

LARISSA ROCHA PEREIRA

**AJUSTE DE CURVA B-SPLINE FECHADA COM
PESO**



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA
2014

LARISSA ROCHA PEREIRA

AJUSTE DE CURVAS B-SPLINE FECHADA COM PESO

Dissertação apresentada ao Programa de Pós-graduação em Engenharia Mecânica da Universidade Federal de Uberlândia, como parte dos requisitos para obtenção do título de **MESTRE EM ENGENHARIA MECÂNICA.**

Área de concentração: Mecânica dos Sólidos e Vibrações.

Orientador: João Carlos Mendes Carvalho

**UBERLÂNDIA – MG
2014**

AGRADECIMENTOS

Agradeço aos meus pais pelo amor, apoio, confiança e motivação incondicional.

Agradeço ao João Carlos pela brilhante orientação neste trabalho, pelas oportunidades, confiança e incentivo ao longo deste período.

Aos meus amigos que sempre estiveram me apoiando, ajudando e proporcionando momentos de descontração quando se fazia necessário.

Ao Laboratório de Automação e Robótica (LAR - UFU) por me proporcionar conhecer pessoas que me ajudaram e me deram força para seguir a diante.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro.

À Universidade Federal de Uberlândia e ao Programa de Pós-graduação em Engenharia Mecânica pela oportunidade de realizar este curso e desenvolver esse trabalho.

E a todos que direta ou indiretamente estiveram ao meu lado e acreditaram em mim.

PEREIRA, L. R. **Ajuste de Curvas B-spline Fechada com Peso**. 2014. 165f. Dissertação de Mestrado, Universidade Federal de Uberlândia, Uberlândia.

RESUMO

O objetivo desse trabalho é desenvolver um método de ajuste de curvas B-spline fechada para a aplicação na reconstrução de seções transversais de um objeto. Por especificamente nesse trabalho as seções serem seções fechadas, foi implementado um método para o fechamento da curva B-spline, de modo que a mesma possuía suavidade no seu fechamento. O método desenvolvido e utilizado foi baseado na aproximação por mínimos quadrados com pesos, que define que a curva obtida deva ser mais próxima possível da curva real. Os pesos nesse caso são responsáveis pela aproximação ou afastamento da curva em relação aos pontos dados, pontos esses que melhor representam as coordenadas da seção do objeto que se deseja reconstruir. Além disso, foram desenvolvidos ajustes e imposições na curva para que ela tivesse um melhor resultado e representasse de forma mais fiel a seção transversal desejada. Para a imposição e definição dos ajustes foram utilizadas características particulares da curva. Para a análise, curvas B-spline utilizando o método desenvolvido, foram traçadas e foram constatados os resultados desejados.

Palavras – chave: B-spline fechada. Ajustes de curva. B-spline com peso. Mínimos quadrados.

LARISSA, L. R. **Curve Fitting whith Closed Weigthed B-spline**. 2014. 165f. Master Thesis, Universidade Federal de Uberlândia, Uberlândia.

ABSTRACT

The aim of this work is to develop a method of curve fitting using closed B-spline closed for application on reconstruction of cross-sections of objects. For this study specifically where the sections are closed curves, it has been implemented a method to close the curve B-spline curve, in such way that the curve is smooth on the closing point. The developed method is based on least squares approximation with weights, which defines that the curve should be as close as possible to the real curve. The weights in this case are responsible for the “tightness” of the approximation to each data points, whose points represent the coordinate of the object section that will be rebuild. Moreover, adjustments and impositions on the curve have been proposed so that it has a better result and represent more accurately the desired cross section. Particular characteristics of the curve were used to help enforce and define the settings. For the analysis, B-spline curves using the developed method, were obtained showing good results.

LISTA DE FIGURAS

Figura 2.1 - Pilha de seções planares obtidas por Tomografia Computadorizada (HOPPE, 1994)	7
Figura 2.2 - Plano cortando o objeto de interesse (CESAR, 1994)	8
Figura 2.3 – Curva obtida em uma secção (CESAR, 1994)	8
Figura 2.4 - Braço robótico (http://valentinerobotics.wordpress.com/)	10
Figura 2.5 – (a) CMM manual, (b) Estação de CMM (http://www.itpgroup.co.uk/products/new_cmms/gantry.html)	10
Figura 2.6 – Etapas da digitalização de um manequim (HSIAO; CHEN, 2013)	16
Figura 3.1 - a) Interpolação; b) Aproximação (FERREIRA, 2011)	19
Figura 3.2 – Exemplo de curva paramétrica (BARBARINI, 2007)	20
Figura 3.3 – Exemplo de curva com pontos e derivadas definidas (BARBARINI, 2007)	21
Figura 3.4 - Exemplo de curva com laço (BARBARINI, 2007)	21
Figura 3.5 – Exemplo de curva que passa por pontos dados (BARBARINI, 2007)	21
Figura 3.6 – Exemplo de uma curva de Hermite (BARBARINI, 2007)	22
Figura 3.7 – Geometria de composição da função de Hermite	23
Figura 3.8 – Exemplo de diferentes curvas formadas alterando-se apenas o vetor T_1 (MOTTA, 2010)	25
Figura 3.9 – Curva de Bézier de primeiro grau	27
Figura 3.10 – Curva de Bézier de segundo grau	27
Figura 3.11 - (a) e (b) Exemplos de curvas de Bézier cúbicas	28
Figura 3.12 - Obtendo um ponto na curva quadrática de Bézier por uma interpolação linear em $u_0=2/5$ (PIEGL; TILLER, 1997)	30
Figura 3.13 - Interpolação linear em $u_0=2/5$ de uma curva de Bézier cúbica (PIEGL; TILLER, 1997)	30
Figura 3.14 – Situação inicial (WANGNHEIM; WAGNER, 2013)	31
Figura 3.15 – $P_1^{(1)}$ o ponto médio do segmento $\overline{P_0P_1}$ (WANGNHEIM; WAGNER, 2013)	31

Figura 3.16 – $P_2^{(1)}$ o ponto médio do segmento $\overline{P_1P_2}$ (WANGNHEIM; WAGNER, 2013)	32
Figura 3.17 – $P_2^{(2)}$ o ponto médio do segmento $\overline{P_1^{(1)}P_2^{(1)}}$ (WANGNHEIM; WAGNER, 2013)	32
Figura 3.18 – Cria-se uma nova sub-curva à esquerda, a qual começa a subdividir (WANGNHEIM; WAGNER, 2013)	33
Figura 3.19 – Subdividir à esquerda (WANGNHEIM; WAGNER, 2013)	33
Figura 3.20 – Subdivisão do novo $\overline{P_1P_2}$ (WANGNHEIM; WAGNER, 2013)	33
Figura 3.21 – $P_2^{(2)}$ o ponto médio do segmento $\overline{P_1^{(1)}P_2^{(1)}}$ (WANGNHEIM; WAGNER, 2013)	34
Figura 3.22 – Sequência do algoritmo (WANGNHEIM; WAGNER, 2013)	35
Figura 3.23 – Interpolação por spline (FERREIRA, 2011)	36
Figura 3.24 – Interpolação com spline quadrática. Interpolação com os pontos originais (linha contínua), interpolação com a mudança do sétimo ponto P (linha tracejada) (FERREIRA, 2011)	38
Figura 3.25 – B-spline quadrática. A linha cheia mostra a curva original, a linha pontilhada representa a curva com a mudança do ponto de controle P_4 (FERREIRA, 2011)	39
Figura 3.26 – Funções base de grau zero, $U = \{0,0,0,1,2,3,4,4,5,5,5\}$, (PIEGL; TILLER, 1997)	42
Figura 3.27 – Funções base do primeiro grau, $U = \{0,0,0,1,2,3,4,4,5,5,5\}$, (PIEGL; TILLER, 1997)	43
Figura 3.28 – Funções de base do segundo grau, $U = \{0,0,0,1,2,3,4,4,5,5,5\}$ (PIEGL; TILLER, 1997)	44
Figura 3.29 – Esquema da propriedade de suporte local (PIEGL; TILLER, 1997)	45
Figura 3.30 – Esquema propriedade 2 (PIEGL; TILLER, 1997)	45
Figura 3.31 – Função base do segundo grau, $U = \{0,0,0,1,2,3,4,4,5,5,5\}$ (PIEGL; TILLER, 1997)	47
Figura 3.32 – Classificação das curvas em aberta, “clamped” e fechada	50

(BARBARINI, 2007)	
Figura 3.33 – Curva aberta quadrática (PIEGL; TILLER, 1997)	51
Figura 3.34 – Exemplo de curva fechada	52
Figura 3.35 – B-spline cúbica com $U = \{0,0,0,0,1,1,1,1\}$, i.e., Curva de Bézier	53
Figura 3.36 – Propriedade do fecho convexo para uma curva B-spline quadrática; para $u \in [u_i, u_{i+1})$, $C(u)$ está no triângulo $P_{i-2}P_{i-1}P_i$, (PIEGL; TILLER, 1997)	53
Figura 3.37 – (a) Propriedade do fecho convexo (ROGERS, 2001); (b) Curvas B-spline definidas pelo fecho convexo	54
Figura 3.38 – Curva Cúbica em $U=\{0, 0, 0, 0, 1/4, 1/2, 3/4, 1, 1, 1, 1\}$; movendo P_4 (para P'_4) a curva muda no intervalo $[1/4, 1)$	55
Figura 3.39 – (a) Curva de grau 9 usando vetor nó $U=\{0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\}$; (b) Curva quadrática usando o mesmo polígono de controle definido em $U=\{0\ 0\ 0\ 1/8\ 2/8\ 3/8\ 4/8\ 5/8\ 6/8\ 7/8\ 1\ 1\ 1\}$	56
Figura 3.40 – B-spline cúbica em $U=\{0, 0, 0, 0, 1/4, 3/4, 1, 1, 1, 1\}$ com a derivada primeira, segunda e terceira computadas em $u=2/5$ (PIEGL; TILLER, 1997)	57
Figura 3.41 – Interpolação com diferentes metodologias de parametrização do vetor \bar{U} (FERREIRA, 2011)	61
Figura 3.42 – Interpolação com diferentes metodologias de obtenção do vetor U (FERREIRA, 2011)	61
Figura 3.43 – Interpolação por B-spline cúbica usando parametrização por comprimento de corda e vetor nós obtido pelo método de <i>averaging</i>	62
Figura 3.44 – Mudança do peso w_3 relacionado ao ponto de controle P_3	67
Figura 3.45 – Influência do peso no comportamento da curva (FERREIRA, 2011)	68
Figura 4.1 - Taxonomia dos métodos ativos de aquisição de formas (CURLESS, 1997)	72
Figura 4.2. Diagrama da cadeia cinemática do digitalizador desenvolvido no LAR (estrutura tipo RRP+PR) (DÍAZ, 2011)	73
Figura 4.3 – Estrutura mecânica do digitalizador desenvolvido no LAR	74
Figura 4.4 – Exemplo de nuvem de pontos adquirido pelo sensor laser para	75

uma distância fixa	
Figura 4.5 – Pontos adquiridos e regiões (AQUINO et al., 2009)	76
Figura 4.6 – Regiões e seus centros de massa (AQUINO et al., 2009)	77
Figura 4.7–Definição da macrorregião para o início do processo do método de detecção de contorno, a partir da região 1 (FONSECA; CARVALHO, 2007)	78
Figura 5.1 – Pontos para definição do vetor normal (DÍAZ, 2011)	83
Figura 5.2 – Pontos de inversão de curvatura	84
Figura 5.3 – Quantidade de pontos de controle definido pelos pontos de inversão de curvatura	85
Figura 5.4 – Circunferência e raio definidos por três pontos sequenciais de Q_k	85
Figura 5.5 - B-spline cúbica aproximada com os pontos sem considerar os pesos	89
Figura 5.6 – B-spline cúbica aproximada considerando os pontos com peso	89
Figura 5.7 – (a), (b) e (c) Curvas B-splines para diferentes pontos iniciais	91
Figura 5.8 – Pontos Q_k para a definição do ponto inicial	92
Figura 5.9 – Triângulo formado por Q_k , Q_{k+1} e Q_{k+2} . Cálculo do ângulo formado pelos segmentos $\overline{Q_k Q_{k+1}}$ e $\overline{Q_{k+1} Q_{k+2}}$	93
Figura 5.10 – Pontos considerados alinhados	95
Figura 5.11 – Distâncias entre os pontos de controle e a reta tangente	96
Figura 5.12 – Ponto inicial considerando o alinhamento entre os pontos dados	97
Figura 5.13 – Curva B-spline com alteração na região inicial considerando a direção da tangente	98
Figura 5.14 – Zoom da região inicial considerando a direção da tangente	98
Figura 5.15 – Curva B-spline com os pontos de controle P_1 e P_2 modificados	99
Figura 5.16 – Zoom da região inicial com os pontos de controle P_1 e P_2 modificados	100
Figura 5.17 – Repetição da Figura 5.3	101
Figura 5.18 – Curva obtida considerando a quantidade de pontos de	101

controle igual a 50% dos pontos dados, os pesos e a definição do ponto inicial conforme apresentado	
Figura 5.19 – Pontos conhecidos que definem a seção do objeto – exemplo 1	102
Figura 5.20 – Curva B-spline sem ajustes (exemplo1)	103
Figura 5.21 – Curva B-spline cúbica final para os dados do exemplo 1. (a) Curva B-spline com os pontos que a definem e os pontos de controle; (b) Curva B-spline com pontos e sem os pontos de controle	104
Figura 5.22 - Pontos conhecidos que definem a seção do objeto – exemplo 2	105
Figura 5.23 - Curva B-spline sem ajustes (exemplo2)	106
Figura 5.24 – Curva B-spline cúbica final para os dados do exemplo 2. (a) Curva B-spline com os pontos que a definem e os pontos de controle; (b) Curva B-spline com pontos e sem os pontos de controle	107
Figura 5.25 – Pontos conhecidos que definem a seção do objeto - exemplo 3	108
Figura 5.26 – Curva B-spline sem ajustes (exemplo 3)	108
Figura 5.27 – Curva B-spline cúbica final para os dados do exemplo 3. (a) Curva B-spline com os pontos que a definem e os pontos de controle; (b) Curva B-spline com pontos e sem os pontos de controle	109
Figura 5.28 - Pontos conhecidos que definem a seção do objeto - exemplo 4	110
Figura 5.29 – Curva B-spline sem ajustes (exemplo 4)	110
Figura 5.30 – Curva B-spline cúbica final para os dados do exemplo 4. (a) Curva B-spline com pontos e pontos de controle; (b) Curva B-spline com pontos e sem os pontos de controle	111

LISTA DE SÍMBOLOS E SIGLAS

\bar{u}_k	Parâmetro de interpolação da B-spline.
$N_{i,p}^{(k)}$	k -ésima derivada da função de base da B-spline.
\bar{U}	Vetor contendo todos os \bar{u}_k .
$B_{i,n}(u)$	Polinômios de Bernstein de grau n .
C	Matriz com os coeficientes de Hermite.
$C(u)$	Curva paramétrica.
$C'(0)$	Derivada do ponto inicial.
$C'(1)$	Derivada do ponto final.
C^k	Continuidade paramétrica de ordem k .
$C^{(k)}(u)$	k -ésima derivada da curva B-spline.
CM	Centro de Massa.
CAD	<i>Computer Aided Design</i> .
CAM	<i>Computer Aided Manufacturing</i> .
CCD	<i>Charge-Coupled Device</i> .
CCM	<i>Coordinate Measuring Machines</i> .
ℓ	Comprimento total de corda.
d_c	Somatório das raízes quadradas dos comprimentos de corda.
d_{t1}	Distância do ponto de controle P_1 à reta tangente.
d_{t2}	Distância do ponto de controle P_2 à reta tangente.
$FEMEC$	Faculdade de Engenharia Mecânica.
\vec{G}	Vetor normal aos pontos Q_k que definem a mudança de curvatura.
G	Matriz geométrica.

LAR	Laboratório de Automação e Robótica.
$N_{i,p}(u)$	Função de base da B-spline de grau p no vão i .
m	Último índice do vetor nó.
M_H	Matriz de Base de Hermite.
n	Último índice dos pontos de controle
p	Grau da curva B-spline.
P_i	Pontos de controle que definem uma curva, onde $(i = 1, \dots, n)$.
PR	Estrutura com uma articulação de rotação e uma prismática.
Q_k	Pontos de interpolação ou aproximação de uma B-spline.
$R_{i,p}(u)$	Função de base da NURBS de grau p no vão i .
RRP	Estrutura com duas articulações de rotação e uma prismática.
$S_i(x)$	i -ésimo segmento de uma spline.
T	Matriz de tempo.
t	Tempo.
u	Parâmetro de uma curva paramétrica (nó).
U	Vetor dos nós da B-spline.
UFU	Universidade Federal de Uberlândia.
W	Matriz diagonal de pesos ω_k
ω_k	Peso associado ao ponto Q_k em uma curva B-spline.
ω_i	Peso associado ao ponto de controle P_i em uma curva NURBS.
x_{CM}	Coordenada x do centro de massa.
y_{CM}	Coordenada y do centro de massa.
α	Ângulos formados pelos pontos Q_k .
$\bar{\alpha}$	Média dos ângulos formados pelos pontos Q_k .
\bar{r}	Média dos raios das circunferências formadas pelos pontos Q_k .
σ	Desvio padrão.

