dos Domínios Paramétricos. Tempo para determinação destes pontos:

1508.530s.

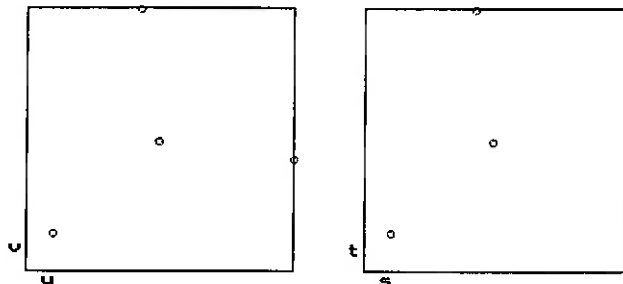
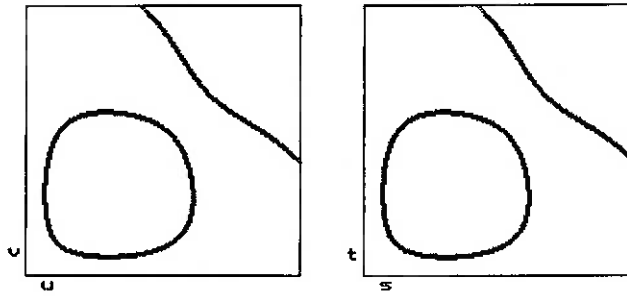
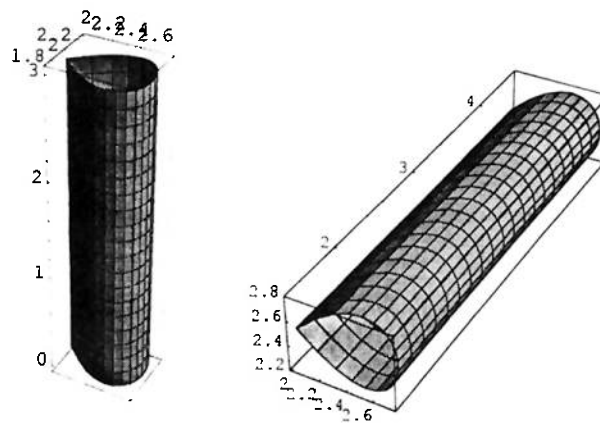


Figura 3.26: Pontos iniciais relativos à Figura 3.24 obtidos através do Método da Análise dos Domínios Paramétricos. Tempo para determinação destes pontos: 51.360s.

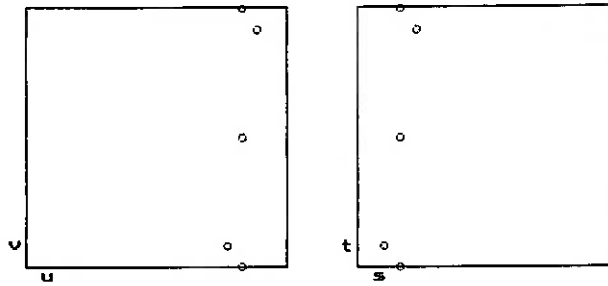


*Figura 3.27: Curvas de intersecção entre as superfícies da Figura 3.24, traçadas através do Método da Caminhada com equações diferenciais.*

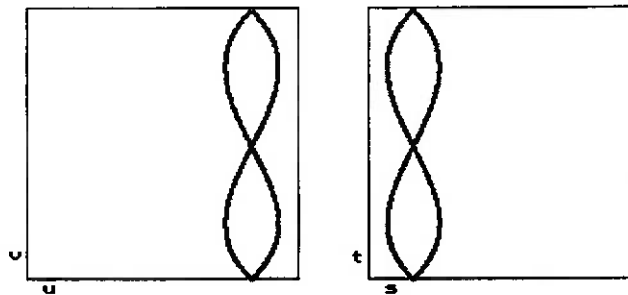
O terceiro exemplo envolve as superfícies de Bézier apresentadas na Figura 3.28. A Figura 3.29 mostra os pontos iniciais obtidos através dos Métodos da Análise dos Domínio Paramétrico (o Método da Subdivisão apresenta um problema no ponto central, relativo à singularidade causada pelo cruzamento entre as curvas de intersecção: se a precisão utilizada não for bastante alta, são encontrados pontos redundantes em grande quantidade, por outro lado, ao aumentar-se a precisão para valores muito altos, o tempo de processamento fica impraticável). Na Figura 3.30 pode-se ver o traçado das curvas de intersecção obtidas (convém notar que o algoritmo lidou adequadamente com a singularidade, através da metodologia apresentada na seção 3.5.1).



*Figura 3.28: Superfícies de Bézier cuja intersecção se deseja determinar.*



*Figura 3.29: Pontos iniciais relativos à Figura 3.28 obtidos através do Método da Análise dos Domínios Paramétricos. Tempo para determinação destes pontos: 4.230s.*



*Figura 3.30: Curvas de intersecção entre as superfícies da Figura 3.28, traçadas através do Método da Caminhada com equações diferenciais.*

## 4) B-Splines e NURBS

Antes de demonstrar como a proposta apresentada nos capítulos anteriores pode ser estendida para ser utilizada com superfícies NURBS, é necessário fazer uma pequena introdução sobre a natureza destas superfícies paramétricas. Contudo, deve-se, inicialmente, entender o conceito de curvas e superfícies B-Spline.

### 4.1) Curvas e Superfícies B-Spline

Uma curva B-Spline é uma curva paramétrica definida por [Piegel97]:

$$C(u) = \sum_{i=0}^n N_{i,p}(u) \cdot P_i \quad (103)$$

onde  $u \in \mathfrak{R}$  e  $a \leq u \leq b$ ,  $P_i$  são pontos de controle da curva no espaço, e  $N_{i,p}(u)$  são funções da família das chamadas *funções base*, de grau  $p$ .

As *funções base* são funções geradas a partir de um vetor de nós reais do seguinte tipo:

$$U = \{u_0, u_1, \dots, u_m\} \quad (104)$$

com  $u_i \leq u_{i+1}$  e  $m = n + p + 1$ .

Mais ainda, o grau  $p$  das *funções base* também é dado pelo seu vetor de nós, da seguinte maneira:

$$U = \left\{ \underbrace{a, a, \dots, a}_{p+1}, u_{p+1}, u_{p+2}, \dots, u_{m-p-2}, u_{m-p-1}, \underbrace{b, \dots, b}_{p+1} \right\} \quad (105)$$

Assim, as *funções base* de um vetor de nós são definidas por:

$$N_{i,0}(u) = \begin{cases} 1 & \text{se } u_i \leq u < u_{i+1} \\ 0 & \text{se fora da faixa acima} \end{cases} \quad (106)$$

para os valores iniciais, sendo os valores de grau mais elevado obtidos recursivamente, através de:

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (107)$$

Tomemos como exemplo o seguinte vetor de nós:  $U = \{0,0,0,1,1,1\}$ .