SUMÁRIO

Resumo	iii
Abstract	iv
Lista de figuras	v
Lista de Símbolos e Siglas	x
1 CAPÍTULO I	1
INTRODUÇÃO	1
2 CAPÍTULO II	5
REVISÃO BIBLIOGRÁFICA	5
2.1 Introdução	5
2.2 Reconstrução de superfícies	6
2.3 Digitalizadores	9
2.4 Métodos de reconstrução de superfícies usando B-spline e NURBS	11
2.4.1 Reconstrução de seções transversais	11
2.4.2 Reconstrução de superfícies	14
2.5 Conclusão	16
3 CAPÍTULO III	18
AJUSTE DE CURVAS	18
3.1 Introdução	18
3.2 Curvas paramétricas	20
3.3 Curvas de Hermite	22
3.4 Curvas de Bézier	26
3.5 Spline	35
3.6 B-spline	39
3.6.1 Propriedades importantes das funções de base B-spline	44
3.6.2 Derivadas das funções de base B-spline	48
3.6.3 Curvas B-spline	49
3.6.4 Propriedades das Curvas B-spline	53

	3.6.5 Derivada da curva B-spline	57
	3.6.7 Interpolação com B-spline	58
	3.6.8 Aproximação com B-spline	62
	3.7 NURBS (<i>Non-Uniform Rational B-spline</i>)	65
	3.8 Conclusão	68
4	CAPÍTULO IV	70
	DIGITALIZAÇÃO DE CURVAS E SUPERFÍCIES	70
	4.1 Introdução	70
	4.2 Sistemas de digitalização tridimensionais	71
	4.3 O Digitalizador Desenvolvido no LAR	72
	4.4 Redução dos pontos lidos	75
	4.5 Conclusão	78
5	CAPÍTULO V	80
	APLICAÇÃO DE B-SPLINE FECHADA COM PESO NA RECONSTRUÇÃO DAS SEÇÕES TRANSVERSAIS	80
	5.1 Introdução	80
	5.2 Desenvolvimento do método	81
	5.3 Metodologia de ajuste de curva	83
	5.3.1 Definição da quantidade de pontos de controle em função do número de inversões da curva	83
	5.3.2 Definição dos pesos	85
	5.3.3 Definição do ponto inicial	90
	5.3.4 Ajuste da quantidade dos pontos de controle	100
	5.4 Aplicação da B-spline nas seções transversais	102
	5.4.1 Exemplo 1	102
	5.4.2 Exemplo 2	105
	5.4.3 Exemplo 3	107
	5.4.4 Exemplo 4	110
	5.5 Resumo da metodologia de ajuste de curvas fechadas	112
	5.6 Conclusão	113

6	CAPÍTULO VI	114
	CONCLUSÕES	114
	6.1 Conclusões	114
	6.2 Sugestões para trabalho futuros	115
7	REFERÊNCIAS BIBLIOGRÁFICAS	116
8	APÊNDICE	124
	Apêndice I – Função que Encontra o Vão em que o Nó se Encontra	124
	Apêndice II – Programa que Calcula as Funções de Base não Nulas	125
	Apêndice III – Função que multiplica as funções base não nulas pelos seus respectivos pontos de controle e plota a curva	127
	Apêndice IV – Programa que calcula uma curva fechada	128
	Apêndice V – Desenvolvimento da Eq. (3.47)	129
	Apêndice VI – Exemplos de aplicação do método desenvolvido	131
	Apêndice VI – Programa Final	137

CAPÍTULO I

INTRODUÇÃO

A Engenharia Reversa surgiu da necessidade de substituir ou modificar um produto já existente, cujas informações sobre seu desenvolvimento e fabricação são desconhecidas, por outro produto com as mesmas características. Os produtos obtidos através dessa tecnologia oferecem alta qualidade, baixo custo e tempo de obtenção reduzido, despertando grande interesse de diversas áreas. A busca incessante da indústria em reduzir os custos e o tempo no desenvolvimento do produto, mantendo ou aumentando a qualidade do mesmo, tornou a engenharia reversa uma tecnologia poderosa na obtenção de sistemas produtivos mais ágeis e flexíveis. Assim, o número de pesquisas em engenharia reversa cresceu em todo mundo objetivando aperfeiçoar os processos envolvidos nesta tecnologia e aplicá-la às mais diversas áreas existentes.

Esta tecnologia envolve três etapas básicas, que são: digitalização de forma (aquisição de dados), pré-processamento (tratamento dos dados) e geração da superfície (reconstrução gráfica).

A aquisição de dados é realizada por sistemas de medição de coordenadas (ALVES, 1996), que podem ser de dois tipos: de contato e sem contato. Na medição com contato o sistema possui um apalpador esférico que sempre deve “tocar” o objeto para a aquisição de pontos de sua superfície. Dependendo do formato geométrico, este sistema não consegue realizar leituras de todos os pontos da superfície. Máquina sem contato é aquela em que, ao invés de um apalpador, emite um sinal em forma de onda sobre o objeto tais como câmeras, Raio X, laser, etc. A resolução deste sistema é muito maior que do sistema por contato e pode ser utilizado para digitalização de diversos tipos de objetos, inclusive os macios, sem a preocupação de danificá-los.

Fazer uma escolha de qual sistema ou qual método é o mais apropriado para fins práticos às vezes resulta em uma tarefa difícil, já que o sistema selecionado deve ler uma variedade considerável de formas.

O resultado da leitura realizada por um sistema de medição de superfícies consiste em uma “massa” de dados, em geral com ruídos, denominado de nuvens de pontos, nem sempre organizados e que requerem algoritmos eficientes e confiáveis no seu tratamento e eliminação de pontos desnecessários.

Em alguns casos é necessário definir alguns parâmetros para que o algoritmo possa solucionar o problema de reconstrução da superfície. Por esta razão, o método de reconstrução depende da aplicação e, para cada aplicação deve ser usado o algoritmo adequado. A reconstrução da superfície é um problema difícil: primeiro porque os pontos medidos são desorganizados e, geralmente, cheios de ruídos; depois, a superfície pode ser arbitrária, com topologia desconhecida e com formas angulares. Então, o método de reconstrução deve identificar a geometria correta, a topologia e a forma a partir de um número finito de pontos (REMONDINO, 2003).

De forma a eliminar estes problemas, é necessário reduzir o número de pontos visando extrair o conjunto de dados que melhor descrevem a superfície digitalizada. Existem alguns métodos que reduzem os pontos que definem o contorno do objeto. O método desenvolvido no Laboratório de Automação e Robótica da FEMEC/UFU, e utilizado neste trabalho consiste na substituição de um conjunto de pontos existentes em uma determinada região, por seu centro de massa (CM), considerando que cada ponto possui massa unitária. Assim, é obtido um conjunto com reduzido número de pontos, representado pelos CM de cada região, que é organizado de maneira a representar fielmente o contorno real de seções transversais da superfície.

Mesmo com os dados já tratados, é impossível traçá-los manualmente a fim de recuperar a imagem desejada. Recuperação por forma é considerada um dos aspectos mais difíceis e desafiadores. Assim, é necessário um sistema eficaz que se assemelha à capacidade humana na recuperação do objeto a partir de um banco de dados relevantes (LIANG et al., 2002b).

Existem diversas técnicas de processamento dos dados para a reconstrução de superfície através de nuvens de pontos. Apesar dos grandes avanços nesta área, a correta modelagem de superfícies fechadas ou de objetos de formas complexas não está completamente solucionada e ainda constitui uma área de muitas atividades de pesquisa (REMONDINO, 2003). Várias

publicações vêm sendo feitas, principalmente no uso de técnicas como NURBS (*Non Uniform Rational B-spline*) e B-splines, sendo algumas abordadas no Cap. II.

Curvas B-spline e NURBS são versões da spline que implementam o controle local da curva, de forma que a alteração de um ponto de controle modifica a curva somente na região próxima ao ponto alterado.

A B-Spline, caso particular de curva NURBS, é um ajustador de aproximação, pois a curva gerada não passa necessariamente pelos pontos de controle. No entanto, os pontos de controle podem ser calculados para que a interpolação ou aproximação seja realizada nos pontos desejados.

As curvas NURBS possuem as mesmas características que as B-splines com um acréscimo de pesos associados a cada ponto de controle. Assim curvas NURBS possuem maior flexibilidade, pois além dos pontos de controle a curva pode ser alterada com a mudança desses pesos. Se todos os pesos forem iguais a 1, uma curva NURBS reduz para uma curva B-spline.

Utilizando B-splines e NURBS é possível obter uma superfície aproximada ou interpolada utilizando os pontos oriundos da digitalização de um objeto.

A curva B-spline se destaca como um dos métodos de representação de forma (superfície) mais eficiente. As seguintes propriedades tornam-na atraente e adequada para a representação de forma: suavidade e continuidade, o que permite que qualquer curva consista de uma concatenação de segmentos de curvas que podem ser tratados como uma só unidade; facilidade de especificar o intervalo de uma curva de valores múltiplos; invariância de forma de transformação afim, que significa que a curva de transformação afim é ainda um B-spline cujos pontos de controle estão relacionados com os pontos de controle do objeto através da transformação; controlabilidade local, o que implica que as mudanças locais da forma estão confinadas aos parâmetros locais dessa mudança (COHEN et al., 1995).

Ainda que seja comum verificar na bibliografia sobre o tema que, tanto no traçado de curvas como de superfícies utilizando B-splines, existem diversas vantagens de sua utilização, ainda existem muitas dificuldades na sua implementação, principalmente na reconstrução de curvas e superfícies fechadas.

Embora as curvas B-spline tenham que atender a pré-requisitos matemáticos bem definidos, a curva final, ou seja, o ajuste final da curva depende de aspectos estéticos e requerimentos funcionais que, em geral, não podem ser totalmente definidos em termos de critérios quantitativos, ficando na dependência do “gosto visual” do operador.

Neste sentido, este trabalho visa obter um ajuste de curva B-spline sem a interferência do operador. Assim, o objetivo desse trabalho, portanto, é utilizar curvas B-spline para reconstruir seções transversais de um objeto a partir de um conjunto de pontos que, em princípio, representam essa seção transversal. A partir das seções transversais é possível, posteriormente, reconstruir a superfície do objeto pelo método conhecido como reconstrução através de fatias. O procedimento utiliza o método de pesos associados aos pontos previamente definidos (pontos que melhor representam as coordenadas do objeto) e do ajuste de curva por aproximação pelo método dos mínimos quadrados.

Os pesos são usados para aproximar a curva dos pontos digitalizados a fim de diminuir o erro entre eles, e assim permitir obter a melhor curva que represente a seção transversal do objeto.

Para o ajuste das curvas foram realizadas simulações computacionais utilizando B-spline fechada com diferentes pontos pré-definidos para a definição da curva, permitindo definir métodos de ajustes baseados em características específicas da curva, para que ela se torne um modelo mais próximo possível do real.

Para que os objetivos deste trabalho sejam atingidos, no Capítulo II é apresentada uma revisão sobre a utilização de B-spline e NURBS para a representação de curvas e superfícies; no Capítulo III é feito um estudo das metodologias de ajuste de curva, com ênfase à B-spline e NURBS, visando suas aplicações; no Capítulo IV é demonstrado o processo de digitalização de forma (aquisição de dados) e o pré-processamento (tratamento dos dados) mostrando quais estruturas e métodos foram utilizados; no Capítulo V é desenvolvido o método de aproximação para a B-spline fechada com peso, assim como seus ajustes, e são realizadas simulações aplicando a curva; finalmente, no Capítulo VI, as conclusões e considerações para trabalhos futuros.

CAPÍTULO II

REVISÃO BIBLIOGRÁFICA

2.1 Introdução

Reconstrução de superfícies é um problema que surge em inúmeras aplicações científicas e de engenharia, como por exemplo: indústria aeroespacial, automóveis, biomédica, sistemas de aquisição volumétrica (scanner 3D) (FRANÇA et al., 2005), engenharia reversa e geração automática de modelos CAD a partir de modelos físicos (SON; PARK; LEE, 2002), imagens médicas, visualização de dados científicos, cartografia, simulações numéricas de equações diferenciais parciais (GOIS, 2004; MALISKA, 1995) e indústrias de produtos de consumo para facilitar o design de produto, análise e produção de produtos pré-existentes (HUANG; QIAN, 2007).

Reconstrução de superfícies pode ser definida como um processo que gera uma malha que represente corretamente a forma e a topologia do objeto original, de onde é amostrado um conjunto de pontos (MARI, 2007). Definir a forma de um objeto é uma tarefa difícil, especialmente na era do computador que introduziu a necessidade de descrever formas ainda muito complicadas com precisão e tempo de computação mais rápido (LIANG; KHOO; RAJESWARI, 2002), e obter essa forma partindo apenas de pontos desconexos não é uma tarefa trivial. Muitas vezes é necessário que o conjunto de pontos amostrado passe por um estágio de pré-processamento, para garantir o sucesso da reconstrução. Este estágio tem a função de processar os “dados numéricos brutos” obtidos durante o sensoriamento, para que, a partir dela, possam ser extraídas as informações necessárias para a fase de reconstrução de superfícies. Essas informações (conjunto de pontos) são muitas vezes denominadas nuvens de pontos.

Desenvolver, testar e expandir métodos que sejam robustos e eficazes para a reconstrução de superfícies, a partir das informações obtidas na fase de pré-processamento, é um importante campo de pesquisa na atualidade, mesmo que seja direcionado para um tipo específico de dispositivo de aquisição e/ou objeto alvo (MARI, 2007).

2.2 Reconstrução de superfícies

O objetivo da reconstrução da superfície é encontrar uma superfície a partir de um determinado conjunto finito de valores obtidos de uma amostra geométrica.

Duas classes de aplicações podem ser distinguidas quando se fala acerca da qualidade de uma superfície reconstruída. O primeiro é que a superfície da qual as amostras foram retiradas é conhecida. A situação de uma superfície conhecida é dada, por exemplo, na engenharia reversa da engenharia mecânica, onde o objetivo é obter uma superfície reconstituída com a mesma forma que a original amostrada. A segunda categoria é que as amostras vêm sem qualquer informação adicional. Nesse caso, a reconstrução da superfície é um método de análise de dados. Neste contexto, pode ser útil não só para oferecer uma superfície como uma solução, mas uma variedade de superfícies, a partir do qual o utilizador pode selecionar as mais razoáveis. Em ambos os casos, o objetivo da reconstrução da superfície está em visualizar a “forma escondida” nas amostras de dados de uma forma que é mais intuitiva do que exibir imediatamente os dados da amostra.

Em muitas aplicações, os valores das amostras possuem ruídos, isto é, que se desviam da localização correta devido a erros de medição. Dados com ruídos é uma das razões para uma classificação das soluções do problema de reconstrução da superfície: a reconstrução da superfície interpolada ou reconstrução da superfície de aproximação. Se uma abordagem de reconstrução da superfície interpolada é escolhida, a sua sensibilidade a dados com ruídos tem de ser levada em consideração (MULLER, 1997).

Quando os pontos de dados são obtidos por medições de forma empírica, a aproximação é mais desejável do que interpolação, para a superação de erros, reduzindo o ruído e a complexidade computacional (FISCHER; MANOR, 1999).

Conforme (MULLER,1997) os métodos de reconstrução de superfície podem ser categorizados em três tipos:

Método automático: Isto é, sem intervenção do usuário. Polinômios e polinômios por partes (splines) são exemplos de método automático. Eles são frequentemente escolhidos

como candidatos à reconstrução da superfície devido às suas versatilidades, bem como suas propriedades e eficiências computacionais. Em especial, no caso de análise de dados, pode ser difícil para o usuário expressar seu objetivo de interpretação abrangente de uma maneira formal de avaliação automática. Pelo contrário, é mais razoável usar as habilidades humanas imediatamente para a seleção, mostrando possíveis soluções que permitem ao usuário intervir e dizer ao sistema como proceder.

Método semiautomático: é a edição geométrica interativa, por exemplo, reparar uma malha triangular de superfície reconstruída em locais em que um algoritmo automático não foi bem sucedido. Uma possibilidade mais indireta de intervenção é o controle de parâmetros interativos de um método automático. Esta função funciona bastante bem em áreas da superfície que não contêm descontinuidades.

Método manual: A abordagem de base para recriar um conjunto de compensações é criar manualmente uma superfície, compará-la e refiná-la, se necessário, até que chegue a um resultado satisfatório. Com esta abordagem, o designer pode construir uma superfície que tem a topologia correta (juntas, voltas circulares, curva cônica, seções retas, etc), sem depender de um programa automático de discernir todos esses recursos. Uma parte importante deste processo é que dá ao designer a possibilidade de fazer as mudanças necessárias.

Muitas aplicações de processamento de imagens e computação gráfica envolvem uma superfície reconstruída a partir de uma sequência de contornos planares. Exemplos típicos são técnicas baseadas em imagens tomográficas e ressonância magnética. Estes dispositivos geram uma sequência de imagens de corte e a partir dos contornos, a superfície do objeto tem de ser reconstruída, mantendo a estrutura original, conforme pode ser observado na Fig. 2.1.

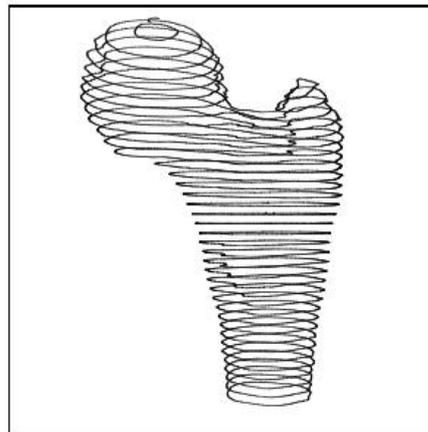


Figura 2.1 - Pilha de seções planares obtidas por Tomografia Computadorizada (HOPPE, 1994)

Esses contornos são obtidos de dados adquiridos por um dispositivo a partir do objeto tridimensional original, onde o usuário especifica os contornos em várias seções de uma superfície que deseja desenvolver. Pode-se encarar o processo de formação dos dados como a interseção entre um plano seccional e o objeto de interesse, como mostrado na Fig. 2.2. As informações contidas numa seção de um objeto são resultantes dessa intersecção, que pode ser pensada como uma imagem em níveis de cinza (PAVLIDIS,1982). Os contornos da superfície a ser reconstruída são obtidos através de um processo de segmentação e extração de contornos em cada seção, Fig. 2.3. Se o processo for repetido em posições diferentes do objeto, obtém-se um conjunto de seções deste, que unidas representam a superfície do objeto desejado.

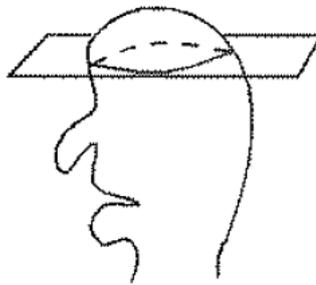


Figura 2.2 - Plano cortando o objeto de interesse (CESAR, 1994)

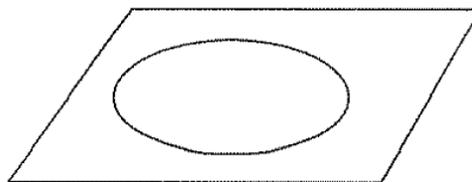


Figura 2.3 - Curva obtida em uma seção (CESAR, 1994)

Basicamente, todas as possibilidades de descrições formais podem ser utilizadas para a reconstrução da superfície. A descrição também escolhida depende de alguma forma do tipo, da qualidade da superfície desejada e de como as superfícies reconstituídas podem ser utilizadas. O tipo de descrição da superfície também influencia o algoritmo de reconstrução. Se ela deve ser colocada em um programa CAD para posterior projeto e modificações, a pessoa deve se esforçar para usar uma representação que seja compatível com o programa de CAD, por exemplo, uma superfície NURBS. Para a visualização em um computador, deve ser compatível com a função de processamento gráfico que é usado. Os algoritmos de

reconstrução, em geral, além manipular superfícies de topologia arbitrária, utilizar conjunto de pontos não-uniformes e produzir superfícies que garantam uma boa representação da forma original do objeto, ainda buscam satisfazer requisitos como eficiência de tempo e espaço (GOIS, 2004).

2.3 Digitalizadores

Um grande desafio para a reconstrução de uma superfície é adquirir um conjunto "ideal" de pontos, que representa o objeto físico o mais fielmente possível. Na maioria dos casos, designers criam curvas ou superfícies a partir de pontos de dados medidos. Estes dados medidos são frequentemente provenientes de um dispositivo de digitalização, p.ex. uma máquina de medição coordenada (CMM) ou um dispositivo de varredura a laser (MA; KRUTH, 1995). Assim, o digitalizador desempenha um papel importante no processo de reconstrução (DÍAZ, 2011).

Os digitalizadores tridimensionais são equipamentos que permitem capturar dados de determinada superfície permitindo realizar a construção ou reconstrução de objetos transformando-os em modelos virtuais (SILVA et al., 2010).

Os métodos de aquisição dos dados dos objetos podem ser por sensoriamento passivo ou ativo. A técnica passiva não interage com o objeto ao passo que o método ativo tem contato com objeto ou projeta alguma forma de energia sobre ele. Entre os métodos ativos pode-se distinguir: o uso de sensores de contato e sensores sem contato. Sensores de contato são tipicamente “pontas de prova” que são montados em uma estrutura articulada. Quando a ponta de prova toca a superfície do objeto, os dados de sua posição, obtidos pelos ângulos e parâmetros cinemáticos da estrutura articulada, são armazenados.

Os digitalizadores que utilizam estas pontas de prova podem ser identificados por dois grupos distintos: os sistemas robotizados, que consistem em utilizar um robô industrial onde seu elemento terminal é uma ponta de prova, Fig. 2.4, e os sistemas de medições de coordenadas conhecidos por CMM (*Coordinate Measuring Machines*), Fig. 2.5.



Figura 2.4 - Braço robótico

(<http://valentinerobotics.wordpress.com/>)

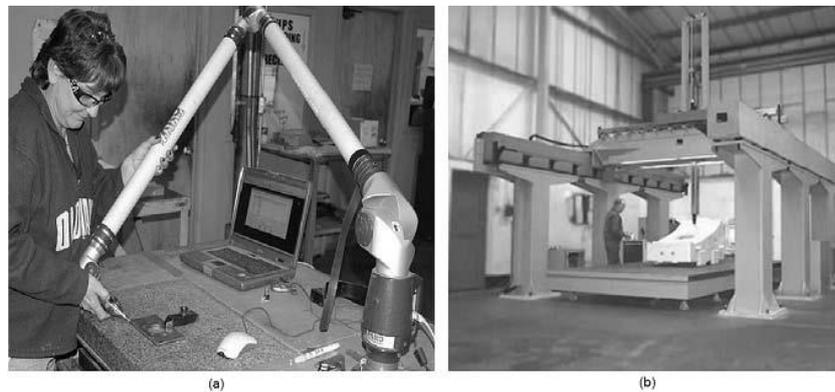


Figura 2.5 - (a) CMM manual, (b) Estação de CMM

(http://www.itpgroup.co.uk/products/new_cmms/gantry.html)

As técnicas sem contato podem ser divididas em dois grupos: transmissiva e reflexiva. As técnicas transmissivas consistem em fazer a energia atravessar o objeto para registrar sua forma tais como a tomografia, ressonância magnética e ultrassom, onde são feitas “radiografias” do objeto. Estas técnicas têm sido aplicadas dentro do campo da medicina e da engenharia (ARANA FERNANDEZ et al., 2006; ISAWA et al., 2005; NWODOH et al., 1997).

As técnicas sem contato reflexivas são aquelas que empregam algum tipo de radiação ou onda (eletromagnética ou mecânica) como fonte para examinar um objeto, onde a radiação ou onda refletida do objeto é captada por algum tipo de sensor. Podem ser classificadas como óticas e não óticas. Dentro das técnicas não óticas têm-se as magnéticas e as acústicas. Os digitalizadores óticos tridimensionais utilizam geralmente métodos como: triangulação, interferometria e análises da imagem (CURLESS, 1997).

A escolha de um digitalizador deve ser feita tendo em conta os seguintes aspectos: redução da quantidade de dados “lidos”, e redução dos erros e zonas escuras nas leituras. Esses aspectos são tipos de problemas encontrados em digitalizadores tridimensionais. Para a solução deles, procura-se desenvolver algoritmos especializados e utilizar novas técnicas de programação, inclusive na área da computação gráfica, obtendo-se soluções e algoritmos cada vez mais eficientes. Algumas vezes as correções do contorno da superfície são feitas manualmente (DÍAZ, 2011).

2.4 Métodos de reconstrução de superfícies usando B-spline e NURBS

Existem várias técnicas para a reconstrução de uma superfície, mas devido à reprodução fiel, a possibilidade de controle da curva e a sua utilização por diferentes maneiras, apenas as curvas NURBS e B-splines são apresentadas a seguir.

Curvas B-spline e NURBS são versões da spline que implementam o controle local da curva, de forma que a alteração de um ponto de controle modificará a curva somente na região dos pontos vizinhos mais próximos em função da ordem de continuidade. Estas curvas são ajustadores de aproximação, isto é, não passam pelos pontos de controle. No entanto, os pontos de controle podem ser calculados para que a interpolação seja realizada nos pontos desejados.

2.4.1 Reconstrução de seções transversais

Uma superfície pode ser ajustada segundo os pontos dos contornos de cada seção transversal, sendo eles abertos ou fechados. O essencial, então, é definir qual o melhor método a ser utilizado na reconstrução desses contornos. É procurada uma curva que se aproxima dos pontos, oriundos da digitalização, em que a distância da curva satisfaz um critério de qualidade de aproximação. Em aplicações práticas, uma tolerância pode ser especificada, a fim de se obter uma curva satisfatória em que a distância entre a curva e os pontos de dados devem ser menores do que essa tolerância (PARK; LEE, 2007). Nesse caso tem-se uma aproximação. Se a curva ajustada passa exatamente sobre os pontos, tem-se uma curva interpolada.

Existem muitas técnicas para representação e análise da forma da curva, tais como, descritores de Fourier, momentos, polinômios implícitos, geometria diferencial,

características da geometria diferencial, modelos de séries temporais, B-splines, NURBS, entre outros (COHEN et al., 1995).

Segundo (COHEN et al., 1995) e (LIANG et al., 2002a), as B-splines se portam como uma das mais eficientes representações de curva e superfície, pois possuem as seguintes propriedades:

- Representam formas livres com muito pouco dados e bem definidos em forma matemática;
- Tem controlabilidade local, o que implica que as alterações locais na forma da curva estão confinados apenas aos parâmetros locais da mudança.
- Tem a capacidade para controlar a suavidade e a continuidade da curvatura.
- Possui a característica de invariância de forma sob transformação afim, o que significa que a curva de transformação afim ainda é uma curva B-spline cujos pontos de controle estão relacionados com os pontos de controle das curvas originais.

Devido a isso, (COHEN et al., 1995), trabalharam na correspondência e reconhecimento de curvas planas modeladas com B-spline, independente de possíveis transformações afins a que a curva original foi submetida. Utilizou B-spline cúbica, com o método de otimização por mínimos quadrados iterativo para a diminuição de erro, obtendo resultados animadores e promissores.

Ma e Kruth (1995) discutem a parametrização dos dados digitalizados, ou seja, a determinação dos valores dos parâmetros de pontos digitalizados a fim de realizar ajuste por mínimos quadrados de curvas B-spline e superfícies.

Em (HUANG; TAI, 2000) a B-spline é usada como representação de curva e superfície, tendo uma aplicação prática na engenharia reversa, devido as exigências de aplicações no mundo real. Segundo os autores a B-spline é utilizada por possuir vantagens, como os pontos de controle, que influenciam a forma da curva em uma forma natural previsível, tornando-a boa candidata para uso em um ambiente interativo.

Em (CALIÒ et al., 2003) é utilizada a λ -spline que é parte integrante da B-spline, para a reconstrução de um perfil. Esta classe de curva é uma spline integrante dependendo de um parâmetro λ . O valor deste parâmetro, que é um número real, afeta todos os pontos de controle e, portanto, a forma global da curva. A fim de minimizar a diferença entre o modelo reconstruído e a curva original, um algoritmo de otimização foi aplicado para encontrar o valor ótimo do parâmetro de forma.

Em (PARK; LEE, 2007) foi utilizada uma abordagem de ajuste de curva por B-spline usando método dos mínimos quadrados. Este trabalho decorre de uma percepção de que alguns pontos adequadamente selecionados, a partir dos pontos dados, desempenham um papel importante originando melhores aproximações. A abordagem refina uma curva B-spline, selecionando menos pontos dominantes em regiões com poucas curvaturas, e mais em regiões com grande quantidade de curvaturas.

Chen et al. (2010) propuseram uma nova abordagem de aproximação para a B-spline cúbica a pontos previamente definidos. O método é conhecido como técnica de destravamento, onde a B-spline é aproximada aos pontos usando a tangente deles. A curva B-spline é aproximada às direções dos vetores derivados dos pontos ao mesmo tempo em que tende a aproximar-se da curvatura. O método mostrou-se simples e eficiente, mostrando que pode representar com precisão segmentos de linha reta.

Em (FURFERI et al., 2011) é utilizada a B-spline em uma nova abordagem de ordenação de pontos da nuvem de dados. Tal abordagem realiza uma aproximação ponderada de uma curva B-spline aos dados originais. A abordagem provou ser eficaz na preservação do original de acordo com a percepção humana.

Zheng et al. (2012) propuseram um método rápido de ajuste de curva para encontrar a B-spline que melhor se aproxima do contorno da forma do objeto, a partir de pontos de dados não organizados. O método usa a otimização iterativa de L-BFGS (método semi-Newton para resolver problemas de minimização não linear sem restrições) para otimizar os pontos de controle e o vetor nó ao mesmo tempo. Como resultado, eles obtiveram um novo método muito mais rápido que alguns métodos tradicionais.

Autores como (PIEGL; TILLER, 1987) usam NURBS para representar seções cônicas e círculos. Segundo os autores a NURBS foi escolhida devido esse tipo de curva oferecer uma representação precisa de formas analíticas padrão (linhas, cônicas, círculos, aviões e superfícies quádricas), bem como curvas de forma livre e superfícies; possuir graus extras de liberdade (os pesos), que podem ser usados para gerar uma grande variedade de formas; são generalizações de formas de B-spline, e a maioria das propriedades conhecidas e técnicas computacionais para as B-splines podem facilmente estender-se às formas racionais (NURBS).

Moustakides et al. (2000) propõem a reconstrução de uma superfície a partir de seções transversais determinadas por aproximação ou interpolação por curva NURBS. Esse tipo de

curva tornou-se um padrão da indústria principalmente porque elas podem representar as duas formas de forma livre e formas de análise comumente utilizados, tais como curvas cônicas.

Liang et al. (2002a) utilizam NURBS como descritor de forma. A escolha de NURBS é justificada porque esse tipo de curva, não só oferece uma forma matemática comum para representar formas de forma livre, mas também formas geométricas que tem cantos. Pelos resultados é evidente que NURBS é um descritor de forma eficiente.

Fonseca e Carvalho (2007) fizeram uma modelagem computacional para reconstrução gráfica de objetos com reduzido número de pontos utilizando as duas técnicas de reconstrução gráfica: NURBS e B-splines. Concluíram que a utilização das funções B-splines nas seções transversais fornecem uma superfície mais uniforme, devido sua capacidade de aproximação, sendo eficaz na minimização das distorções residuais, podendo ser utilizada em sistemas de leitura de precisão média. Enquanto a função NURBS, apesar de ter apresentado superfícies com pequenas distorções, é muito útil para sistemas de leituras de alta precisão, onde os dados adquiridos apresentam pequena margem de erro.

2.4.2 Reconstrução de superfícies

Alguns dos métodos de interpolação dos dados estão na situação em que os pontos não são dados num plano, mas sobre a superfície no espaço. Isto significa que uma superfície tem de ser reconstituída numa superfície (MULLER, 1997).

Stoddart et al. (1994) desenvolveram um tipo de algoritmo genérico para reconstrução de superfícies, baseado nas noções de superfície deformável. Esta técnica é mais conhecida como “Slime” e utiliza uma generalização do conceito de curvas biquadráticas B-splines. O método é eficiente na representação de descontinuidade, possui baixa complexidade computacional e pode ser facilmente adaptado. A desvantagem do método é a necessidade de conversão dos dados que se torna complicada devido aos ruídos da aquisição e da complexidade das técnicas de conversão.

Eck e Hoppe (1996) desenvolveram um procedimento para reconstruir uma superfície B-spline usando produto tensorial de um conjunto de pontos 3D digitalizados. Os autores definem a superfície como uma rede de manchas (*patches*) B-spline e demonstram que tal abordagem conduz a um processo eficiente para a montagem da superfície enquanto mantém a continuidade no plano tangente.

A fim de reconstruir a superfície do objeto, Fischer e Manor (1999) propõem um método que integra informações de uma câmera CCD com dados digitalizados a laser. O

método, após adquirir os pontos amostrados, reconstrói a superfície com B-spline pela aplicação de um método de aproximação mínimos quadrados com restrições de fronteira. Segundo os autores, a superfície digitalizada preserva as características da forma da superfície reconstituída.

Manchas (*patches*) B-splines são aplicadas por Douros et al. (1999) para a reconstrução de superfície de um corpo humano. Eles concluíram que a utilização de uma série de pequenas manchas B-spline é significativamente melhor do que a tentativa de utilizar uma superfície grande em torno do objeto.

Um algoritmo para reconstrução da superfície usando B-splines triangulares foi desenvolvido por (HE; QIN, 2004). Segundo os autores, o método possui vantagens: pode lidar com um domínio paramétrico com topologia arbitrária, gera um único *patch* B-spline triangular que tem continuidade sobre regiões suaves e características nítidas, demonstrados pelos resultados obtidos.

Huang e Qian (2007) apresentam uma nova abordagem para a digitalização 3D, permitindo a reconstrução dinâmica da superfície usando B-spline. A abordagem baseia-se na natureza recursiva do filtro de Kalman. Os experimentos demonstram que superfícies de alta qualidade podem ser obtidas de forma muito mais rápida, pois não é necessário armazenar a nuvem de pontos durante o processo de aquisição de dados, uma vez que todos os pontos são diretamente incorporados no modelo da superfície durante o processo de detecção.

Em (GÁLVEZ et al., 2012) foi apresentado um novo método para a reconstrução da superfície a partir de nuvens de pontos. O método utiliza um algoritmo genético de forma iterativa utilizando superfícies B-spline com o ajuste calculado por mínimos quadrados. As experiências mostram que o método proposto produz resultados muito bons, mesmo em presença de aspectos problemáticos. Em todos os exemplos os erros de ajuste são surpreendentemente pequenos.

Kineri et al. (2012) utilizam a B-spline para a reprodução da superfície de um capô de automóvel, usando interpolação, e de um manequim, usando aproximação, através de suas nuvens de pontos. O método é feito de forma iterativa com base em regras geométricas. O algoritmo atinge rapidamente um ajuste grosseiro com qualidade das superfícies resultantes geralmente boas.

Hsiao e Chen (2013) propõem um método para a reconstrução 3D da superfície de um manequim, onde ele é reconstruído através dos dados digitalizados e utilizando superfícies B-spline. A continuidade entre as superfícies é ajustada através do método de ajuste das

tangentes a fim de obter uma melhora na suavidade da superfície. Tal procedimento é feito utilizando as etapas da engenharia reversa, que envolve a digitalização do objeto, a organização dos dados e a reconstrução e ajustes da superfície. Na Figura 2.6 são apresentadas as etapas desse processo na reconstrução do manequim.

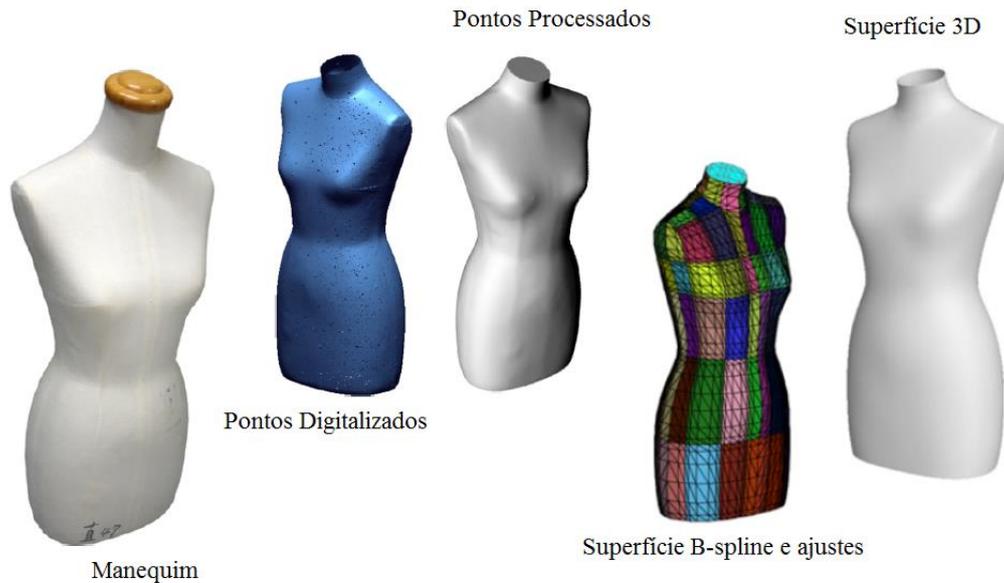


Figura 2.6 – Etapas da digitalização de um manequim (HSIAO; CHEN, 2013)

Em (MOUSTAKIDES et al., 2000), uma superfície 3D é reconstruída utilizando aproximação/interpolação de uma curva NURBS, utilizando o método “*averaging*”, juntamente com a otimização dos parâmetros da curva para um melhor ajuste. O sistema apresentado é aplicável tanto para superfícies abertas quanto fechadas.

2.5 Conclusão

A reconstrução tridimensional de objetos é um tema muito pesquisado e com amplas aplicações em áreas como arquitetura, medicina, odontologia, robótica, dentre outros. Definir os melhores métodos a serem utilizados nesse processo ainda constitui um campo de muitas atividades de pesquisa. Tais métodos devem possuir propriedades e características suficientes para que sejam capazes de reproduzir o objeto de forma mais fiel possível.

Existem vários métodos que são utilizados nessa reconstrução, tanto na reconstrução da superfície com um todo, quanto na reconstrução através de fatias. Porém, as B-splines e

NURBS tem sido alvo de estudos e muito úteis devido as suas características, tais como o controle local da curva e pesos associados a cada ponto de controle (no caso das curvas NURBS), sendo utilizadas por meio de aproximação ou interpolação.

A fim de obter a melhor reconstrução, cada autor utiliza a curva que melhor se adequa ao seu projeto. Muitas vezes elas se apresentam de forma adaptada ou juntamente com métodos e ajustes que melhoram sua eficiência, como no caso desse trabalho. Para um melhor resultado da curva, é utilizado, neste trabalho, um método em que a B-spline está associada com pesos, sendo estes relacionados aos pontos que representam as coordenadas do objeto. Esses pesos proporcionam um outro tipo de flexibilidade à curva, que permite a aproximação ou o afastamento da mesma em relação aos pontos, melhorando o perfil final do objeto que se deseja reconstruir, ou seja, minimizando o erro entre a curva construída e a curva original. Além dos pesos, a curva B-spline é apresentada de forma fechada, o que permite reconstruir seções de objetos, como por exemplo cilindros, que possuem superfícies fechadas. A B-spline é utilizada por seu processamento ser mais rápido que a curva NURBS, devido à simplicidade do modelo matemático.