Construindo as *funções base* de grau 2, teremos:

$$\begin{array}{lll} N_{0,0}(u) = 0 & N_{0,1}(u) = 0 & N_{0,2}(u) = u^2 - 2u + 1 \\ N_{1,0}(u) = 0 & N_{1,1}(u) = 1 - u & N_{1,2}(u) = 2u - 2u^2 \\ N_{2,0}(u) = 1 & N_{2,1}(u) = u & N_{2,2}(u) = u^2 \\ N_{3,0}(u) = 0 & N_{3,1}(u) = 0 & \\ N_{4,0}(u) = 0 & & \end{array}$$

É interessante notar que estas  $N_{i,2}(u)$  são exatamente iguais aos valores do polinômio de Bernstein de grau 2,  $B_i^2(u)$  (vide ( 8 )). Na verdade, isto não ocorre por acaso. Curvas e superfícies de Bézier são casos especiais de curvas e superfícies B-Spline.

Vejamos outro exemplo, gerado a partir do vetor de nós  $U = \{0,0,0,0.4,1,1,1\}$ .

As funções geradas serão:

$$\begin{array}{ll} N_{0,0}(u) = 0 & N_{0,1}(u) = 0 \\ N_{1,0}(u) = 0 & N_{1,1}(u) = \begin{cases} \frac{2-5u}{2} & \text{se } 0 \leq u < 0.4 \\ 0 & \text{se } 0.4 \leq u < 1 \end{cases} \\ N_{2,0}(u) = \begin{cases} 1 & \text{se } 0 \leq u < 0.4 \\ 0 & \text{se } 0.4 \leq u < 1 \end{cases} & N_{2,1}(u) = \begin{cases} \frac{5u}{2} & \text{se } 0 \leq u < 0.4 \\ \frac{5-5u}{3} & \text{se } 0.4 \leq u < 1 \end{cases} \end{array}$$

$$N_{3,0}(u) = \begin{cases} 0 & \text{se } 0 \leq u < 0.4 \\ 1 & \text{se } 0.4 \leq u < 1 \end{cases}$$

$$N_{3,1}(u) = \begin{cases} 0 & \text{se } 0 \leq u < 0.4 \\ \frac{5u-2}{3} & \text{se } 0.4 \leq u < 1 \end{cases}$$

$$N_{4,0}(u) = 0$$

$$N_{4,1}(u) = 0$$

$$N_{5,0}(u) = 0$$

$$N_{0,2}(u) = \begin{cases} \frac{4-20u+25u^2}{4} & \text{se } 0 \leq u < 0.4 \\ 0 & \text{se } 0.4 \leq u < 1 \end{cases}$$

$$N_{1,2}(u) = \begin{cases} \frac{20u-35u^2}{4} & \text{se } 0 \leq u < 0.4 \\ \frac{5-10u+5u^2}{3} & \text{se } 0.4 \leq u < 1 \end{cases}$$

$$N_{2,2}(u) = \begin{cases} \frac{5u^2}{2} & \text{se } 0 \leq u < 0.4 \\ \frac{50u-40u^2-10}{9} & \text{se } 0.4 \leq u < 1 \end{cases}$$

$$N_{3,2}(u) = \begin{cases} 0 & \text{se } 0 \leq u < 0.4 \\ \frac{25u^2-20u+4}{9} & \text{se } 0.4 \leq u < 1 \end{cases}$$

Fica claro, então, que a curva B-Spline definida por

$$C(u) = \sum_{i=0}^3 N_{i,2}(u) \cdot P_i \quad (108)$$

terá duas regiões bem definidas em seu domínio paramétrico, conforme pode ser visto na Figura 4.1.

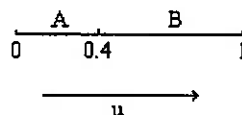


Figura 4.1: Regiões do domínio paramétrico de uma curva B-Spline.

Ou seja, na região A da Figura 4.1 valem os valores de  $N_{i,2}(u)$  cuja condição é  $u$  entre 0 e 0.4, de modo que fica descrito um pedaço da curva definida pelos polinômios resultantes do somatório ( 108 ) (um polinômio para cada coordenada do espaço):  $C^A(u) = [f_x^A(u), f_y^A(u), f_z^A(u)]$ . Analogamente, no outro pedaço da curva, B, onde  $u$  está entre 0.4 e 1, valerão os demais valores para cada  $N_{i,2}(u)$ , sendo a curva definida pelos seguintes polinômios (um para cada coordenada):  $C^B(u) = [f_x^B(u), f_y^B(u), f_z^B(u)]$ .

De fato, haverá, na curva, tantos conjuntos de 3 polinômios (um relativo a cada coordenada do espaço) quanto for o número de espaços entre nós adjacentes e distintos do vetor de nós gerador das *funções base* da curva.

Já as superfícies B-Spline podem ser definidas por:

$$S(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) \cdot N_{j,q}(v) \cdot P_{ij} \quad (109)$$

com  $u$  e  $v$  reais e  $0 \leq u, v \leq 1$ , sendo

$$U = \left\{ \underbrace{0, 0, \dots, 0}_{p+1}, u_{p+1}, \dots, u_{r-p-1}, \underbrace{1, \dots, 1}_{p+1} \right\} \quad (110)$$

o vetor de nós para as *funções base* da variável  $u$ , com  $r = n + p + 1$ , e

$$V = \left\{ \underbrace{0, 0, \dots, 0}_{q+1}, v_{q+1}, \dots, v_{s-q-1}, \underbrace{1, \dots, 1}_{q+1} \right\} \quad (111)$$

o vetor de nós para as *funções base* da variável  $v$ , com  $s = m + q + 1$ .

Vamos então imaginar uma superfície B-Spline cujas *funções base* são geradas, tanto para a variável  $u$  quanto para a  $v$ , pelo vetor de nós  $U = \{0, 0, 0, 0.4, 1, 1, 1\}$ . Neste caso, extrapolando o que foi concluído no exemplo da curva gerada anteriormente a partir deste mesmo vetor de nós, teremos uma superfície cujo domínio paramétrico pode ser dividido em regiões, conforme mostrado na Figura 4.2.



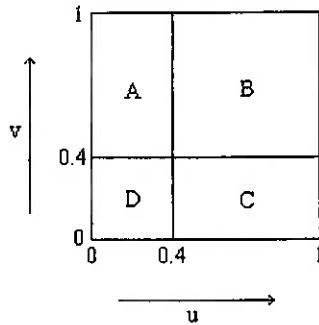


Figura 4.2: Regiões do domínio paramétrico de uma superfície B-Spline.

Cada uma dessas regiões, chamadas retalhos (patches), é descrita por um conjunto de 3 polinômios, um para cada coordenada do espaço, da forma  $S^{região}(u, v) = [f_x^{região}(u, v), f_y^{região}(u, v), f_z^{região}(u, v)]$ . Todavia, podemos transformar cada polinômio  $f_i$  de uma dada região para a base de Bernstein (66), através do procedimento descrito a partir de (47) na seção 3.3. Com isto, obtém-se uma matriz de pontos de controle  $W_{((grau\ em\ u)+1) \times ((grau\ em\ v)+1)}$  para cada  $f_i$ , cada matriz relativa a uma coordenada do espaço. Então, estas 3 matrizes podem ser rearranjadas de forma a constituírem a matriz  $P_{ij}$  de pontos de controle de um retalho, equivalente a uma superfície de Bézier descrita em (18). Portanto, é possível descrever cada região da superfície B-Spline como um retalho. No exemplo em questão, a superfície B-Spline dará origem a 4 retalhos. Num caso geral, uma superfície B-Spline dará origem a  $\varphi$  retalhos, sendo  $\varphi$  o resultado do produto entre o número de espaços entre nós adjacentes e distintos do vetor de nós gerador de *funções base* para variável  $u$  e o número de espaços entre nós adjacentes e distintos do vetor de nós gerador de *funções base* para variável  $v$ .