Em cada caso, portanto, é necessário escolher qual o melhor tipo de curva e desenvolver métodos de ajustes para que a superfície ou a curva originada se torne mais próxima do real.

CAPÍTULO III

AJUSTE DE CURVAS

3.1 Introdução

Intuitivamente, o conceito de curva nos passa a ideia de um objeto, uma linha contínua no espaço. Rigorosamente, segundo Gleicher (2004), uma curva paramétrica é:

Um mapeamento contínuo de um espaço unidimensional para um espaço n -dimensional.

Desenhar curvas suaves para modelar uma topologia específica de interesse sempre foi considerado um esforço árduo e tedioso. Em 1959, Paul de Fagnet de Casteljaou, um funcionário da Citroën, deu o primeiro passo para a obtenção de curvas e superfícies suaves combinando e iterando malhas de polígonos para modelagem de chassis. Concomitantemente, na Renault, o empregado Pierre Bézier também obteve curvas para o mesmo fim, que atualmente são denominadas curvas de Bézier. A curva de Bézier somente ficou formalmente conhecida após a publicação do trabalho “*Emploi dès Machines a Commande Numerique*” em 1970 pelo próprio Pierre Bézier.

A escolha do tipo de curva está relacionada ao objeto que se deseja definir, em outras palavras, à sua aplicação. Basicamente, traçar uma curva que passe por um conjunto de pontos caracteriza um problema de interpolação, ao passo que definir uma curva que se aproxime de um determinado conjunto de pontos é um problema de aproximação, Fig. 3.1.

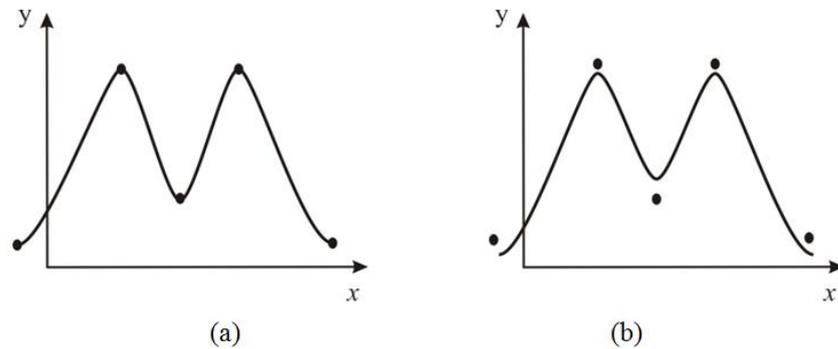


Figura 3.1 - a) Interpolação; b) Aproximação (FERREIRA, 2011)

Polinômios foram uma das primeiras escolhas para a solução de ambos os problemas. Porém os polinômios possuem alguns aspectos indesejados como instabilidade, erro de truncamento e globalidade.

A instabilidade é quando a curva possui oscilações de amplitudes elevadas nas regiões das extremidades. Isso ocorre devido à disposição dos pontos de controle associado com o aumento da ordem do polinômio (FERREIRA, 2011). Nesse caso quanto mais pontos de controle mais instável é a curva nas extremidades, o que não acontece com uma curva B-spline. No caso de interpolação ou aproximação com a B-spline, quanto mais pontos de controle melhor o formato da curva.

Uma alternativa para contornar os problemas de instabilidades numéricas dos polinômios, é trabalhar com polinômios de graus menores. E para garantir uma interpolação de um grande conjunto de pontos, pode-se calcular um polinômio para cada segmento definido por dois pontos, respeitando certo grau de continuidade da curva nesses pontos. Esse processo é denominado interpolação ou aproximação por polinomiais por partes.

Todavia, sua principal desvantagem está na sua globalidade, ou seja, ao se alterar o valor de qualquer um dos pontos, toda a curva é afetada. Isso implica no número reduzido de aplicações que um polinômio pode ter em modelagens de curvas, restrito a pequenos intervalos ou a um número baixo de pontos de controle (BARBARINI, 2007). Ao contrário dos polinômios, as curvas B-splines e NURBS possuem controle local da curva, ou seja, ao alterarmos um ponto de controle apenas um pequeno segmento da curva é afetado e o restante permanece inalterado.

Neste capítulo, serão tratadas as principais curvas utilizadas em computação gráfica como Hermite, Bézier, Splines, B-Splines e NURBS, com o objetivo de apresentar o desenvolvimento realizado para se chegar à formulação matemática da modelagem da curva

escolhida, as curvas B-Splines. Com ela é possível obter uma superfície aproximada ou interpolada, através do ajuste de curva de suas seções transversais, utilizando os pontos oriundos da digitalização de uma imagem.

3.2 Curvas paramétricas

Considere a curva da Fig. 3.2. Devido à sua forma, descrever a curva como uma função $y = f(x)$ pode ser bastante difícil (YAMAGUCHI, 1988), pois não permite o cálculo direto das coordenadas dos pontos, sendo portanto, difícil definir a equação não paramétrica de uma curva que passe por um conjunto de pontos pré-definidos. No entanto, pode-se descrever as coordenadas dos pontos da curva como funções $x = f(t)$ e $y = g(t)$. Estas duas equações são apropriadas para descrever curvas e nos leva a definição de curva paramétrica.

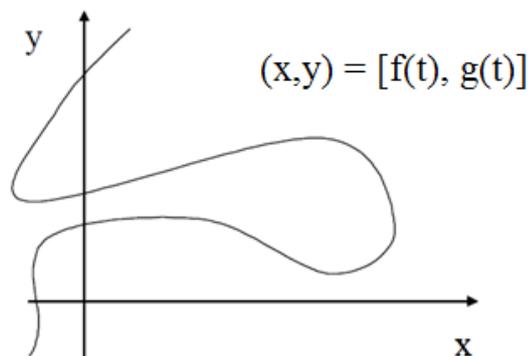


Figura 3.2 - Exemplo de curva paramétrica (BARBARINI, 2007)

Supondo que x e y sejam funções de uma terceira variável t (parâmetro), dadas pelas equações (paramétricas):

$$x=f(t) \quad y=g(t) \quad (3.1)$$

Cada valor de t determina um ponto (x,y) no plano. Conforme t varia, $(x,y)=[f(t),g(t)]$ também varia, traçando a curva, denominada de curva paramétrica.

Curvas paramétricas são a representação mais comum para curvas em computação gráfica e são muito úteis em processos de modelagem. Além da formulação paramétrica, existem ainda dois tipos de formulações para as curvas:

- Formulação explícita: função na forma $y = f(x)$
- Formulação implícita: representação do tipo $f(x, y) = 0$

No entanto, somente a formulação paramétrica nos permite resolver, simultaneamente, problemas como:

- a) Definir uma curva através de seus pontos e de suas derivadas nestes pontos, Fig. 3.3.

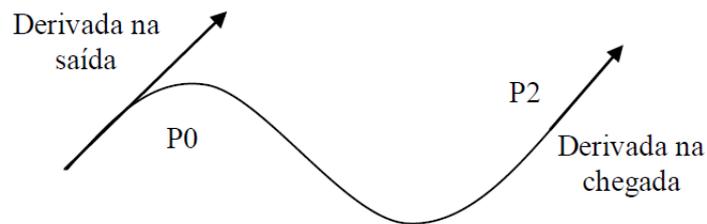


Figura 3.3 - Exemplo de curva com pontos e derivadas definidas (BARBARINI, 2007)

- b) Criar curvas fechadas ou com laços (múltiplos valores de y dado um valor de x), Fig.3.4.

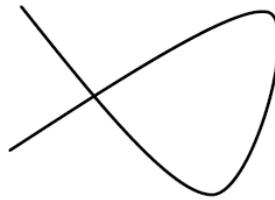


Figura 3.4 - Exemplo de curva com laço (BARBARINI, 2007)

- c) Obter uma curva suave que passe por um conjunto de pontos, Fig. 3.5

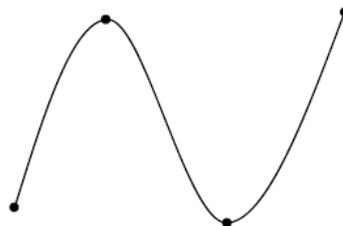


Figura 3.5: Exemplo de curva que passa por pontos dados (BARBARINI, 2007)

3.3 Curvas de Hermite

Segundo Motta (2010) e Barbarini (2007), a formulação da curva de Hermite surgiu da necessidade de solucionar problemas de definição de uma curva dados seus pontos extremos e as derivadas nestes pontos, Fig. 3.6. Geralmente são empregadas para interpolar, de forma suave, pontos, como por exemplo, em animações ou controle de câmera de filmagens. Entender sua formulação matemática (simples, porém poderosa) é o primeiro passo para o completo entendimento da família de curvas Splines e demais polinômios de ajuste de curvas.

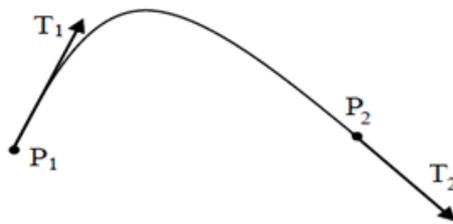


Figura 3.6: Exemplo de uma curva de Hermite (BARBARINI, 2007)

A forma polinomial de Hermite é definida por dois pontos extremos e dois vetores tangentes nestes pontos. Hermite propõe um interpolador local, controlado por 4 fatores a cada intervalo de 2 pontos: P_1 e P_2 , os pontos inicial e final, e os vetores tangentes T_1 e T_2 .

A Figura 3.7 ilustra a geometria de mistura proposta por Hermite para controlar a influência de cada um dos quatro fatores de controle de suas curvas. Essas curvas representam as funções de base de Hermite, onde P_1 e P_2 representam as funções de base associadas aos pontos P_1 e P_2 , respectivamente, e T_1 e T_2 representam as funções de base associadas às tangentes T_1 e T_2 , respectivamente.

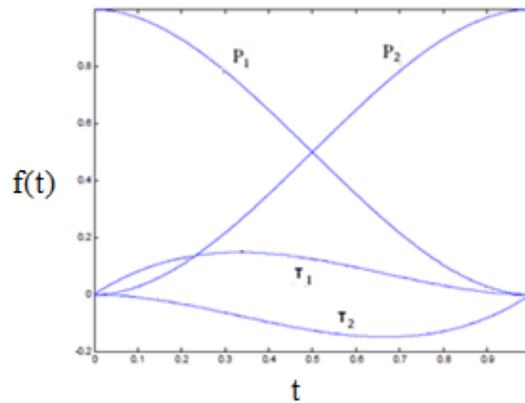


Figura 3.7 – Geometria de composição da função de Hermite

De acordo com a Fig 3.7 para um dado instante (t) qualquer, sendo o instante (t) o descritor da posição do ponto na curva, o fator de ponderação das coordenadas de um ponto para a geração da curva Hermite é resultado da soma de cada um dos fatores de controle (P_1 , P_2 , T_1 T_2) ponderados pelas curvas da geometria de Hermite. Considerando o parâmetro (t) normalizado entre 0 e 1, com 0 o instante no ponto P_1 e 1 no ponto P_2 .

As curvas de ponderação de Hermite são definidas por polinômios de terceira ordem com a forma geral:

$$\begin{aligned} x(t) &= P_x = a_x t^3 + b_x t^2 + c_x t + d_x = C_x T = G_x M_H T \\ y(t) &= P_y = a_y t^3 + b_y t^2 + c_y t + d_y = C_y T = G_y M_H T \end{aligned} \quad (3.2)$$

Onde C_x e C_y são as matrizes com os coeficientes, G_x e G_y são denominadas matrizes geométricas e M_H é a Matriz de Base de Hermite. Para solucionar a matriz geométrica de Hermite, ou seja, determinar a matriz de base é necessário determinar os fatores (a , b , c e d) para cada uma das curvas de ponderação (P_1 , P_2 , T_1 e T_2).

Far-se-á a seguir apenas o desenvolvimento para a função $x(t)$, já que para $y(t)$ o raciocínio é análogo. A determinação dos fatores (a , b , c e d) da Eq. (3.2) é feita da seguinte forma:

$$P_1: \text{ para } t=0,$$

$$P_{1x} = a_x 0^3 + b_x 0^2 + c_x 0 + d_x = [0 \ 0 \ 0 \ 1] \begin{bmatrix} a_x \\ b_x \\ c_x \\ d_x \end{bmatrix} \quad (3.3)$$

P_2 : para $t=1$,

$$P_{2x} = a_x 1^3 + b_x 1^2 + c_x 1 + d_x = [1 \ 1 \ 1 \ 1] \begin{bmatrix} a_x \\ b_x \\ c_x \\ d_x \end{bmatrix} \quad (3.4)$$

As tangentes em P_1 e P_2 podem ser obtidas através da derivada da função em cada um dos pontos:

$$\frac{d}{dt} x(t) = 3a_x t^2 + 2b_x t + c_x \quad (3.5)$$

T_1 : para $t=0$,

$$T_{1x} = a_x 3 \cdot 0^2 + b_x 2 \cdot 0 + c_x = [0 \ 0 \ 1 \ 0] \begin{bmatrix} a_x \\ b_x \\ c_x \\ d_x \end{bmatrix} \quad (3.6)$$

T_2 : para $t=1$,

$$T_{2x} = a_x 3 \cdot 1^2 + b_x 2 \cdot 1 + c_x = [3 \ 2 \ 1 \ 0] \begin{bmatrix} a_x \\ b_x \\ c_x \\ d_x \end{bmatrix} \quad (3.7)$$

Das Eqs. (3.3), (3.4), (3.6) e (3.7) obtém-se a seguinte relação na forma matricial:

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} a_x & a_y \\ b_x & b_y \\ c_x & c_y \\ d_x & d_y \end{bmatrix} = \begin{bmatrix} P_{1x} & P_{1y} \\ P_{2x} & P_{2y} \\ T_{1x} & T_{1y} \\ T_{2x} & T_{2y} \end{bmatrix} \quad (3.8)$$

Considerando a Eq. (3.8) como $A = B^{-1}C$ é possível simplificar para $BA = C$ após inserir B multiplicando os dois lados $BA = BB^{-1}C$, obtendo a Eq. (3.9).

$$\begin{bmatrix} a_x & a_y \\ b_x & b_y \\ c_x & c_y \\ d_x & d_y \end{bmatrix} = \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 1 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_{1x} & P_{1y} \\ P_{2x} & P_{2y} \\ T_{1x} & T_{1y} \\ T_{2x} & T_{2y} \end{bmatrix} \quad (3.9)$$

Como $C = M_H G$, na Eq. (3.9) já estão descritas as duas matrizes, M_H e G , para a curva de Hermite. Substituindo na forma geral $P(t) = T M_H G$, se obtém a Eq. (3.10), também podendo ser descrita como na Eq. (3.11).

$$P(t) = [t^3 \ t^2 \ t \ 1] \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 1 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_{1x} & P_{1y} \\ P_{2x} & P_{2y} \\ T_{1x} & T_{1y} \\ T_{2x} & T_{2y} \end{bmatrix} = [x(t) \ y(t)] \quad (3.10)$$

$$\begin{aligned} x(t) &= P_{1x}(2t^3 - 3t^2 + 1) + P_{2x}(-2t^3 + 3t^2) + T_{1x}(t^3 - 2t^2 + t) + T_{2x}(t^3 - t^2) \\ y(t) &= P_{1y}(2t^3 - 3t^2 + 1) + P_{2y}(-2t^3 + 3t^2) + T_{1y}(t^3 - 2t^2 + t) + T_{2y}(t^3 - t^2) \end{aligned} \quad (3.11)$$

Exemplos de diferentes resultados obtidos alterando apenas o vetor T_1 são apresentados na Fig. 3.8. Lembrando que, para as curvas Hermite, a ordem da continuidade está relacionada com a definição dos vetores.

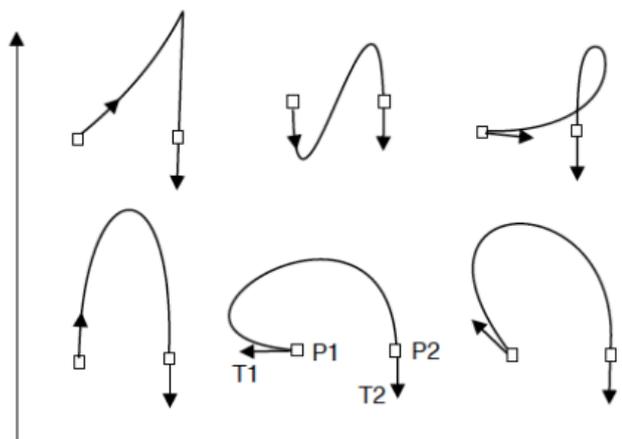


Figura 3.8 - Exemplo de diferentes curvas formadas alterando-se apenas o vetor T_1 (MOTTA, 2010)

3.4 Curvas de Bézier

A curva de Bézier foi desenvolvida por Pierre Bézier e ganhou grande importância quando estudada por Paul de Casteljaou no fim da década de 1950, com o fim de criar um método eficaz para a modelagem de carros. Baseado nos princípios de Hermite, Bézier acrescentou dois pontos de controle às constantes de Hermite (P_1 e P_2), determinando os dois vetores tangentes nos pontos inicial e final. A curva de Bézier é uma forma aproximada de curva polinomial, especialmente concebida para ser controlada de forma conveniente.

As formas mais importantes das curvas de Bézier são as quadráticas e cúbicas. Curvas de grau maior implicam em um custo mais elevado de cálculo e processamento. O método atualmente mais utilizado para avaliar as curvas de Bézier é o algoritmo de De Casteljaou (BARBARINI, 2007).

A sua formulação é relevante para se entender os conceitos de curvas B-splines que serão vistas mais adiante.

Uma curva de Bézier de grau n é definida por:

$$C(u) = \sum_{i=0}^n B_{i,n}(u)P_i \quad 0 \leq u \leq 1 \quad (3.12)$$

As funções de base, $B_{i,n}(u)$, são os polinômios de Bernstein de grau n , dados por:

$$B_{i,n}(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i} \quad (3.13)$$

E os coeficientes geométricos, P_i , são denominados de *pontos de controle* e u corresponde ao parâmetro que varia de 0 a 1.

Através da Eq. (3.12) é possível desenvolver as curvas de Bézier variando o valor de n . Para uma melhor compreensão são apresentados exemplos extraídos de (PIEGL; TILLER, 1997):

- Para $n=1$: Da Eq. (3.13) tem-se $B_{0,1}(u) = 1 - u$ e $B_{1,1}(u) = u$, e da Eq. (3.12) tem-se:

$$C(u) = (1-u)P_0 + uP_1 \quad (3.14)$$

que resulta em um segmento de linha reta de P_0 a P_1 , Fig. 3.9.

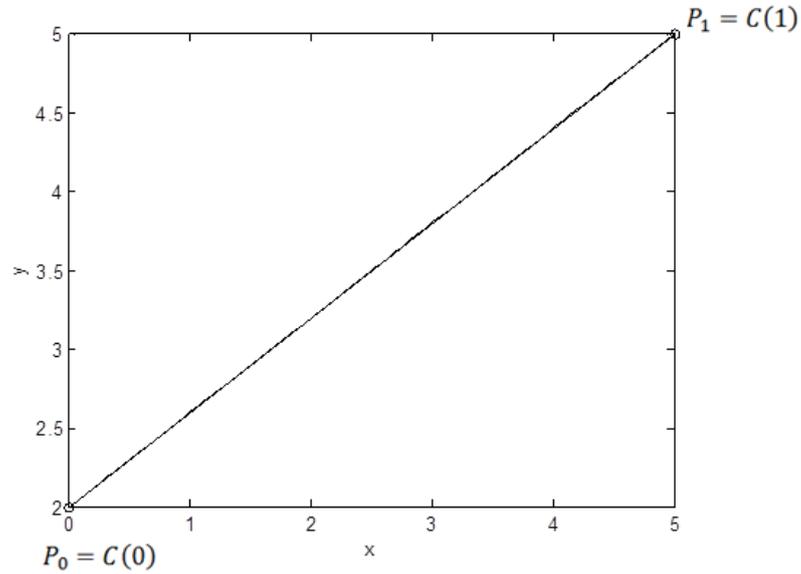


Figura 3.9 - Curva de Bézier de primeiro grau

- Para $n=2$: Das Eqs. (3.12) e (3.13) tem-se:

$$C(u) = (1 - u)^2 P_0 + 2u(1 - u)P_1 + u^2 P_2 \quad (3.15)$$

que resulta um arco parabólico de P_0 a P_2 , Fig. 3.10.

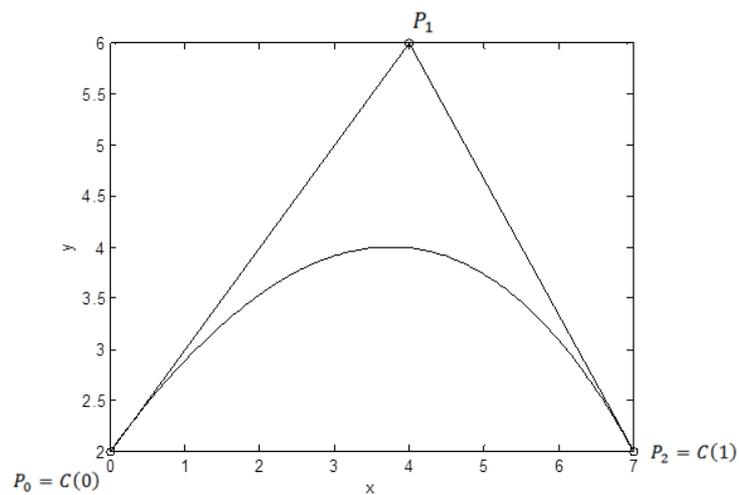


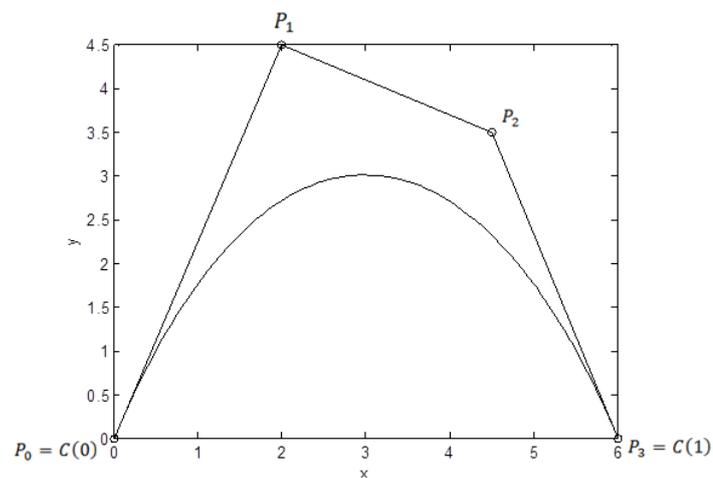
Figura 3.10 - Curva de Bézier de segundo grau

Note que: O polígono formado por (P_0, P_1, P_2) é denominado de *polígono de controle*; $P_0 = C(0)$ e $P_2 = C(1)$; as direções das tangentes da curva nos pontos finais são paralelas a $P_1 - P_0$ e $P_2 - P_1$; a curva de Bézier está contida no triângulo formado por $P_0P_1P_2$.

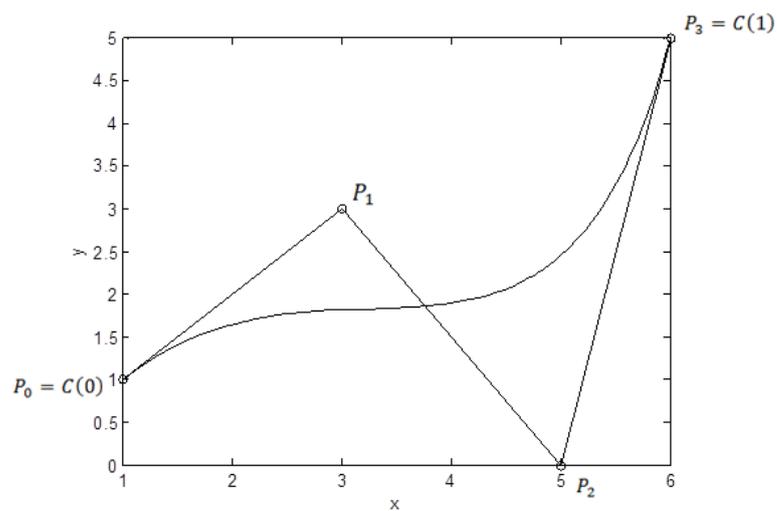
- Para $n=3$: Tem-se:

$$C(u) = (1 - u)^3 P_0 + 3u(1 - u)^2 P_1 + 3u^2(1 - u) P_2 + u^3 P_3 \quad (3.16)$$

Exemplos de curvas cúbicas de Bézier são mostradas nas Fig. 3.11a e 3.11b.



(a)



(b)

Figura 3.11 - (a) e (b) Exemplos de curvas de Bézier cúbicas

Note que: Os polígonos formados por (P_0, P_1, P_2, P_3) , Figs. 3.11a e 3.11b, são denominados de *polígonos de controle*, os quais aproximam as formas das curvas, ou seja, as formas das curvas são definidas a partir das formas dos polígonos de controle; $P_0 = C(0)$ e $P_3 = C(1)$; as direções das tangentes nos pontos inicial e final são paralelas a $P_1 - P_0$ e $P_3 - P_2$, respectivamente.

Em todos os casos as curvas geradas estão contidas no fecho convexo (polígono convexo que contém todos os pontos) definidos pelos pontos de controle. Isso define uma das suas propriedades que é a propriedade do fecho convexo, que garante que a curva sempre estará no interior do fecho convexo. Uma outra característica das curvas de Bézier é que a curva interpola o primeiro e último ponto do polígono de controle, visualizado nas Figs. 3.9, 3.10, 3.11a e 3.11b.

Seja $n=2$ e

$$C(u) = \sum_{i=0}^2 B_{i,2}(u)P_i$$

então,

$$\begin{aligned} C(u) &= (1-u)^2P_0 + 2u(1-u)P_1 + u^2P_2 \\ &= (1-u)\underbrace{((1-u)P_0 + uP_1)}_{\text{linear}} + u\underbrace{((1-u)P_1 + uP_2)}_{\text{linear}} \end{aligned}$$

Portanto, a curva de Bézier de segundo grau é obtida como uma interpolação linear de duas curvas de Bézier de primeiro grau. Em particular, qualquer ponto de $C(u)$ é obtido por três interpolações lineares.

Assumindo um valor fixo $u = u_0$ e sabendo que $P_{1,0} = (1-u_0)P_0 + u_0P_1$, $P_{1,1} = (1-u_0)P_1 + u_0P_2$, e $P_{2,0} = (1-u_0)P_{1,0} + u_0P_{1,1}$, segue-se que $C(u) = P_{2,0}$. Essa situação é descrita na Fig. 3.12, e o caso cúbico é mostrado na Fig. 3.13, para o parâmetro $u_0 = 2/5$.

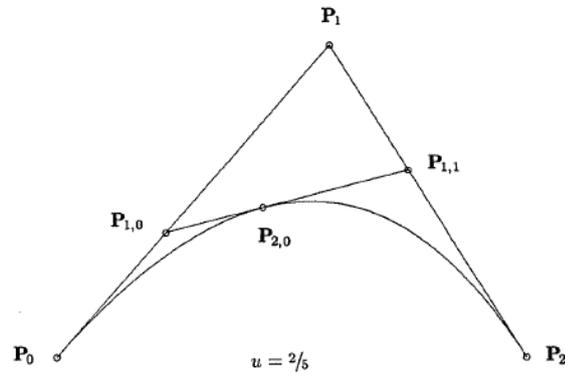


Figura 3.12 - Obtendo um ponto na curva quadrática de Bézier por uma interpolação linear em $u_0=2/5$ (PIEGL; TILLER, 1997)

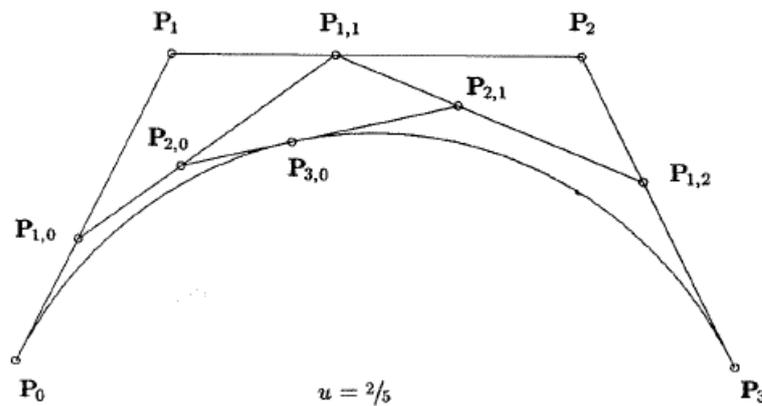


Figura 3.13 - Interpolação linear em $u_0=2/5$ de uma curva de Bézier cúbica (PIEGL; TILLER, 1997)

Denotando uma curva Bézier geral de grau n por $C_n(P_0, \dots, P_n)$ tem-se

$$C_n(P_0, \dots, P_n) = (1 - u)C_{n-1}(P_0, \dots, P_{n-1}) + uC_{n-1}(P_1, \dots, P_n) \quad (3.17)$$

Fixando $u = u_0$ e denotando P_i por $P_{0,i}$, a Eq. (3.17) se torna um algoritmo recursivo para calcular o ponto $C(u_0) = P_{n,0}$ de um curva de Bézier de grau n , isto é:

$$P_{k,i}(u_0) = (1 - u_0)P_{k-1,i}(u_0) + u_0P_{k-1,i+1}(u_0) \quad \text{para} \quad \begin{cases} k = 1, \dots, n \\ i = 0, \dots, n - k \end{cases} \quad (3.18)$$

A Eq. (3.18) é denominada *Algoritmo de Casteljau*. A seguir é descrito o método para construir uma curva pelo método desenvolvido por Casteljau. No exemplo foi considerado o parâmetro $u_0 = 0,5$, que significa que para a construção da curva, cada segmento entre dois pontos de controle consecutivos é dividido ao meio. Inicialmente, sejam três pontos de controle P_0, P_1 e P_2 , onde P_0 e P_2 definem os extremos e P_1 atua como um parâmetro de curvatura ou “ponto de controle”, Fig. 3.14.

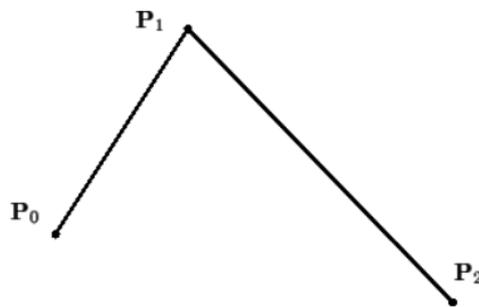


Figura 3.14 - Situação inicial (WANGNHEIM; WAGNER, 2013)

Para definir a curva é necessário seguir os seguintes passos: Seja $P_1^{(1)}$ o ponto médio do segmento $\overline{P_0P_1}$, pois $u_0 = 0,5$, Fig. 3.15.

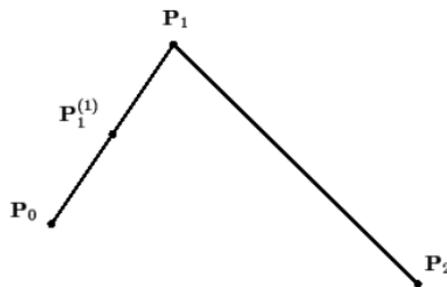


Figura 3. 15 - $P_1^{(1)}$ o ponto médio do segmento $\overline{P_0P_1}$ (WANGNHEIM; WAGNER, 2013)

A seguir é obtido $P_2^{(1)}$ que é o ponto médio do segmento $\overline{P_1P_2}$. Dessa forma cada um dos dois segmentos é dividido ao meio ($u_0 = 0,5$), Fig. 3.16.

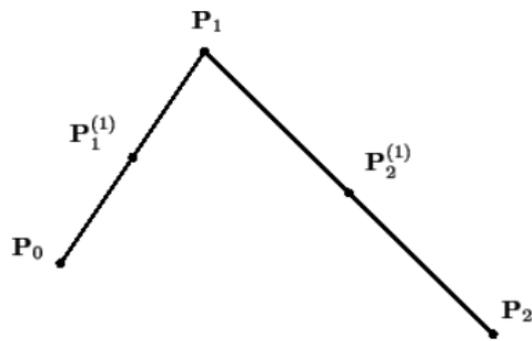


Figura 3.16 - $P_2^{(1)}$ o ponto médio do segmento $\overline{P_1P_2}$ (WANGNHEIM; WAGNER, 2013)

Seja agora o ponto $P_2^{(2)}$ o ponto médio do segmento $\overline{P_1^{(1)}P_2^{(1)}}$. Assim cria-se um novo ponto, que, pelo algoritmo de Casteljau, estará sobre a curva, Fig. 3.17.

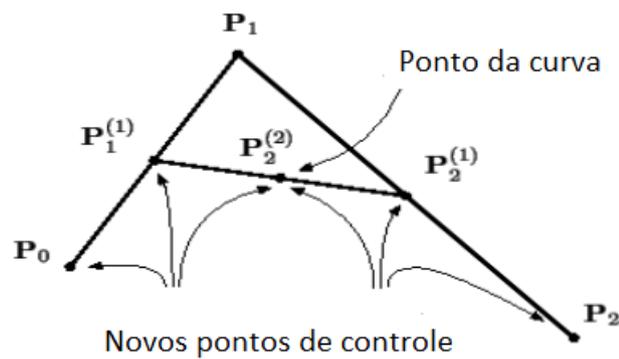


Figura 3.17 - $P_2^{(2)}$ o ponto médio do segmento $\overline{P_1^{(1)}P_2^{(1)}}$ (WANGNHEIM; WAGNER, 2013)

Pode-se utilizar dois novos conjuntos de pontos como novos pontos de controle para continuar subdividindo a “curva” de forma hierárquica: $[P_0, P_1^{(1)} \text{ e } P_2^{(2)}]$ e $[P_2^{(2)}, P_2^{(1)} \text{ e } P_2]$. Recursivamente recria-se o problema com mais três novos pontos, fazendo primeiro $[P_0, P_1^{(1)} \text{ e } P_2^{(2)}]$ serem $P_0, P_1 \text{ e } P_2$ e reaplicando o método, Fig. 3.18.

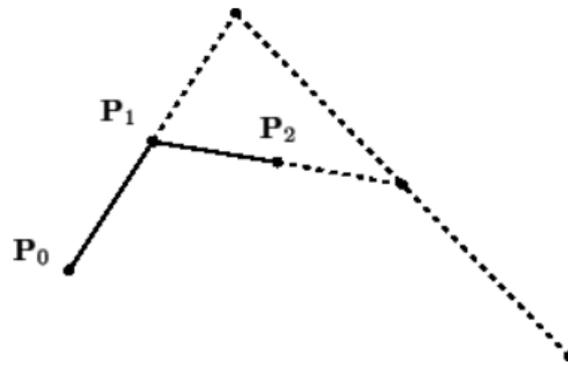


Figura 3.18- Cria-se uma nova sub-curva à esquerda, a qual começa a subdividir (WANGNHEIM; WAGNER, 2013)

Seja o novo $P_1^{(1)}$ o ponto médio do segmento rebatizado de $\overline{P_0P_1}$, Fig. 3.19.

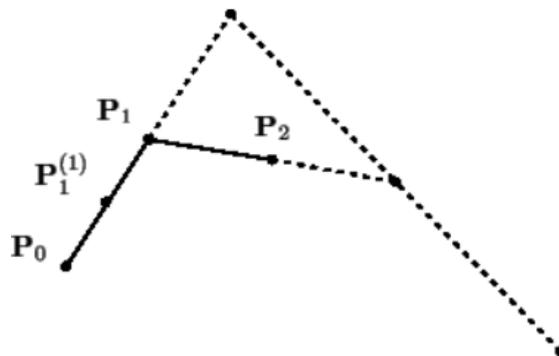


Figura 3.19 - Subdividir à esquerda (WANGNHEIM; WAGNER, 2013)

Seja o novo $P_2^{(1)}$ o ponto médio do segmento rebatizado de $\overline{P_1P_2}$, Fig. 3.20.

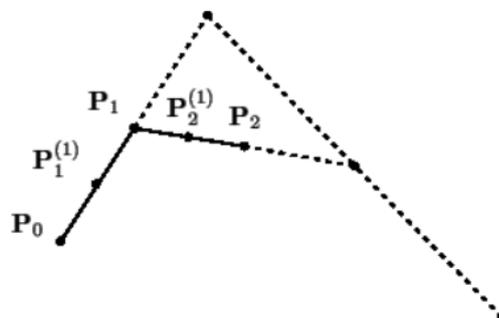


Figura 3.20 – Subdivisão do novo $\overline{P_1P_2}$ (WANGNHEIM; WAGNER, 2013)

Seja agora um novo ponto $P_2^{(2)}$ o ponto médio do segmento $\overline{P_1^{(1)}P_2^{(1)}}$ dessa subcurva esquerda, Fig. 3.21.