Desta maneira, determinar a intersecção entre duas superfícies B-Spline, sendo que uma dá origem a  $\varphi$  retalhos e a outra gera  $\eta$  retalhos, significa aplicar  $\varphi \times \eta$  vezes o procedimento de determinação de intersecção entre superfícies de Bézier descrito nos capítulos anteriores. Na verdade, este número pode ser sensivelmente menor, bastando para tal verificar, antes de executar o procedimento de determinação da intersecção em si, se o convex-hull formado pelos pontos de controle de um retalho intercepta o convex-hull dos pontos de controle do outro retalho. Se isto não ocorrer,

sabe-se, de antemão, que este par de superfícies não se cruza. Como tal verificação envolve somente a determinação da existência de intersecção entre dois poliedros, ela pode significar uma vantajosa maneira de poupar algum tempo de execução do procedimento global.

## 4.2) Superfícies NURBS

Agora, vamos avançar em direção do tratamento de superfícies NURBS (Non-Uniform Rational B-Spline). Uma superfície deste tipo é uma forma ainda mais geral de superfície paramétrica, quando comparada com uma B-Spline ordinária. A superfície NURBS é definida a seguir [Piegel97]:

$$S(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) \cdot N_{j,q}(v) \cdot w_{ij} \cdot \mathbf{P}_{ij}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) \cdot N_{j,q}(v) \cdot w_{ij}} \quad (112)$$

cujos termos são os mesmos enumerados após ( 109 ), na definição de superfícies B-Spline, acrescidos dos pesos  $w_{ij}$ , que formam a matriz de pesos na qual cada elemento  $ij$  irá definir o peso do ponto de controle  $\mathbf{P}_{ij}$  na superfície.

Da mesma forma que foi mostrado para o caso das B-Splines, a superfície NURBS pode ter seu domínio paramétrico subdividido em regiões (retalhos) dentro das quais vale um determinado conjunto de polinômios, um para cada coordenada do espaço. A diferença para o caso anterior está no fato destes polinômios serem racionais, de forma que uma região da superfície é descrita por:

$$S^{região}(u, v) = \left[ \frac{f_x^{região}(u, v)}{q^{região}(u, v)}, \frac{f_y^{região}(u, v)}{q^{região}(u, v)}, \frac{f_z^{região}(u, v)}{q^{região}(u, v)} \right] \quad (113)$$

Com isto, uma forma de utilizar a proposta feita neste trabalho para superfícies NURBS é proceder da mesma forma descrita anteriormente em relação a B-Splines. Para fazê-lo, contudo, é necessário deixar as equações na forma racional da expressão ( 113 ). Por exemplo, dadas duas superfícies NURBS  $F(u, v)$  e  $G(s, t)$ , deseja-se

determinar a intersecção entre elas. Identifica-se, então, os retalhos nos quais elas se subdividem, procedendo-se, daí, com o problema de determinar a intersecção de cada retalho de uma superfície com todos os retalhos da outra.

Tomemos o caso do retalho  $i$  da superfície  $F$ , definida por  $\mathbf{F}^i(u, v) = \left[ \frac{f_x^i(u, v)}{q_F^i(u, v)}, \frac{f_y^i(u, v)}{q_F^i(u, v)}, \frac{f_z^i(u, v)}{q_F^i(u, v)} \right]$ , e do retalho  $j$  da superfície  $G$ , definida por  $\mathbf{G}^j(s, t) = \left[ \frac{g_x^j(s, t)}{q_G^j(s, t)}, \frac{g_y^j(s, t)}{q_G^j(s, t)}, \frac{g_z^j(s, t)}{q_G^j(s, t)} \right]$ . Daí, substituindo estes valores em ( 33 ), que designa o problema da intersecção, então, para a coordenada  $\alpha$ , é obtida a equação:

$$\frac{f_\alpha^i(u, v)}{q_F^i(u, v)} - \frac{g_\alpha^j(s, t)}{q_G^j(s, t)} = 0 \quad (114)$$

que, devidamente rearranjada, dá origem a:

$$\underbrace{\frac{1}{q_F^i(u, v) \cdot q_G^j(s, t)}}_{\text{termo1}} \underbrace{(f_\alpha^i(u, v) \cdot q_G^j(s, t) - g_\alpha^j(s, t) \cdot q_F^i(u, v))}_{\text{termo2}} = 0 \quad (115)$$

Assim, a equação relativa a cada coordenada do espaço fica reduzida à parcela *termo2*. Logo, conforme explicado na seção 3.1 e na 3.2, pode-se substituir as variáveis  $u$ ,  $v$ ,  $s$  e  $t$  pelos valores das bordas dos domínios paramétricos das regiões em questão em ( 33 ), determinando-se, ao resolver o sistema 3x3 restante, os pontos iniciais onde as curvas de intersecção cruzam cada borda. A parcela *termo1*, igual para as 3 coordenadas do espaço, fica sendo apenas uma restrição, devendo ser utilizada para verificar se algumas das soluções encontradas para o sistema 3x3 não são, na verdade, singularidades matemáticas (ou seja, se não geram um zero no denominador de *termo1*). É interessante notar que o processo de solução dos sistemas 3x3, feito pelo método do poliedro projetado, exigirá, para esses retalhos de superfícies NURBS, o mesmo esforço computacional envolvido quando os sistemas eram originados por superfícies de Bézier, posto que o grau máximo das variáveis no *termo2* de ( 115 ) é o grau das superfícies em questão, sejam elas NURBS ou Bézier. Por exemplo, se estivermos lidando com NURBS

de 3º grau, então o grau máximo das variáveis em ( 33 ) será 3, o mesmo valor gerado em ( 33 ) por superfícies de Bézier de 3º grau.

Um processo análogo dá-se na detecção de laços de intersecção entre estes retalhos de superfícies NURBS, embora certamente haja um considerável acréscimo no tempo de processamento se comparado com o necessário para superfícies de Bézier, uma vez que as derivadas necessárias para montar o sistema 4x4 de ( 45 ) (Método da Análise) geram um aumento bastante significativo no grau das equações envolvidas.

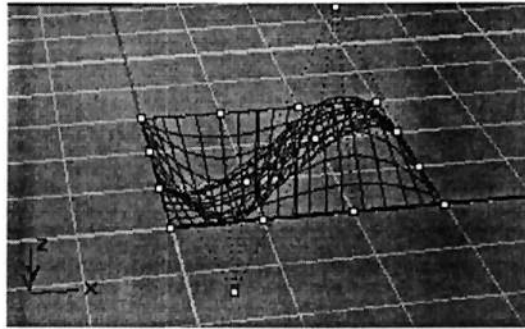
#### **4.3) Exemplos Resolvidos com o Sistema Implementado**

Foi implementado um sistema, em linguagem C (compilador Visual C++), para determinar as curvas de intersecção entre superfícies NURBS. Este sistema lê os parâmetros das superfícies NURBS de um arquivo, gerando as equações polinomiais relativas aos retalhos de cada superfícies. Então, realiza-se a determinação dos pontos iniciais entre cada retalho de uma superfície em relação a cada retalho da outra através do Método da Análise, visto na seção 3.2, cujos sistemas de equações são resolvidos com as propostas das seções 3.3 e 3.4. Tendo-se o conjunto total deste pontos iniciais obtidos destes cruzamentos, utiliza-se o Método da Caminhada (seção 3.5) para traçar todas as curvas de intersecção.