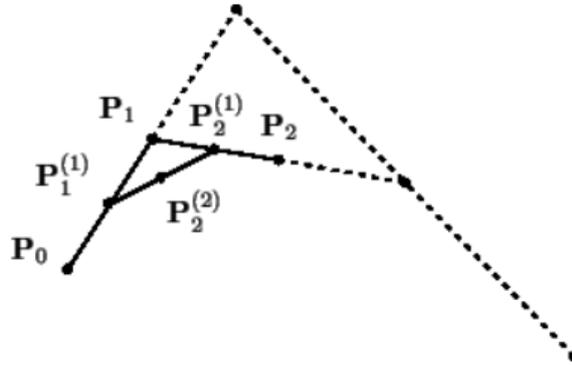


Figura 3.21 - $P_2^{(2)}$ o ponto médio do segmento $\overline{P_1^{(1)}P_2^{(1)}}$ (WANGNHEIM; WAGNER, 2013)

Esta sequência demonstra o processo para a subárvore esquerda. O processo é o mesmo para o outro lado, por isso é demonstrada a seguir apenas a sequência de operações. Quando for decidido terminar, pode-se plotar a curva desejada, Fig. 3.22.

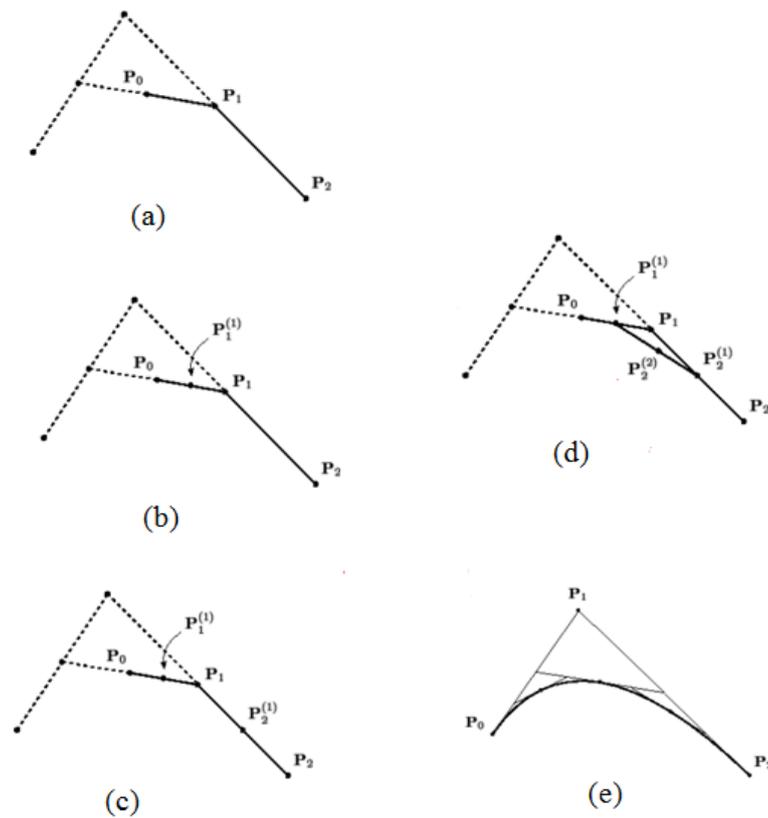


Figura 3.22 - Sequência do algoritmo (WANGNHEIM; WAGNER, 2013)

Com o mesmo método é possível definir outras curvas com uma maior quantidade de pontos de controle.

3.5 Spline

O termo Spline é designado na matemática para definir curvas especiais formadas por segmentos de polinômios (BARBARINI, 2007).

O princípio básico da spline é a interpolação de Hermite por partes. Enquanto na Interpolação de Hermite por partes a derivada em cada ponto é imposta, na spline é estabelecida somente a condição de continuidade. Isso significa que a cada dois segmentos consecutivos da curva, as derivadas de ordem k do final de um segmento e do início do segmento seguinte são iguais.

Considerando um conjunto de $n+1$ pontos, é possível definir uma spline que os interpola. Portanto, para obter a spline deve-se encontrar as polinomiais, $S_i(x)$ ($i = 0, \dots, n-1$) do p -ésimo grau conforme a Eq. (3.19), para os n segmentos como mostrado na Fig. 3.23. Dado

um polinômio do p -ésimo grau, sabe-se que ele possui $(p+1)$ coeficientes, dessa forma é preciso resolver um sistema de equações para encontrar $n(p+1)$ coeficientes da spline.

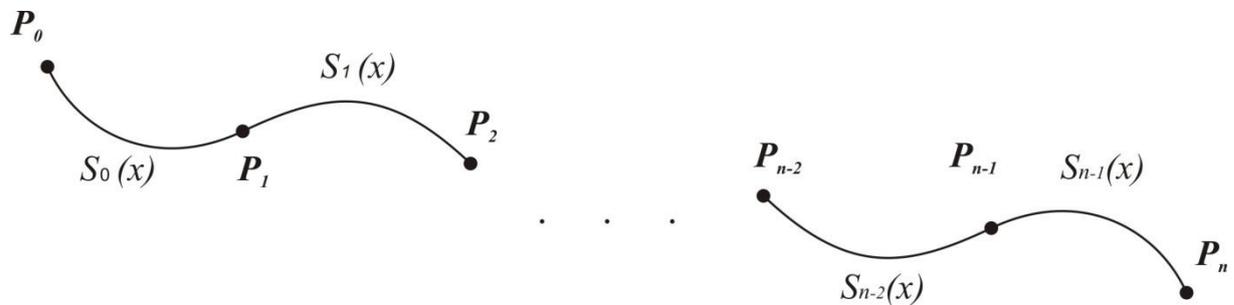


Figura 3.23 - Interpolação por spline (FERREIRA, 2011)

$$S_i(x) = a_{p,i}x^p + a_{p-1,i}x^{p-1} + \dots + a_{1,i}x + a_{0,i} \quad (3.19)$$

Assim, as $n(p+1)$ equações devem ser formuladas para que os coeficientes dos polinômios sejam encontrados. Essas equações são obtidas pelas condições de contorno dadas por:

- Continuidade C^0 na união dos segmentos $\rightarrow 2(n-1)$ equações

C^k representa a continuidade paramétrica. Para que uma curva possua continuidade paramétrica, as curvas têm no ponto de junção todas as derivadas iguais até à ordem k .

Na continuidade C^0 o último ponto da primeira curva é igual ao primeiro ponto da segunda.

A continuidade paramétrica de uma curva pode ser definida até a $(p-1)$ -ésima ordem C^k ($k=1, \dots, p-1$).

$$\begin{aligned} S_0(x_1) &= y_1 \\ S_1(x_1) &= y_1 \\ &\vdots \\ &\vdots \\ &\vdots \\ S_{n-2}(x_{n-1}) &= y_{n-1} \\ S_{n-1}(x_{n-1}) &= y_{n-1} \end{aligned} \quad (3.20)$$

- Continuidade C^l à C^{p-l} na união dos segmentos $\rightarrow (p-1)(n-1)$ equações

Na união dos segmentos da spline define-se a continuidade desejada. Para isso as derivadas de um segmento devem ser iguais as dos seus segmentos vizinhos até a p -ésima ordem.

Condição de continuidade na derivada primeira

$$S'_0(x_1) = S'_1(x_1)$$

$$S'_1(x_2) = S'_2(x_2)$$

·
·
·

$$S'_{n-2}(x_{n-1}) = S'_{n-1}(x_{n-1}) \quad (3.21)$$

Condição de continuidade na derivada segunda

$$S''_0(x_1) = S''_1(x_1)$$

$$S''_1(x_2) = S''_2(x_2)$$

·
·
·

$$S''_{n-2}(x_{n-1}) = S''_{n-1}(x_{n-1}) \quad (3.22)$$

O processo é realizado sucessivas vezes até a condição da $(p-1)$ -ésima derivada,

$$S^{p-1}_0(x_1) = S^{p-1}_1(x_1)$$

$$S^{p-1}_1(x_2) = S^{p-1}_2(x_2)$$

·
·
·

$$S^{p-1}_{n-2}(x_{n-1}) = S^{p-1}_{n-1}(x_{n-1}) \quad (3.23)$$

- Valor da curva nos pontos extremos \rightarrow 2 equações

$$S_0(x_0) = y_0$$

$$S_{n-1}(x_n) = y_n \quad (3.24)$$

- Derivadas nos pontos extremos \rightarrow $(p-1)$ equações

Neste caso são impostas as condições de contorno necessárias para completar as equações do sistema. Por exemplo, se $p = 3$, pode-se calcular a spline com a derivada segunda nula nos extremos, neste caso a curva é conhecida como spline cúbica natural.

$$S''_0(x_0) = 0$$

$$S''_{n-1}(x_n) = 0 \quad (3.25)$$

Com essas condições de contorno é possível construir um sistema com $n(p+1)$ equações, cuja solução são os coeficientes dos polinômios da spline.

Apesar de ser possível definir a continuidade desejada de uma spline, a alteração de um ponto deste tipo de curva pode modificar o perfil de toda a curva, como mostrado na Fig. 3.24, para a alteração do ponto P .

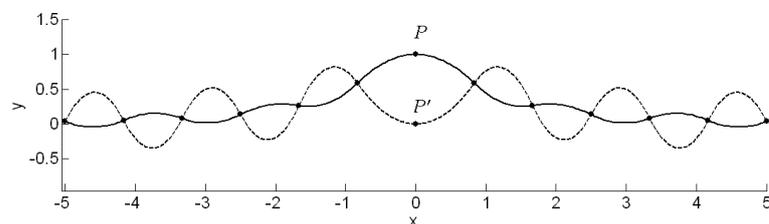


Figura 3.24 - Interpolação com spline quadrática. Interpolação com os pontos originais (linha contínua), interpolação com a mudança do sétimo ponto P (linha tracejada) (FERREIRA, 2011)

3.6 B-spline

Autores como (PIEGL; TILLER, 1997), (DE BOOR, 2000) e (ROGERS, 2001) definem a B-spline como uma versão da spline que implementa o controle local da curva, de forma que a alteração de um ponto de controle modifica a curva somente na região dos pontos vizinhos mais próximos em função da ordem de continuidade, como mostrado na Fig. 3.25.

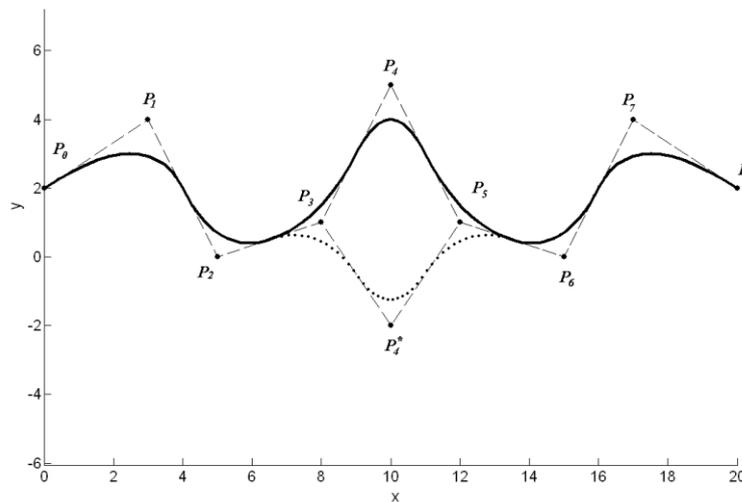


Figura 3.25 - B-spline quadrática. A linha cheia mostra a curva original, a linha pontilhada representa a curva com a mudança do ponto de controle P_4 (FERREIRA, 2011)

Para obter uma curva B-spline, precisa-se de um vetor de nós, das funções de base e um conjunto de pontos de controle.

De acordo com Piegl e Tiller (1997), uma função de base é denotada por $N_{i,p}(u)$, que significa que é a i -ésima função de base B-spline de grau p (ordem $p+1$). Existem várias formas de definir uma função de base. No entanto, a que tem sido mais utilizada, em função de sua facilidade de implementação computacional, é a fórmula recursiva de Cox DeBoor, e que será também utilizada neste trabalho. Para defini-la é necessário um vetor de números reais não decrescente, denominado de vetor nó, e definido por $U = \{u_0, \dots, u_m\}$, i.e., $u_i \leq u_{i+1}$, $i = 0, \dots, m - 1$, onde os u_i são denominados nós (*knots*).

$N_{i,p}(u)$ é definida como:

$$N_{i,0}(u) = \begin{cases} 1 & \text{para } u_i \leq u < u_{i+1} \\ 0 & \text{caso contrário.} \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (3.26)$$

Note que:

- $N_{i,0}(u)$ é uma função pulso, é sempre igual a zero exceto no intervalo aberto $u \in [u_i, u_{i+1})$;
- Para $p > 0$, $N_{i,p}(u)$ é uma combinação linear de duas funções de base de grau $p-1$;
- O cálculo de um conjunto de funções de base requer a especificação de um vetor de nós U , e o grau p ;
- A Equação (3.26) pode levar a quocientes do tipo $0/0$. Esses quocientes são definidos como zero;
- $N_{i,p}(u)$ são polinomiais por partes, definidos em todo domínio real. Geralmente somente o intervalo $[u_0, u_m]$ é de interesse;
- O intervalo aberto, $[u_i, u_{i+1})$, é denominado de *i-ésimo* intervalo do nó. Ele pode ter comprimento nulo, desde que os nós não necessitem ser distintos;
- O cálculo das funções de grau p gera um diagrama triangular truncado na forma:

$$\begin{array}{cccc}
 & & N_{0,0} & \\
 & & & N_{0,1} \\
 & N_{1,0} & & N_{0,2} \\
 & & N_{1,1} & & N_{0,3} \\
 & N_{2,0} & & N_{1,2} & \\
 & & N_{2,1} & & N_{1,3} \\
 & N_{3,0} & & N_{2,2} & \vdots \\
 & & N_{3,1} & \vdots & \\
 & N_{4,0} & \vdots & & \\
 & \vdots & & &
 \end{array}$$

O procedimento de cálculo das funções de base é apresentado com um exemplo extraído de (PIEGL; TILLER, 1997):

Considerando o vetor nó dado por $U = \{u_0 = 0, u_1 = 0, u_2 = 0, u_3 = 1, u_4 = 2, u_5 = 3, u_6 = 4, u_7 = 4, u_8 = 5, u_9 = 5, u_{10} = 5\}$ e $p = 2$. As funções de base de zero, primeiro e segundo graus são calculadas, sendo que as funções de base não nulas são mostradas nas Figs. 3.26,

3.27 e 3.28, respectivamente. Nas figuras, no eixo das ordenadas, cada traço representa a quantidade de vezes em que o nó aparece no vetor nó.

$$N_{0,0} = N_{1,0} = 0 \quad \text{para } -\infty < u < \infty$$

$$N_{2,0} = \begin{cases} 1 & 0 \leq u < 1 \\ 0 & \text{outros casos} \end{cases}$$

$$N_{3,0} = \begin{cases} 1 & 1 \leq u < 2 \\ 0 & \text{outros casos} \end{cases}$$

$$N_{4,0} = \begin{cases} 1 & 2 \leq u < 3 \\ 0 & \text{outros casos} \end{cases}$$

$$N_{5,0} = \begin{cases} 1 & 3 \leq u < 4 \\ 0 & \text{outros casos} \end{cases}$$

$$N_{6,0} = 0 \quad \text{para } -\infty < u < \infty$$

$$N_{7,0} = \begin{cases} 1 & 4 \leq u < 5 \\ 0 & \text{outros casos} \end{cases}$$

$$N_{8,0} = N_{9,0} = 0 \quad \text{para } -\infty < u < \infty$$

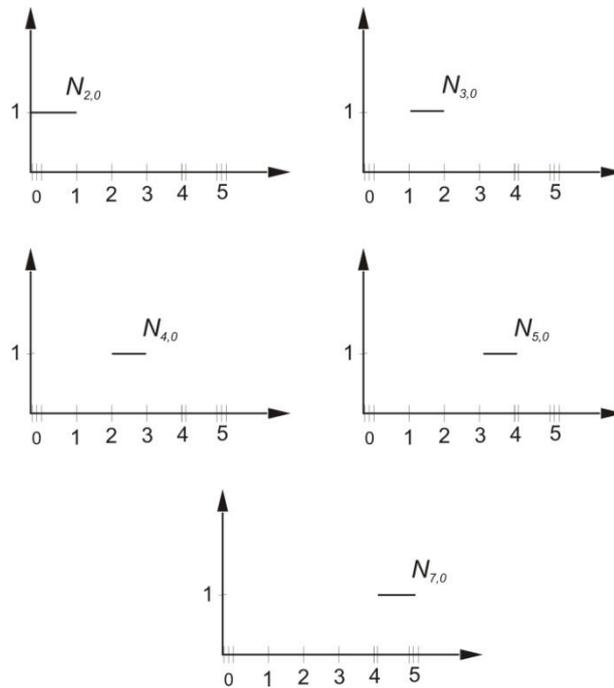


Figura 3.26 - Funções base de grau zero, $U = \{0,0,0,1,2,3,4,4,5,5,5\}$, (PIEGL; TILLER, 1997)

$$N_{0,1} = \frac{u-0}{0-0} N_{0,0} + \frac{0-u}{0-0} N_{1,0} = 0 \quad -\infty < u < \infty$$

$$N_{1,1} = \frac{u-0}{0-0} N_{1,0} + \frac{1-u}{1-0} N_{2,0} = \begin{cases} 1-u & 0 \leq u < 1 \\ 0 & \text{outros casos} \end{cases}$$

$$N_{2,1} = \frac{u-0}{1-0} N_{2,0} + \frac{2-u}{2-1} N_{3,0} = \begin{cases} u & 0 \leq u < 1 \\ 2-u & 1 \leq u < 2 \\ 0 & \text{outros casos} \end{cases}$$

$$N_{3,1} = \frac{u-1}{2-1} N_{3,0} + \frac{3-u}{3-2} N_{4,0} = \begin{cases} u-1 & 1 \leq u < 2 \\ 3-u & 2 \leq u < 3 \\ 0 & \text{outros casos} \end{cases}$$

$$N_{4,1} = \frac{u-2}{3-2} N_{4,0} + \frac{4-u}{4-3} N_{5,0} = \begin{cases} u-2 & 2 \leq u < 3 \\ 4-u & 3 \leq u < 4 \\ 0 & \text{outros casos} \end{cases}$$

$$N_{5,1} = \frac{u-3}{4-3} N_{5,0} + \frac{4-u}{4-4} N_{6,0} = \begin{cases} u-3 & 3 \leq u < 4 \\ 0 & \text{outros casos} \end{cases}$$

$$N_{6,1} = \frac{u-4}{4-4} N_{6,0} + \frac{5-u}{5-4} N_{7,0} = \begin{cases} 5-u & 4 \leq u < 5 \\ 0 & \text{outros casos} \end{cases}$$

$$N_{7,1} = \frac{u-4}{5-4} N_{4,0} + \frac{5-u}{5-5} N_{8,0} = \begin{cases} u-4 & 4 \leq u < 5 \\ 0 & \text{outros casos} \end{cases}$$

$$N_{8,1} = \frac{u-5}{5-5} N_{8,0} + \frac{5-u}{5-5} N_{9,0} = 0 \quad -\infty < u < \infty$$