Para esclarecer estes procedimentos, são apresentados alguns exemplos resolvidos com este sistema. Nestes, foi definida uma *tolerância* de 0.0001 para o Método do Poliedro Projetado (vide seção 3.4). Os testes foram feitos num Pentium 200 MHz.

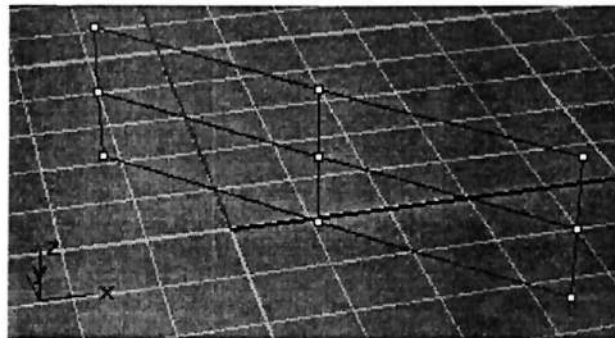
O primeiro exemplo consiste da determinação da intersecção entre as superfícies NURBS das Figuras 4.3 e 4.4. A superfície da Figura 4.3, denominada **F**, é gerada a partir dos pontos de controle nela destacados e dos vetores de nós  $U = \{0,0,0,0.4,1,1,1\}$  e  $V = \{0,0,0,0.2,2,2,2\}$ . Assim, esta superfície dá origem a dois retalhos

(cada qual dado por uma expressão do tipo da ( 113 )): um definido para  $0 \leq u \leq 0.4$  e  $0 \leq v \leq 2$ , e o outro para  $0.4 \leq u \leq 1$  e  $0 \leq v \leq 2$ .



*Figura 4.3: Superfície NURBS, com os respectivos pontos de controle destacados.*

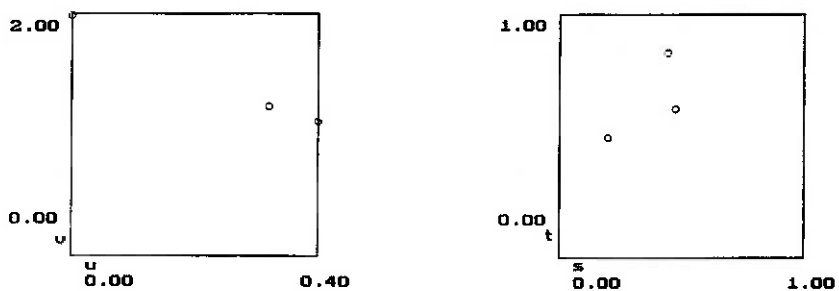
Já a superfície da Figura 4.4, denominada **G**, é obtida a partir dos vetores de nós  $U = V = \{0,0,0,1,1,1\}$ . Com isto, tal superfície é dada por apenas um retalho.



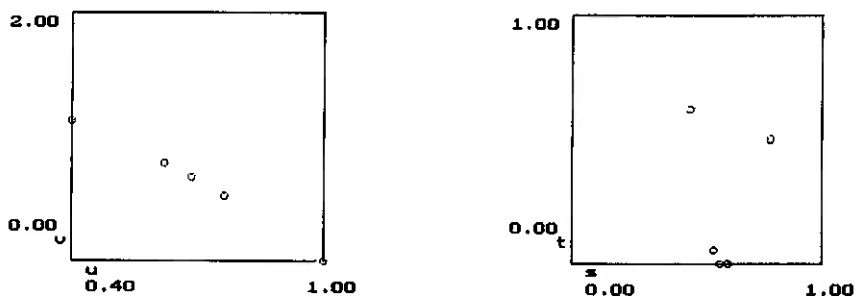
*Figura 4.4: Superfície NURBS, com os respectivos pontos de controle destacados.*

Utilizando o Método da Análise para determinar os pontos iniciais da interseção entre o primeiro retalho da superfície **F** com a superfície **G** encontram-se os pontos destacados na Figura 4.5.

Já na Figura 4.6, são exibidos os pontos iniciais obtidos com a aplicação do Método da Análise para o segundo retalho da superfície **F** com a superfície **G**.

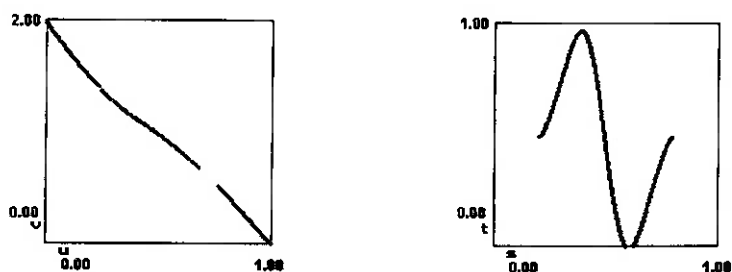


*Figura 4.5: Pontos iniciais para o primeiro retalho da superfície F com a superfície G, vistos dos respectivos domínios paramétricos. Tempo de processamento: 49,5s.*

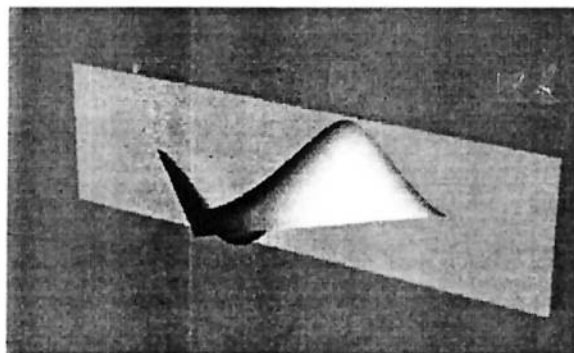


*Figura 4.6: Pontos iniciais para o segundo retalho da superfície F com a superfície G, vistos dos respectivos domínios paramétricos. Tempo de Processamento: 58,6s.*

Na Figura 4.7, estão finalmente apresentados os traçados das curvas de interseção obtidos a partir do conjunto total dos pontos iniciais através do Método da Caminhada. Esta interseção pode ser melhor visualizada na imagem renderizada das superfícies F e G apresentada na Figura 4.8.

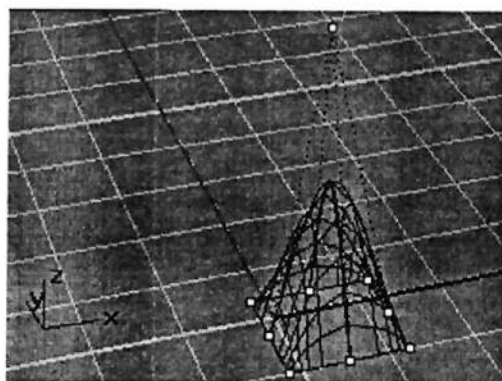


*Figura 4.7: Curvas de intersecção traçadas através do Método da Caminhada e vistas a partir dos respectivos domínios paramétricos.*



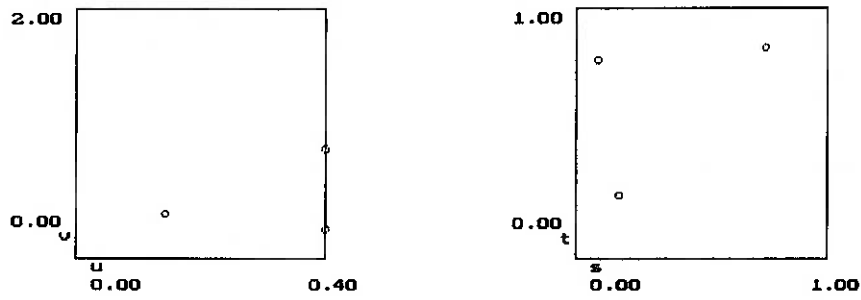
*Figura 4.8: Imagem renderizada das superfícies **F** e **G**.*

A superfície da Figura 4.9, denominada **H**, também é dada pelos vetores de nós  $\mathbf{U} = \mathbf{V} = \{0,0,0,1,1,1\}$ , sendo constituída, portanto, por apenas um retalho. Assim, a Figura 4.10 mostra os pontos iniciais da intersecção entre o primeiro retalho de **F** com **H**, e a Figura 4.11 os pontos da intersecção do segundo retalho de **F** com **H**.

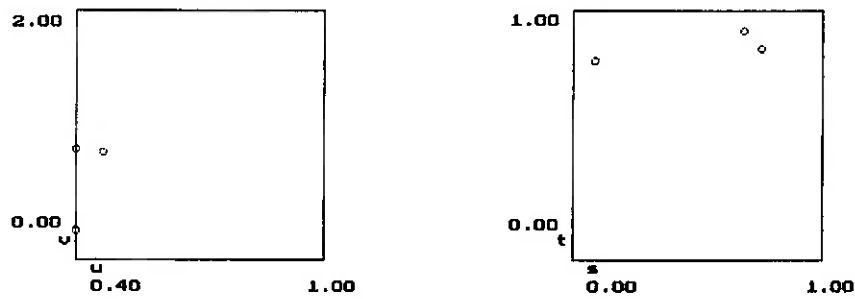


*Figura 4.9: Superfície NURBS, com os respectivos pontos de controle destacados.*

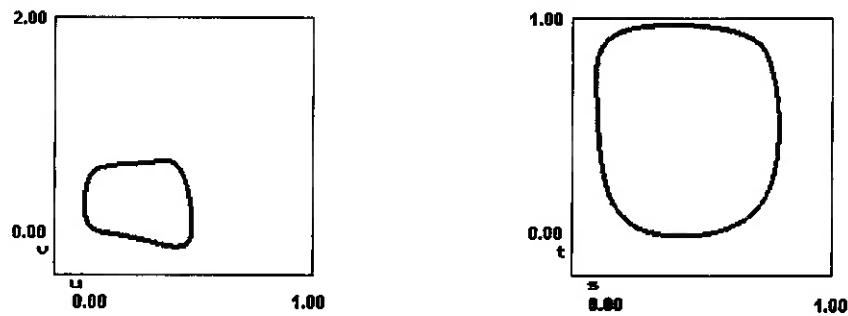
Finalmente, na Figura 4.12 temos as curvas de intersecção entre **F** e **H** traçadas com o Método da Caminhada a partir dos pontos iniciais. Uma melhor visualização destas superfícies é dada pela imagem renderizada da Figura 4.13.



*Figura 4.10: Pontos iniciais para o primeiro retalho da superfície  $F$  com a superfície  $H$ , vistos dos respectivos domínios paramétricos. Tempo de processamento: 1658s.*

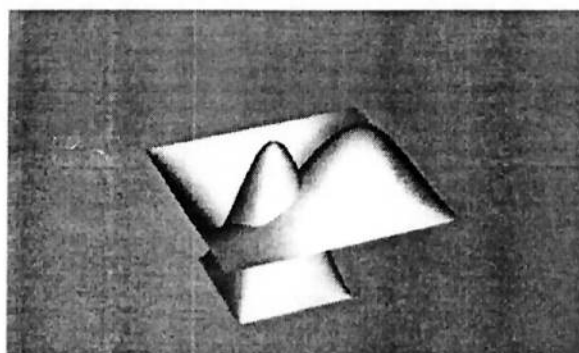


*Figura 4.11: Pontos iniciais para o segundo retalho da superfície  $F$  com a superfície  $H$ , vistos dos respectivos domínios paramétricos. Tempo de Processamento: 2339s.*



*Figura 4.12: Curvas de intersecção traçadas através do Método da Caminhada e vistas a partir dos respectivos domínios paramétricos.*





*Figura 4.13: Imagem renderizada das superfícies  $F$  e  $H$ .*

## 5) Conclusões

Nesta dissertação foram apresentados métodos estudados e implementados para determinar a intersecção entre superfícies paramétricas. Inicialmente, foi implementada uma abordagem com superfícies de Bézier para decidir qual método seria mais eficiente na determinação dos pontos iniciais das curvas de intersecção. Por apresentar um tempo de processamento bastante menor e uma saída mais adequada, o Método da Análise dos Domínios Paramétricos foi preferido para tal tarefa. A este foi acoplado o Método da Caminhada com equações diferenciais para proceder com o traçado das curvas de intersecção, gerando um algoritmo híbrido para a determinação da intersecção entre superfícies. Tal algoritmo foi estendido para manipular superfícies NURBS, o que foi conseguido com sucesso, atingindo-se, assim, o objetivo inicial deste trabalho de mestrado.

## 6) Referências Bibliográficas

- Bartels87 Bartels, R.H.; Beatty, J.C.; Barsky, B.A.; *An Introduction to Splines for Use in Computer Graphics & Geometric Modelling*. Morgan Kaufmann Publishes, Inc. 1987.
- Carnahan69 Carnahan, B. e Luther, H. A.; *Applied Numerical Methods*. John Wiley, New York, 1969.
- Farin88 Farin, G.L.; *Curves and Surfaces for Computer Aided Geometric Design: Practical Guide*. Academic Press, San Diego, 1988.
- Fiorot92 Fiorot, J.C. e Jeannin, P.; *Rational Curves and Surfaces Applications to CAD*, John Wiley & Sons, 1992.
- Grandine97 Grandine, T. A. e Klein IV, F. W.; *A new approach to the surface intersection problem*. Computer Aided Geometric Design, No. 14, 1997, pp. 111-134.
- Hosaka92 Hosaka, M.; *Modeling of Curves and Surfaces in CAD/CAM*. Springer-Verlag, Berlim, 1992.
- Hoschek93 Hoschek, Josef e Lasser, Dieter; *Fundamentals of Computer Aided Geometric Design*. A. K. Peters, 1993.
- Hu97 Hu, C. Y., Maekawa, T., Patrikalakis, N. M. e Ye, X.; *Robust interval algorithm for surface intersections*. Computer-Aided Design, Vol. 29, No. 9, 1997, pp.617-627.
- Kriezis92 Kriezis, G. A., Patrikalakis, N. M. e Wolter, F. E.; *Topological and differential-equation methods for surface intersections*. Computer-Aided Design, Vol. 24, No. 1, 1992, pp.41-55.
- Lasser86 Lasser, D.; *Interserction of parametric surfaces in the Bernstein-Bézier representation*. Computer-Aided Design, Vol. 18, No. 4, 1986, pp. 186-192.
- Makita96 Makita, R.K., Tsuzuki, M.S.G.; *Automatic Generation of Toolpath for Sculptured Surface Machining*. SAE BRASIL, 1996.