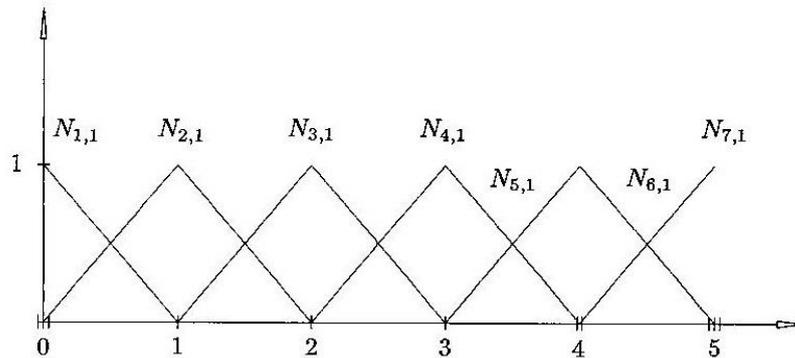


Figura 3.27 - Funções base do primeiro grau, $U = \{0,0,0,1,2,3,4,4,5,5,5\}$, (PIEGL; TILLER, 1997)

Todos os $N_{i,2}$ seguintes são iguais a zero, exceto nos intervalos especificados, tal que:

$$N_{0,2} = \frac{u-0}{0-0} N_{0,1} + \frac{1-u}{1-0} N_{1,1} = (1-u)^2 \quad 0 \leq u < 1$$

$$N_{1,2} = \frac{u-0}{1-0} N_{1,1} + \frac{2-u}{2-0} N_{2,1} = \begin{cases} 2u - \frac{3}{2}u^2 & 0 \leq u < 1 \\ \frac{1}{2}(2-u)^2 & 1 \leq u < 2 \end{cases}$$

$$N_{2,2} = \frac{u-0}{2-0} N_{2,1} + \frac{3-u}{3-1} N_{3,1} = \begin{cases} \frac{1}{2}u^2 & 0 \leq u < 1 \\ -\frac{3}{2} + 3u - u^2 & 1 \leq u < 2 \\ \frac{1}{2}(3-u)^2 & 2 \leq u < 3 \end{cases}$$

$$N_{3,2} = \frac{u-1}{3-1} N_{3,1} + \frac{4-u}{4-2} N_{4,1} = \begin{cases} \frac{1}{2}(u-1)^2 & 1 \leq u < 2 \\ -\frac{11}{2} + 5u - u^2 & 2 \leq u < 3 \\ \frac{1}{2}(4-u)^2 & 3 \leq u < 4 \end{cases}$$

$$N_{4,2} = \frac{u-2}{4-1} N_{4,1} + \frac{4-u}{4-3} N_{5,1} = \begin{cases} 1/2 (u-2)^2 & 2 \leq u < 3 \\ -16 + 10u - 3/2 u^2 & 3 \leq u < 4 \end{cases}$$

$$N_{5,2} = \frac{u-3}{4-3} N_{5,1} + \frac{5-u}{5-4} N_{6,1} = \begin{cases} (u-3)^2 & 2 \leq u < 3 \\ (5-u)^2 & 3 \leq u < 4 \end{cases}$$

$$N_{6,2} = \frac{u-4}{5-4} N_{6,1} + \frac{5-u}{5-4} N_{7,1} = 2(u-4)(5-u) \quad 4 \leq u < 5$$

$$N_{7,2} = \frac{u-4}{5-4} N_{7,1} + \frac{5-u}{5-5} N_{8,1} = (u-4)^2 \quad 4 \leq u < 5$$

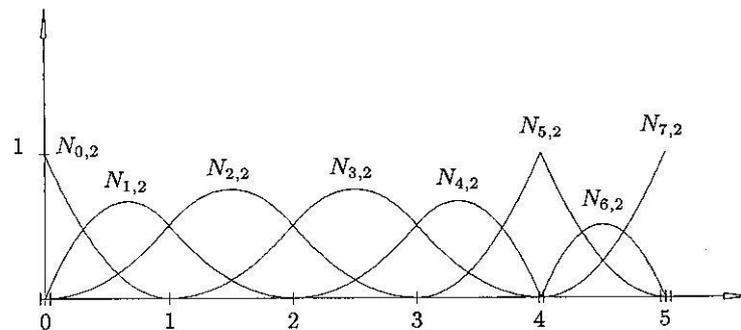


Figura 3.28 – Funções de base do segundo grau, $U = \{0,0,0,1,2,3,4,4,5,5,5\}$ (PIEGL; TILLER, 1997)

3.6.1 Propriedades importantes das funções de base B-spline

As propriedades apresentadas a seguir determinam várias características geométricas desejáveis das curvas B-splines, considerando o grau p e o vetor nó $U = \{u_0, \dots, u_m\}$.

Propriedade 1- Propriedade de suporte local: $N_{i,p}(u) = 0$ se u é fora do intervalo $[u_i, u_{i+p+1})$. Pode-se observar que $N_{1,3}$ é a combinação de $N_{1,0}$, $N_{2,0}$, $N_{3,0}$ e $N_{4,0}$. Então, $N_{1,3}$ é somente não nula para u pertencente à $[u_1, u_5)$. Esta propriedade é ilustrada pelo esquema triangular, apresentado na Fig 3.29.

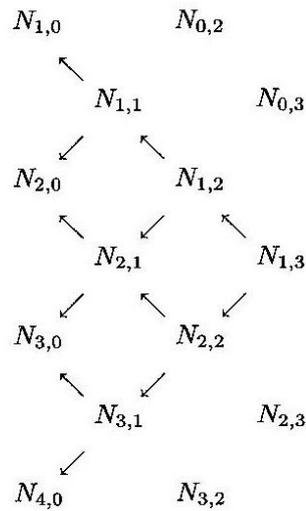


Figura 3.29 – Esquema da propriedade de suporte local (PIEGL; TILLER, 1997)

Propriedade 2- Em um dado intervalo de nó, $[u_j, u_{j+1})$, no máximo $p+1$ das $N_{i,p}(u)$ são não nulas, mais especificamente as funções $N_{j-p,p}, \dots, N_{j,p}$. Por exemplo, no intervalo $[u_3, u_4)$ a única função não nula de grau zero é $N_{3,0}$. Então, as únicas funções cúbicas não nulas no intervalo $[u_3, u_4)$ são $N_{0,3}, \dots, N_{3,3}$. Esta propriedade é ilustrada na Fig. 3.30:

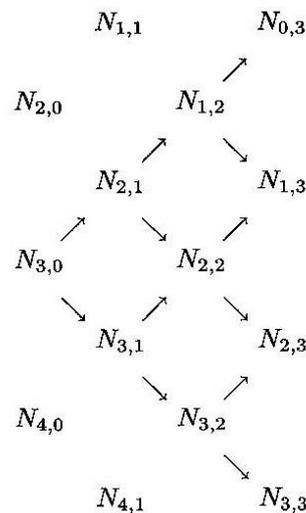


Figura 3.30 – Esquema propriedade 2 (PIEGL; TILLER, 1997)

Pelo cálculo das funções de base ser de forma recursiva, as setas indicam que as funções cúbicas não nulas dependem das funções de base anteriores a ela.

Propriedade 3- Propriedade da Não Negatividade: $N_{i,p}(u) \geq 0$ para todo i, p e u . Isto pode ser provado pela indução em p . Isso é claramente verdade para $p=0$; assume que também seja

verdade para $p-1$, $p \geq 0$, com i e u arbitrários. Por definição, a função de base não nula é dada pela Eq. (3.26), ou seja:

$$N_{i,p}(u) = \frac{u-u_i}{u_{i+p}-u_i} N_{i,p-1}(u) + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}} N_{i+1,p-1}(u) \quad (3.27)$$

Pela Propriedade 1, $N_{i,p-1}(u) = 0$ se $u \notin [u_i, u_{i+p}]$. Mas $u \in [u_i, u_{i+p}]$ implica que:

$$\frac{(u - u_i)}{(u_{i+p} - u_i)}$$

seja não-negativa. Por suposição $N_{i,p-1}(u)$ é não-negativa, e então o primeiro termo da Eq. (3.27) é não-negativa. O mesmo é verdadeiro para o segundo termo e então os $N_{i,p}(u)$ são não-negativos.

Propriedade 4- Propriedade da Partição da Unidade: Para um arbitrário intervalo de nós, $[u_i, u_{i+1})$, $\sum_{j=i-p}^i N_{j,p}(u) = 1$ para todos $u \in [u_i, u_{i+1})$. Para provar é considerado:

$$\sum_{j=i-p}^i N_{j,p}(u) = \sum_{j=i-p}^i \frac{u - u_j}{u_{j+p} - u_j} N_{j,p-1}(u) + \sum_{j=i-p}^i \frac{u_{j+p+1} - u}{u_{j+p+1} - u_{j+1}} N_{j+1,p-1}(u)$$

Mudando a variável de soma na segunda somatória de $i-p$ para $i-p+1$, e considerando que $N_{i-p,p-1}(u) = N_{i+1,p-1}(u) = 0$, tem-se:

$$\sum_{j=i-p}^i N_{j,p}(u) = \sum_{j=i-p+1}^i \left[\frac{u - u_j}{u_{j+p} - u_j} + \frac{u_{j+p} - u}{u_{j+p} - u_j} \right] N_{j,p-1}(u) + \sum_{j=i-p+1}^i N_{j,p-1}(u)$$

Aplicando o mesmo conceito recursivamente, tem-se:

$$\sum_{j=i-p}^i N_{j,p}(u) = \sum_{j=i-p+1}^i N_{j,p-1}(u) = \sum_{j=i-p+2}^i N_{j,p-2}(u) = \dots = \sum_{j=i}^i N_{j,0}(u) = 1$$

Propriedade 5- Todas derivadas de $N_{i,p}(u)$ existem no interior do intervalo de nós (onde ela é uma polinomial). No nó $N_{i,p}(u)$ é $(p-k)$ vezes continuamente diferenciável, onde k é a multiplicidade do nó. Então, aumentando o grau, aumenta a continuidade, e aumentando a multiplicidade do nó, reduz a continuidade.

Para analisar o efeito de múltiplos nós, pode-se considerar as funções $N_{0,2}$, $N_{1,2}$, $N_{2,2}$, $N_{5,2}$ e $N_{6,2}$ da Fig. 3.28, reproduzida na Fig. 3.31.

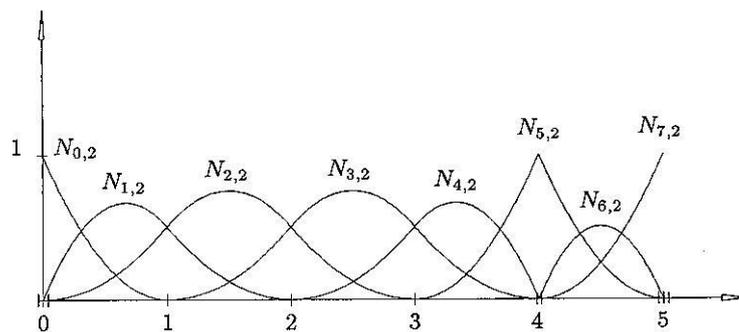


Figura 3.31 - Função base do segundo grau, $U = \{0,0,0,1,2,3,4,4,5,5,5\}$ (PIEGL; TILLER, 1997)

Sabendo que $U=\{0,0,0,1,2,3,4,4,5,5,5\}$ e utilizando a Eq. (3.26) e a Propriedade 1 verifica-se que estas funções são obtidas nos seguintes intervalos de nós e que são nulas fora destes intervalos:

$$N_{0,2}: \{0,0,0,1\}$$

$$N_{1,2}: \{0,0,1,2\}$$

$$N_{2,2}: \{0,1,2,3\}$$

$$N_{5,2}: \{3,4,4,5\}$$

$$N_{6,2}: \{4,4,5,5\}$$

O termo “multiplicidade” é entendido de duas diferentes formas:

- A multiplicidade do nó no vetor nó;
- A multiplicidade de um nó em relação a uma função base específica.

No exemplo, $u=0$ tem multiplicidade 3 no vetor nó U . Mas com relação às funções $N_{0,2}$, $N_{1,2}$, $N_{2,2}$ e $N_{5,2}$, $u=0$ é um nó de multiplicidade 3,2,1 e 0, respectivamente.

Então, a continuidade destas funções em $u=0$ é: $N_{0,2}$ é descontínua; $N_{1,2}$ tem continuidade C^0 ; $N_{2,2}$ tem continuidade C^1 e $N_{5,2}$ não é afetada (todas suas derivadas são nulas em $u=0$, em ambos os lados). $N_{1,2}$ “vê” $u=0$ como um nó duplo, então ela tem continuidade C^0 . $N_{2,2}$ “vê” todos seus nós com multiplicidade 1, então tem continuidade C^1 . Outro efeito da multiplicidade dos nós, como visto pelas funções, é reduzir o número de intervalos “aparentes” nas quais a função é não nula, ou seja, $N_{6,2}$ é não nula somente em u pertencente ao intervalo $[4,5)$, e tem somente continuidade C^0 em $u=4$ e $u=5$.

Propriedade 6- Exceto para o caso em que $p=0$, $N_{i,p}(u)$ possui exatamente um valor de máximo.

3.6.2 Derivadas das funções de base B-spline

A derivada da função de base é dada por (PIEGL; TILLER, 1997):

$$N'_{i,p} = \frac{p}{u_{i+p}-u_i} N_{i,p-1}(u) - \frac{p}{u_{i+p+1}-u_{i+1}} N_{i+1,p-1}(u) \quad (3.28)$$

A diferenciação repetida da Eq. (3.28) permite obter a equação geral da derivada de ordem k :

$$N_{i,p}^{(k)}(u) = p \left(\frac{N_{i,p-1}^{(k-1)}}{u_{i+p}-u_i} - \frac{N_{i+1,p-1}^{(k-1)}}{u_{i+p+1}-u_{i+1}} \right) \quad (3.29)$$

Outra generalização, permite obter a derivada de ordem k de $N_{i,p}(u)$ em termos das funções $N_{i,p-k}(u)$, ..., $N_{i+k,p-k}(u)$:

$$N_{i,p}^{(k)} = \frac{p!}{(p-k)!} \sum_{j=0}^k a_{k,j} N_{i+j,p-k} \quad (3.30)$$

Onde,

$$a_{0,0} = 1$$

$$a_{k,0} = \frac{a_{k-1,0}}{u_{i+p-k+1}-u_i}$$

$$a_{k,j} = \frac{a_{k-1,j} - a_{k-1,j-1}}{u_{i+p+j-k+1} - u_{i+j}} \quad j = 1, \dots, k-1$$

$$a_{k,k} = \frac{-a_{k-1,k-1}}{u_{i+p+1} - u_{i+k}}$$

Deve-se observar que:

- k não deve exceder p (as derivadas de alta ordem são nulas);
- os denominadores envolvendo as diferenças entre os nós podem ser nulos (neste caso o quociente é definido como zero).

Por fim, uma equação adicional para o cálculo das derivadas das funções de base B-spline, em função da k -ésima derivada de $N_{i,p-1}(u)$ e $N_{i+1,p-1}(u)$ é:

$$N_{i,p}^{(k)}(u) = \frac{p}{p-k} \left(\frac{u-u_i}{u_{i+p}-u_i} N_{i,p-1}^{(k)} - \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}} N_{i+1,p-1}^{(k)} \right) \quad k = 0, \dots, p-1 \quad (3.31)$$

3.6.3 Curvas B-spline

De acordo com Piegl e Tiller (1997) uma curva B-spline de grau p é definida por:

$$C(u) = \sum_{i=0}^n N_{i,p}(u) P_i \quad (3.32)$$

Onde P_i são os pontos de controle e $N_{i,p}(u)$ são as funções de base de grau p definidas em um vetor de nós ($m+1$ nós). O polígono formado pelos pontos de controle $\{P_i\}$ é denominado por *polígono de controle*.

Para o cálculo de uma curva B-spline são necessárias três etapas:

- Encontrar o vão em que o nó u_i está no vetor de nós U (Algoritmo no Apêndice I);
- Calcular as funções base não nulas (Algoritmo no Apêndice II);
- Multiplicar as funções base não nulas pelos seus respectivos pontos de controle (Algoritmo no Apêndice III).

As curvas B-Spline podem ser de três maneiras: aberta, fechada e “*clamped*” (do inglês, amarrada) (YAMAGUCHI, 1988), Fig. 3.32. Cada curva dessa é gerada de acordo com a particularidade do vetor nó.

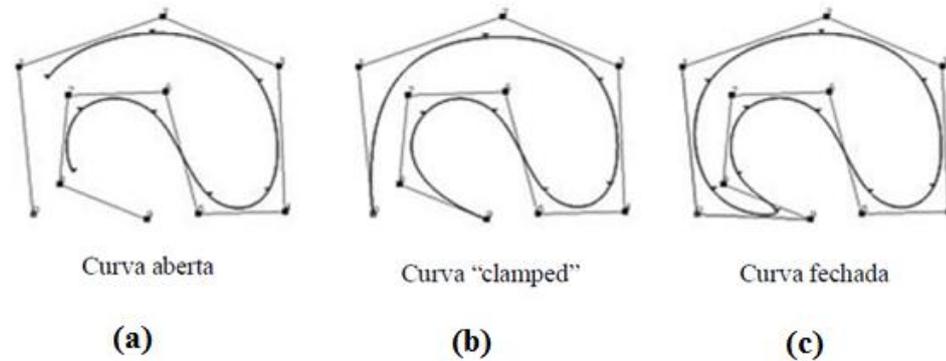


Figura 3.32 - Classificação das curvas B-spline. (a) Curva aberta; (b) Curva “*clamped*” e (c) Curva fechada (BARBARINI,2007)

Segundo (PIEGL; TILLER, 1997) existem dois grupos de vetores nós, os “*unclamped*” e “*clamped*” e dentro de cada grupo eles são divididos em uniforme e não uniforme. Alguns exemplos são:

- “*clamped*” e uniforme: $\{0,0,0,0,1,2,3,4,4,4,4\}$
- “*clamped*” e não uniforme: $\{0,0,0,2,3,6,7,7,7\}$
- “*unclamped*” e uniforme: $\{-3,-2,-1,0,1,2,3,4,5\}$
- “*unclamped*” e não uniforme: $\{0,0,1,2,3,4\}$

Os termos “*clamped*” e “*unclamped*” referem se o primeiro e o último nó se repetem com multiplicidade igual ao grau da curva ou não. Uniforme e não uniforme se refere ao mesmo espaçamento entre os nós, ou não. Para ser uniforme e “*clamped*”, os vãos dos nós “internos” devem possuir o mesmo comprimento e o primeiro e último nó devem ser repetidos $p+1$ vezes, sendo p o grau da curva e *vão* a distância entre um nó e outro. Para ser uniforme e “*unclamped*”, todos os vãos dos nós devem ter o mesmo comprimento.