- Piegel97      Piegel, L. e Tiller, W.; *The NURBS Book*, Springer-Verlag, Second Edition, 1997.
- Pincinato96      Pincinato, C.P. e Tsuzuki, M.S.G.; *Algorithm for Determining the Intersection Curve Between Bezier Surfaces*. SAE BRASIL, 1996.
- Sederberg89      Sederberg, T. W., Christiansen, H. N. e Katz, S. ; *An improved test for closed loops in surface intersections*. Computer-Aided Design, Vol. 21, No. 8, 1989, pp.505-508.
- Sherbrooke93      Sherbrooke, E. C. e Patrikalakis N. M.; *Computation of the solutions of nonlinear polynomial systems*. Computer Aided Geometric Design No. 10, 1993, pp. 379-405.
- Ueda96      Ueda, R.Y.Y. e Tsuzuki, M.S.G.; *Representing Curves and Surfaces in B-Rep Solid Modelers*. SAE BRASIL, 1996.
- Wang91      Wang, Y., Gursoz, E. L., Chen, J. -M., Prinz, F. B. e Patrikalakis, N. M.; *Intersection of Parametric Surfaces For Next Generation Geometric Modelers in Product Modeling for Computer-Aided Design and Manufacturing*, Elsevier Science Publishers B. V., New York, 1991, pp. 75-96.
- Yamaguchi88      Yamaguchi, F.; *Curves and Surfaces in Computer Aided Geometric Design*. Springer-Verlag, Berlim, 1988.
- Zeid91      Zeid, I.; *CAD/CAM Theory and Practice*. McGrawHill International Editions.1991.
- Zhou93      Zhou, J., Sherbrooke, E. C. e Patrikalakis N. M.; *Computation of Stationary Points of Distance Functions*. Engineering with Computers No. 9, 1993, pp. 231-246.

## Apêndice 1: Demonstração de algumas propriedades dos Polinômios de Bernstein

Vamos provar algumas expressões utilizadas no Método do Poliedro Projetado para solução de sistemas de polinômios (seção 3.4.1).

### A1.1

Começemos pela expressão ( 71 ). Reescrevendo-a:

$$\sum_{i=0}^n \frac{i}{n} B_i^n(u) = \sum_{i=0}^n \frac{i}{n} \frac{n!}{(n-i)!i!} (1-u)^{n-i} u^i = u \sum_{i=0}^n \frac{(n-1)!}{(n-i)!(i-1)!} (1-u)^{n-i} u^{i-1} \quad (116)$$

Fazendo uma transformação de variáveis do tipo  $j = i-1$  e  $m = n-1$ , obtemos:

$$\sum_{i=0}^n \frac{i}{n} B_i^n(u) = u \underbrace{\sum_{j=0}^m \frac{m!}{(m-j)!j!} (1-u)^{m-j} u^j}_{1} = u \quad (117)$$

c. q. d.

### A1.2

Agora, tomemos a expressão ( 77 ). Utilizando a definição de multinômio de Bernstein em ( 67 ), vamos reescrever:

$$\sum_I B_I^D(\mathbf{u}) = \sum_{i_1=0}^{d_1} \sum_{i_2=0}^{d_2} \cdots \sum_{i_n=0}^{d_n} B_{i_1}^{d_1}(u_1) B_{i_2}^{d_2}(u_2) \cdots B_{i_n}^{d_n}(u_n) \quad (118)$$

que, através da propriedade distributiva, pode ser rearranjada para:

$$\sum_I B_I^D(\mathbf{u}) = \left( \sum_{i_1=0}^{d_1} B_{i_1}^{d_1}(u_1) \right) \left( \sum_{i_2=0}^{d_2} B_{i_2}^{d_2}(u_2) \right) \cdots \left( \sum_{i_n=0}^{d_n} B_{i_n}^{d_n}(u_n) \right) = \prod_{j=1}^n \left( \sum_{i_j=0}^{d_j} B_{i_j}^{d_j}(u_j) \right) \quad (119)$$

Contudo, através de ( 11 ), sabe-se que cada termo  $\sum_{i_j=0}^{d_j} B_{i_j}^{d_j}(u_j)$  de ( 119 ) é

igual a 1. C. q. d.

Resta provar ( 72 ). Reescrevendo-a e usando ( 71 ) e ( 119 ):

$$\sum_I \frac{i_j}{d_j^{(k)}} B_I^{D^{(k)}}(x) = \underbrace{\left( \sum_{i_1=0}^{d_1^{(k)}} B_{i_1}^{d_1^{(k)}}(x_1) \right)}_1 \underbrace{\left( \sum_{i_2=0}^{d_2^{(k)}} B_{i_2}^{d_2^{(k)}}(x_2) \right)}_1 \cdots \underbrace{\left( \sum_{i_j=0}^{d_j^{(k)}} \frac{i_j}{d_j^{(k)}} B_{i_j}^{d_j^{(k)}}(x_j) \right)}_{x_j} \cdots \underbrace{\left( \sum_{i_n=0}^{d_n^{(k)}} B_{i_n}^{d_n^{(k)}}(x_n) \right)}_1 = x_j \quad (120)$$

c. q. d.

## **Apêndice 2: Demonstração de algumas das equações descritas no Método da Caminhada**

Faz-se necessário provar algumas das fórmulas utilizadas na seção 3.5. A começar das expressões de ( 85 ).

### **A2.1**

Da primeira equação de ( 84 ), sabe-se que  $dC/dw$  é uma combinação linear de  $F_u$  e  $F_v$ . Portanto, os vetores  $dC/dw$ ,  $F_u$  e  $F_v$  são coplanares. Mais ainda, de ( 34 ) sabe-se que  $N^F$  é normal a  $F_u$  e a  $F_v$ , e, por conseguinte, a  $dC/dw$ . Logo, é razoável admitir a validade da seguinte expressão:

$$\lambda N^F = F_v \times dC/dw \quad (121)$$

com  $\lambda \in \mathfrak{R}$ . Substituindo  $dC/dw$  pela primeira equação de ( 84 ) em ( 121 ), teremos:

$$\lambda N^F = F_v \times (du/dw F_u + dv/dw F_v) = -du/dw F_u \times F_v \quad (122)$$

Substituindo ( 34 ) em ( 122 ) chegamos à expressão:

$$\lambda = -du/dw |F_u \times F_v| \quad (123)$$

que, em ( 121 ), origina:

$$\frac{du}{dw} = - \frac{(F_v \times dC/dw) \circ N^F}{|F_u \times F_v|} \quad (124)$$

Esta última expressão equivale ao produto misto da primeira equação de ( 85 ). Utilizando um raciocínio análogo, pode-se obter as expressões das demais variáveis em ( 85 ).

### **A2.2**

Vamos agora demonstrar as expressões para as derivadas de  $\phi$ . Diferenciando-se diretamente a equação ( 86 ) em relação à variável  $u$ , chega-se a

$$\phi_u = \mathbf{N}^G \circ (\mathbf{F}_u - \mathbf{G}_u) + \mathbf{N}_u^G \circ (\mathbf{F} - \mathbf{G}) \quad (125)$$

Pela definição da equação ( 35 ), sabe-se que  $\mathbf{N}^G \circ \mathbf{G}_s = 0$  e que  $\mathbf{N}^G \circ \mathbf{G}_t = 0$ . Portanto,  $\mathbf{N}^G \circ \mathbf{G}_u = 0$  (posto que  $s$  e  $t$  são os parâmetros linearmente independentes que definem  $\mathbf{G}$ , logo qualquer outra variável de  $\mathbf{G}$  será combinação linear destas duas). Mais ainda, sendo  $\mathbf{N}^G$  um vetor unitário por definição, sabe-se que  $\mathbf{N}^G \circ \mathbf{N}^G = 1$ . Derivando-se esta expressão em relação a um parâmetro  $\tau$  e utilizando a regra da cadeia, obtém-se

$$\frac{\partial \mathbf{N}^G}{\partial \tau} \circ \mathbf{N}^G + \mathbf{N}^G \circ \frac{\partial \mathbf{N}^G}{\partial \tau} = 0$$

ou seja,

$$\mathbf{N}^G \circ \mathbf{N}_\tau^G = 0. \quad (126)$$

Todavia, sabe-se que

$$(\mathbf{F} - \mathbf{G}) = \alpha \cdot \mathbf{N}^G, \quad \alpha \in \mathfrak{R} \quad (127)$$

pois  $\phi$  é uma projeção ortogonal de um ponto em  $\mathbf{F}$  sobre  $\mathbf{G}$ . Portanto,  $\mathbf{N}_u^G \circ (\mathbf{F} - \mathbf{G}) = 0$ .

Com isto,

$$\phi_u = \mathbf{N}^G \circ \mathbf{F}_u \quad (128)$$

Analogamente, encontra-se a derivada para a variável  $v$

$$\phi_v = \mathbf{N}^G \circ \mathbf{F}_v \quad (129)$$

As derivadas de segunda ordem de  $\phi$  podem ser encontradas operando sobre ( 128 ) e ( 129 ) diretamente, obtendo ( 89 ), ( 90 ) e ( 91 ).

### A2.3

Pelas expressões ( 35 ) e ( 127 ), deduz-se que:

$$(\mathbf{F} - \mathbf{G}) \circ \mathbf{G}_s = 0 \quad (130)$$

$$(\mathbf{F} - \mathbf{G}) \circ \mathbf{G}_t = 0 \quad (131)$$

Diferenciando ( 130 ) e ( 131 ) em relação ao parâmetro  $w$  da curva  $C(w)$ :

$$\left( \mathbf{F}_u \frac{du}{dw} + \mathbf{F}_v \frac{dv}{dw} - \mathbf{G}_s \frac{ds}{dw} - \mathbf{G}_t \frac{dt}{dw} \right) \circ \mathbf{G}_s + (\mathbf{F} - \mathbf{G}) \circ \left( \mathbf{G}_{ss} \frac{ds}{dw} + \mathbf{G}_{st} \frac{dt}{dw} \right) = 0 \quad (132)$$

$$\left( \mathbf{F}_u \frac{du}{dw} + \mathbf{F}_v \frac{dv}{dw} - \mathbf{G}_s \frac{ds}{dw} - \mathbf{G}_t \frac{dt}{dw} \right) \circ \mathbf{G}_t + (\mathbf{F} - \mathbf{G}) \circ \left( \mathbf{G}_{st} \frac{ds}{dw} + \mathbf{G}_{tt} \frac{dt}{dw} \right) = 0 \quad (133)$$

que, manipulando-se para a forma matricial, origina a expressão ( 98 ).

#### A2.4

Para as expressões das derivadas parciais trocadas ( 92 ) e ( 93 ), utiliza-se as seguintes equações conseguidas por simples diferenciação:

$$\mathbf{N}_u^G = \mathbf{N}_s^G \frac{ds}{du} + \mathbf{N}_t^G \frac{dt}{du} \quad (134)$$

$$\mathbf{N}_v^G = \mathbf{N}_s^G \frac{ds}{dv} + \mathbf{N}_t^G \frac{dt}{dv} \quad (135)$$

Se a curva  $C(w)$  é isoparamétrica em  $v$  na superfície  $F$ , faz-se a substituição de  $w$  por  $u$  em ( 132 ). Já se for isoparamétrica em  $u$  na superfície  $F$ , faz-se a substituição de  $w$  por  $v$  em ( 133 ). As novas equações são então resolvidas para  $ds/du$ ,  $dt/du$ ,  $ds/dv$  e  $dt/dv$ , os quais são substituídos em ( 134 ) e ( 135 ) para encontrar as expressões ( 92 ) e ( 93 ).

À primeira vista, pode parecer intrincado efetuar a derivada do módulo  $|\mathbf{G}_s \times \mathbf{G}_t|$  tanto em ( 95 ) quanto em ( 96 )

$$\mathbf{N}_t^G = \frac{\left( \mathbf{G}_{st} \times \mathbf{G}_t + \mathbf{G}_s \times \mathbf{G}_{tt} - \frac{\partial}{\partial t} (|\mathbf{G}_s \times \mathbf{G}_t|) \cdot \mathbf{N}^G \right)}{|\mathbf{G}_s \times \mathbf{G}_t|} \quad (96)$$

Entretanto, substituindo ( 95 ) em ( 126 ) tem-se que:



$$\frac{\left( \mathbf{N}^G \circ (\mathbf{G}_{ss} \times \mathbf{G}_t + \mathbf{G}_s \times \mathbf{G}_{st}) - \frac{\partial}{\partial s} (|\mathbf{G}_s \times \mathbf{G}_t|) \cdot \left( \overbrace{\mathbf{N}^G \circ \mathbf{N}^G}^1 \right) \right)}{|\mathbf{G}_s \times \mathbf{G}_t|} = 0 \quad (136)$$

o que leva a:

$$\frac{\partial}{\partial s} (|\mathbf{G}_s \times \mathbf{G}_t|) = (\mathbf{G}_{ss} \times \mathbf{G}_t + \mathbf{G}_s \times \mathbf{G}_{st}) \circ \mathbf{N}^G \quad (137)$$

Analogamente, substituindo ( 96 ) em ( 126 ), obtém-se:

$$\frac{\partial}{\partial t} (|\mathbf{G}_s \times \mathbf{G}_t|) = (\mathbf{G}_{st} \times \mathbf{G}_t + \mathbf{G}_s \times \mathbf{G}_{tt}) \circ \mathbf{N}^G \quad (138)$$

### **Apêndice 3: O Método Runge-Kutta de 4ª ordem**

Queremos determinar a solução para o sistema de equações diferenciais ordinárias (EDO's) definido em ( 85 ). Tendo este objetivo, será introduzido, inicialmente, o método de integração numérica Runge-Kutta, sendo posteriormente mostrado como ele pode ser usado para resolver um sistema de EDO's.

Fica, desta forma, fixado o seguinte objetivo inicial:

dados

$$\frac{dy}{dx} = f(x, y) \quad (139)$$

e valores iniciais para  $x$  e  $y$ , deseja-se determinar o próximo valor de  $y$ .

O método de integração numérica Runge-Kutta (RK) [Carnahan69] possui como idéia geral que os valores de  $y$  podem ir sendo determinados somando-se consecutivamente ao valor inicial uma função incremental  $\phi$ , dependente da derivada exibida em ( 139 ), dos valores iniciais de  $x$  e  $y$  e do passo do algoritmo,  $h$ . Ou seja:

$$y_{i+1} = y_i + h \cdot \phi_i(x_i, y_i, h) \quad (140)$$

A ordem do método está relacionada ao nível de recursão da função  $f$  (de ( 139 )) na função incremental  $\phi$ .

Antes de apresentamos o método Runge-Kutta de 4ª ordem, que possui maior complexidade, discutiremos o método Runge-Kutta de 2ª ordem (RK2). Neste, a função  $\phi$  é definida por

$$\phi_i = a \cdot k_1 + b \cdot k_2 \quad (141)$$

sendo

$$k_1 = f(x_i, y_i) \quad (142)$$

$$k_2 = f(x_i + p \cdot h, y_i + q \cdot h \cdot k_1) \quad (143)$$

o que dá origem a:

$$y_{i+1} = y_i + h \cdot [a \cdot f(x_i, y_i) + b \cdot f(x_i + p \cdot h, y_i + q \cdot h \cdot f(x_i, y_i))] \quad (144)$$

Os valores de  $a$ ,  $b$ ,  $p$  e  $q$  são passíveis de escolha, o que é demonstrado da seguinte forma:

Utilizando-se a expansão de Taylor para  $f(x, y)$ , definida abaixo:

$$f(x+r, y+s) = f(x, y) + r \cdot f_x(x, y) + s \cdot f_y(x, y) + \theta[|r| + |s|] \quad (145)$$

Já a expansão de Taylor para  $y(x)$  fornece:

$$y(x_i + h) = y(x_i) + h \cdot f(x_i, y(x_i)) + \frac{h^2}{2} f'(x_i, y(x_i)) + \theta[h^3] \quad (146)$$

Nosso objetivo é que

$$y(x_i + h) = y_{i+1} \quad (147)$$

ou seja, que o valor real  $y(x_i + h)$  seja igual ao valor calculado  $y_{i+1}$ .

Assim, lembrando que  $\frac{df}{dt} = \frac{\partial f}{\partial x} \frac{dx}{dt} + \frac{\partial f}{\partial y} \frac{dy}{dt}$ , ou, no caso,  $f' = f_x + f_y \cdot f$ , e

substituindo ( 145 ) em ( 144 ), e depois igualando a expressão resultante e ( 146 ) em ( 147 ), tem-se:

$$y(x_i) + h \cdot f + \frac{h^2}{2} (f_x + f_y \cdot f) = y_i + h \cdot [a \cdot f + b \cdot (f + p \cdot h \cdot f_x + q \cdot h \cdot f \cdot f_y)] \quad (148)$$

ou seja,

$$y(x_i) + h \cdot f + \frac{h^2}{2} (f_x + f_y \cdot f) = y_i + h \cdot (a \cdot f + b \cdot f) + h^2 \cdot (bp \cdot f_x + bq \cdot f \cdot f_y) \quad (149)$$

Agora, igualando os dois lados de ( 149 ) por potências de  $h$ :

$$\begin{aligned}
 h^1 &\rightarrow f = a \cdot f + b \cdot f \\
 h^2 &\rightarrow \frac{f_x + f_y \cdot f}{2} = b \cdot p \cdot f_x + b \cdot q \cdot f_y \cdot f
 \end{aligned}
 \tag{150}$$

ou

$$\begin{aligned}
 a + b &= 1 \\
 p = q &= \frac{1}{2 \cdot b}
 \end{aligned}
 \tag{151}$$

de modo que podem ser escolhidos valores para  $a$  e  $b$  desde que os mesmos satisfaçam ao critério acima.

Analogamente ao caso anterior, o método Runge-Kutta de 4ª ordem (RK4) é definido por

$$y_{i+1} = y_i + h \cdot \phi_i$$

onde

$$\phi_i = a \cdot k_1 + b \cdot k_2 + c \cdot k_3 + d \cdot k_4 \tag{152}$$

Com isto, utilizando-se as escolhas  $a = 1/6$ ,  $b = 1/3$ ,  $c = 1/3$  e  $d = 1/6$  [Carnahan69], obtém-se:

$$\begin{aligned}
 k_1 &= f(x_i, y_i) \\
 k_2 &= f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}h \cdot k_1\right) \\
 k_3 &= f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}h \cdot k_2\right) \\
 k_4 &= f(x_i + h, y_i + h \cdot k_3)
 \end{aligned}
 \tag{153}$$

e

$$y_{i+1} = y_i + \frac{h}{6}(k_1 + 2 \cdot k_2 + 2 \cdot k_3 + k_4) \tag{154}$$

Todavia, o que queremos resolver é um sistema de equações diferenciais dado por ( 85 ). Para tal, pode ser desenvolvido um algoritmo para a solução sistemas EDO de 1ª ordem usando o método RK4.

Assim, dado o seguinte sistema:

$$\begin{aligned} \frac{dy_1}{dx} &= f_1(x, y_1, y_2, \dots, y_n) \\ \frac{dy_2}{dx} &= f_2(x, y_1, y_2, \dots, y_n) \\ &\vdots \\ \frac{dy_n}{dx} &= f_n(x, y_1, y_2, \dots, y_n) \end{aligned} \quad (155)$$

utiliza-se o algoritmo a seguir para resolvê-lo:

**Sejam**

`num_eq`: número de equações a serem resolvidas (4, no caso de ( 85 ))  
`H`: passo de integração  
`Y[1..num_eq]`: vetor com respostas desejadas, ou seja, os valores das variáveis após efetuar-se um passo da integração  
`Y0[1..num_eq]`: vetor com condições iniciais (pontos iniciais já obtidos)

**Para** `I = 1 até num_eq`  
`Y[I] = Y0[I]` /\* atribui condições iniciais \*/

**Enquanto** `Y` e faixa do espaço paramétrico das superfícies  
`RK4(num_eq, Y, H)`  
 Armazena `Y` na lista de pontos /\* ou seja, guarda na lista os pontos da curva de intersecção que está sendo percorrida \*/

**Procedimento** `RK4(num_eq, Y, H)`

**Sejam**

`K1[1..num_eq]`, `K2[1..num_eq]`, `K3[1..num_eq]` e `K4[1..num_eq]`: as derivadas  
`CalcF(I, Y)`: função externa que calcula  $f_i(Y[1], Y[2], \dots, Y[num\_eq])$

/\* Cálculo de `k1` \*/

**Para** `I = 1 até num_eq`  
`K1[I] = CalcF(I, Y)`

/\* Cálculo de `k2` \*/

**Para** `I = 1 até num_eq`  
`K2[I] = CalcF(I, Y[I] + 0.5 * H * K1[I])`

/\* Cálculo de `k3` \*/

**Para** `I = 1 até num_eq`  
`K3[I] = CalcF(I, Y[I] + 0.5 * H * K2[I])`

/\* Cálculo de `k4` \*/

**Para** `I = 1 até num_eq`  
`K4[I] = CalcF(I, Y[I] + H * K4[I])`

/\* Cálculo de `Yi+1` \*/

**Para** `I = 1 até num_eq`  
`Y[I] = Y[I] + (H/6) * (K1[I] + 2*K2[I] + 2*K3[I] + K4[I])`