Em caso de que a estrutura do vetor de nós não assuma nenhuma função particular, nenhum nó repetido nos extremos (“*unclamped*” uniforme ou não uniforme), a curva gerada não irá tocar os pontos das extremidades da linha de controle da curva, Fig 3.33. Esta curva é denominada de curva aberta. Para curvas B-spline abertas, o domínio está no intervalo $[u_p, u_{n-p}]$.

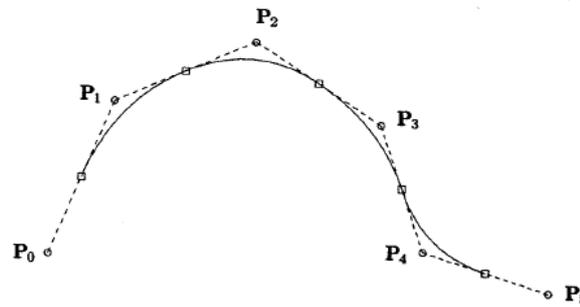


Figura 3.33 - Curva aberta quadrática (PIEGL; TILLER, 1997)

Quando o vetor nó possui multiplicidade dos pontos inicial e final (“*clamped*” uniforme ou não uniforme), impõe-se que a curva inicie e termine nesses nós, assim tem-se a denominada curva “*clamped*”, Fig. 3.32b. Os nós repetidos $p+1$ vezes impõem que os extremos da curva sejam iguais aos extremos dos pontos de controle. Nesse caso o vetor nó é representado da seguinte forma:

$$U = \{\underbrace{a, \dots, a}_{p+1}, u_{p+1}, \dots, u_{m-p-1}, \underbrace{b, \dots, b}_{p+1}\}$$

Assumindo-se que $a = 0$ e $b = 1$, $C(0) = P_0$ e $C(1) = P_n$.

Para esse tipo de curva, o grau, p , o número de pontos de controle, $n+1$, e o número de nós, $m+1$, são relacionados da seguinte forma:

$$m = p + n + 1 \quad (3.33)$$

Para que a curva seja fechada, Fig. 3.32c, o vetor nó deve ser “*unclamped*” (uniforme ou não uniforme) e deve ser acrescentado a ele $p+3$ nós de forma que seus vãos sejam iguais aos $p+3$ primeiros vãos do mesmo vetor nó. Além da particularidade do vetor nó, uma curva fechada exige também que haja repetição dos pontos de controle, onde os $p+1$ últimos pontos acrescentados devem ser iguais ao $p+1$ primeiros pontos.

Em uma curva fechada, o grau, p , o número de pontos de controle, $n+1$, e o número de nós, $m+1$, são relacionados da seguinte forma:

$$m = p + n \quad (3.34)$$

Um exemplo de uma curva fechada, com a repetição de nós e pontos de controle é mostrado a seguir:

Seja o grau da curva igual a 3, o vetor de nós: $U = \{0 \ 2 \ 5 \ 7 \ 8 \ 14 \ 14.5 \ 16\}$ e os pontos de controle: $P = \{(1; 6) (0; 3,5) (2; 2,5) (3,5; 0) (5; 2) (7; 3,5) (5; 6)\}$.

Para obter uma curva fechada é necessário então acrescentar 4 pontos de controle iguais aos 4 primeiros pontos, ficando:

$P = \{(1; 6)(0; 3,5)(2; 2,5)(3,5; 0)(5; 2)(7; 3,5)(5; 6)(1; 6)(0; 3,5)(2; 2,5)(3,5; 0)\}$ e acrescentar 6 novos nós com os vãos iguais aos 6 primeiros vãos do vetor.

Primeiramente calculam-se os valores dos vãos:

$$1^\circ \text{vão} = 2 - 0 = 2$$

$$2^\circ \text{vão} = 5 - 2 = 3$$

$$3^\circ \text{vão} = 7 - 5 = 2$$

$$4^\circ \text{vão} = 8 - 7 = 1$$

$$5^\circ \text{vão} = 14 - 8 = 6$$

$$6^\circ \text{vão} = 14.5 - 14 = 0.5$$

Assim, são acrescentados os novos nós respeitando os valores dos vãos encontrados, ficando: $U = \{0 \ 2 \ 5 \ 7 \ 8 \ 14 \ 14.5 \ 16 \ 18 \ 21 \ 23 \ 24 \ 30 \ 30.5\}$.

A curva fechada obtida está representada na Fig. 3.34.

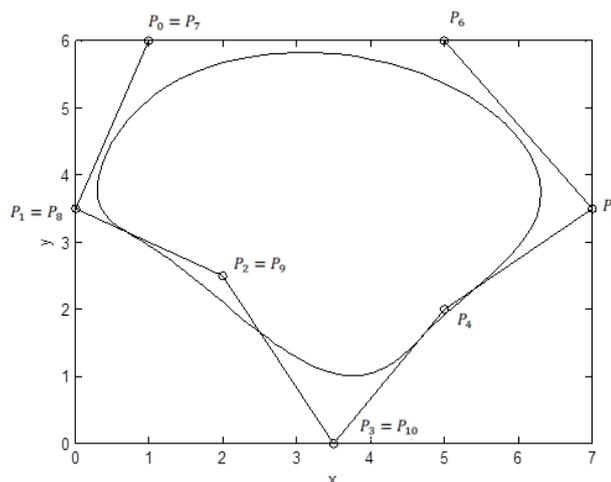


Figura 3.34 – Exemplo de curva fechada

O programa que calcula uma curva fechada está no Apêndice IV.

3.6.4 Propriedades das Curvas B-spline

A seguir estão listadas as principais propriedades das curvas B-spline:

Propriedade 1- Se $n = p$ e $U = \{0, \dots, 0, 1, \dots, 1\}$, então $C(u)$ é uma curva de Bézier, Fig. 3.35.

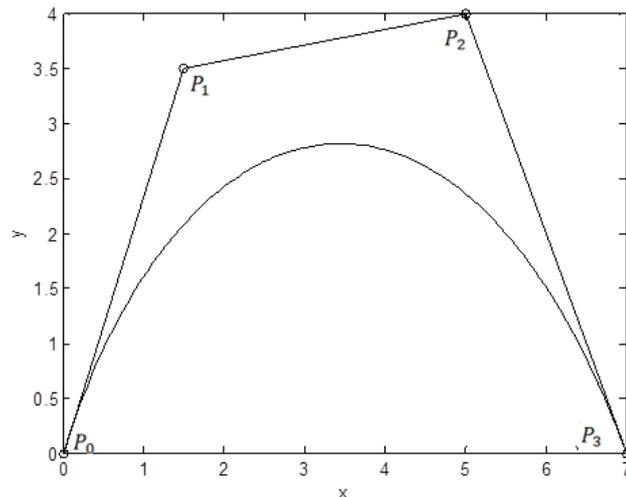


Figura 3.35 - B-spline cúbica com $U = \{0,0,0,0,1,1,1,1\}$, i.e., Curva de Bézier.

Propriedade 2- Propriedade do fecho convexo: a curva B-spline está contida no fecho convexo formado pelo polígono de controle. Se $u \in [u_i, u_{i+1})$, $p \leq i < m - p - 1$, então $C(u)$ está no fecho convexo formado pelos pontos de controle P_{i-p}, \dots, P_i .

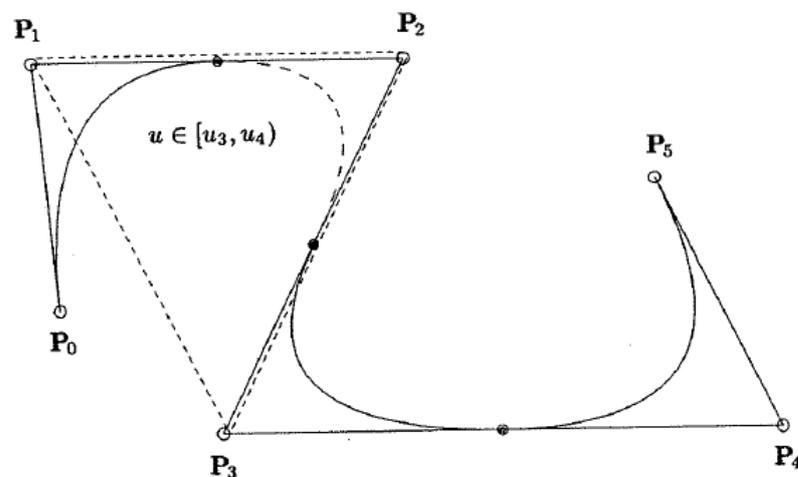


Figura 3.36 - Propriedade do fecho convexo para uma curva B-spline quadrática; para $u \in [u_i, u_{i+1})$, $C(u)$ está no triângulo $P_{i-2}P_{i-1}P_i$, (PIEGL; TILLER, 1997)

Esta propriedade garante que a curva sempre estará no seu interior. Para obter uma curva B-spline de grau p , um ponto sobre a curva se encontra dentro do fecho convexo de p pontos vizinhos. Assim, todos os pontos de uma curva B-spline devem estar dentro da união de todos esses fechos convexos formados por p sucessivos vértices do polígono de controle. A Figura 3.37a ilustra o efeito do fecho convexo para diferentes valores de p . Esse fecho convexo é formado pelo polígono definido por $p+1$ pontos de controle, de forma que esse polígono seja convexo. Ou seja, é encontrado um polígono convexo a cada $p+1$ pontos de controle, onde a curva B-spline deva estar contida dentro dele. Para cada valor de p (grau da curva), é delimitado um polígono diferente, e conseqüentemente um fecho convexo diferente. A Fig. 3.37b ilustra as curvas contidas nos fechos convexos definidos na Fig. 3.37a, para diferentes graus da curva.

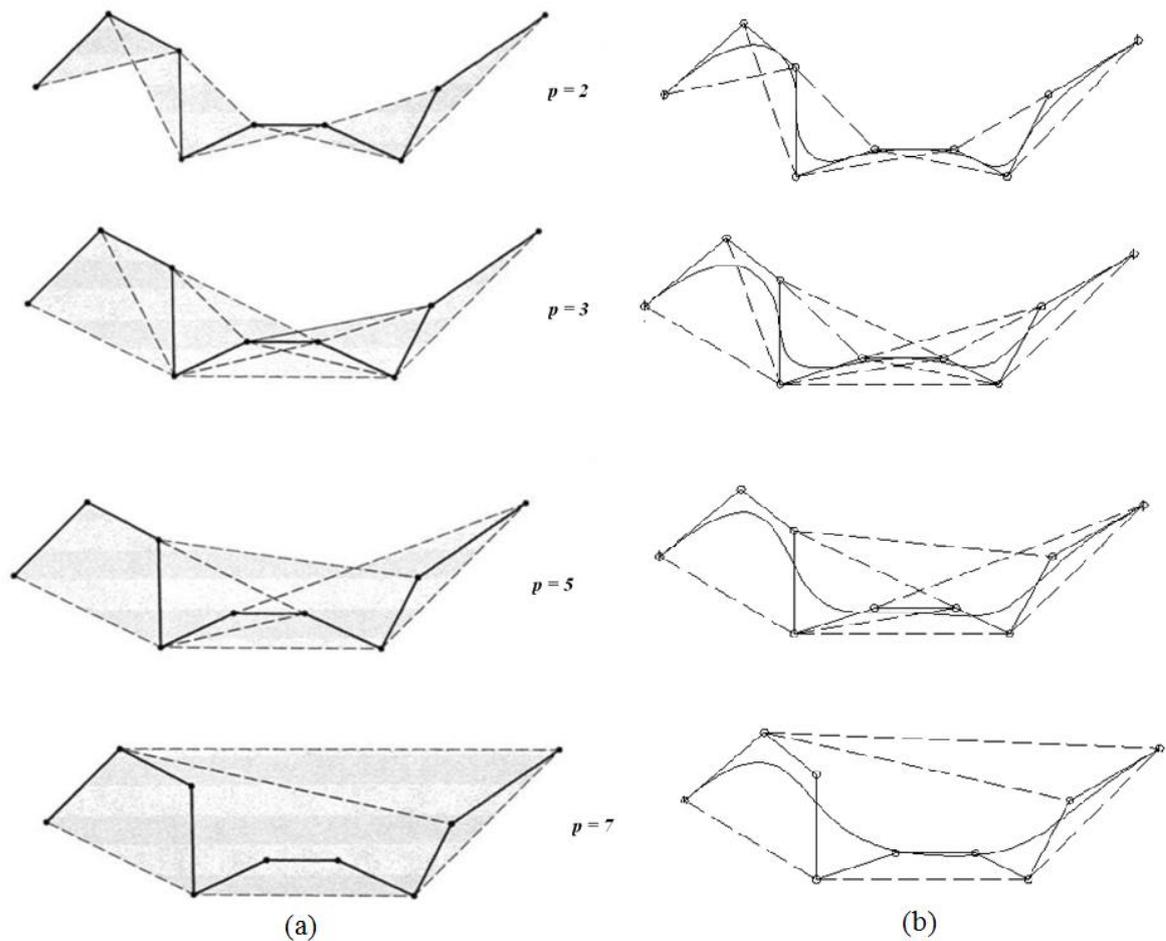


Figura 3.37 – (a) Propriedade do fecho convexo (ROGERS, 2001); (b) Curvas B-spline definidas pelo fecho convexo

Propriedade 3- Suporte local: com a modificação de um ponto P_i da B-spline, a mudança na curva será somente no intervalo $[u_i, u_{i+p+1})$. Esta região corresponde à porção da curva definida entre os $(p+1)/2$ segmentos de retas a esquerda e a direita do P_i modificado. Na Figura 3.38 é mostrada a propriedade do suporte local para uma B-spline cúbica com a modificação do ponto P_4 .

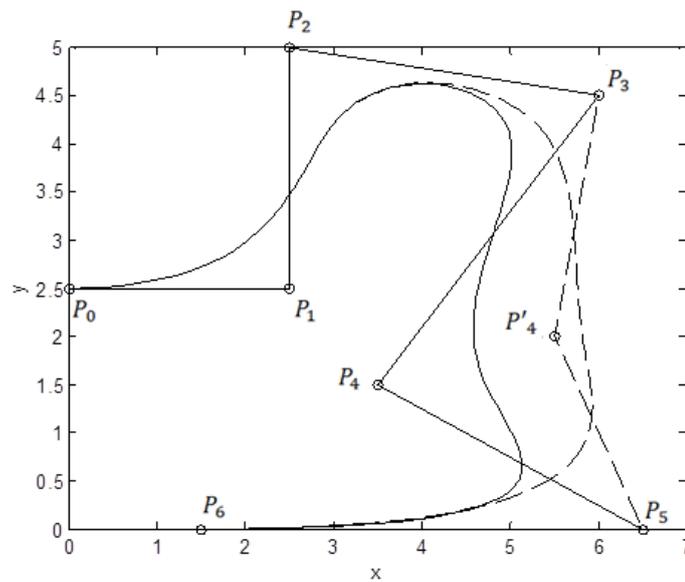
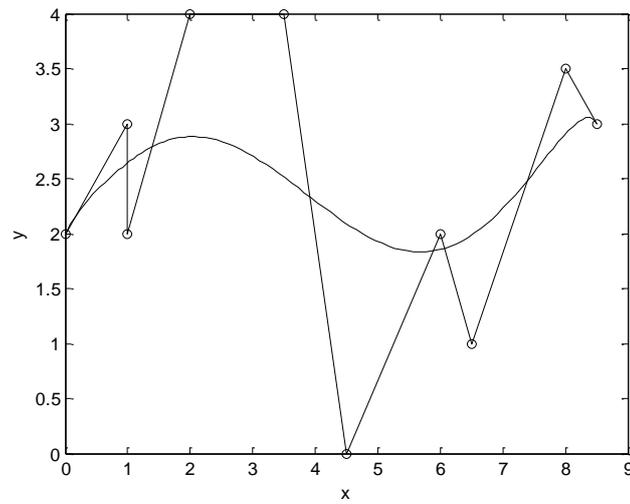
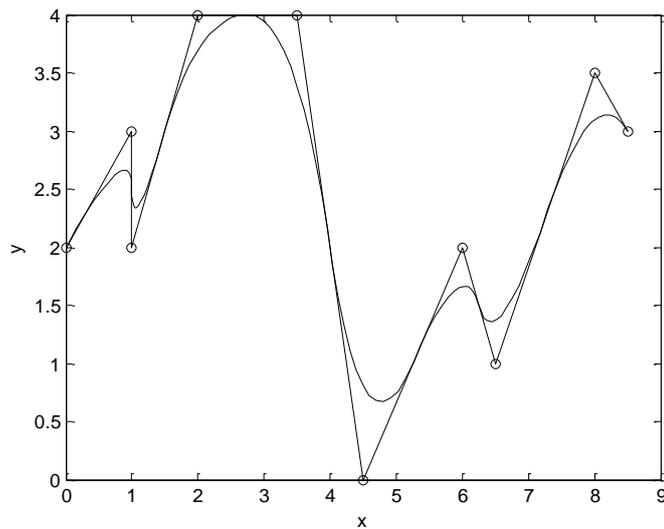


Figura 3.38 - Curva Cúbica em $U=\{0, 0, 0, 0, 1/4, 1/2, 3/4, 1, 1, 1, 1\}$; movendo P_4 (para P'_4) a curva muda no intervalo $[1/4, 1)$

Propriedade 4- O polígono de controle representa uma aproximação linear por partes da curva; essa aproximação é melhorada pela inserção de nó ou pela elevação do grau. Como regra geral, quanto menor o grau, mais perto a curva B-spline fica do polígono de controle. Na Figura 3.39 é representada a diminuição do grau da curva e sua consequente aproximação do polígono de controle.



(a)



(b)

Figura 3.39 - (a) Curva de grau 9 usando vetor nó $U=\{0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\}$;
 (b) Curva quadrática usando o mesmo polígono de controle definido em $U=\{0\ 0\ 0\ 1/8\ 2/8\ 3/8\ 4/8\ 5/8\ 6/8\ 7/8\ 1\ 1\}$

Propriedade 5- A continuidade e diferenciabilidade de $C(u)$ depende da continuidade de $N_{i,p}(u)$. $C(u)$ é infinitamente diferenciável no interior do intervalo de nós $(u_i\ u_{i+1})$ e é $p-k$ vezes diferenciável nos nós com multiplicidade k .

3.6.5 Derivada da curva B-spline

A derivada da curva B-spline é obtida por meio da derivada da função de base e é dada por:

$$C^{(k)}(u) = \sum_{i=0}^n N_{i,p}^{(k)}(u)P_i \quad (3.35)$$

A Figura 3.40 mostra uma B-spline cúbica com a derivada primeira, segunda e terceira computadas em $u = 2/5$.

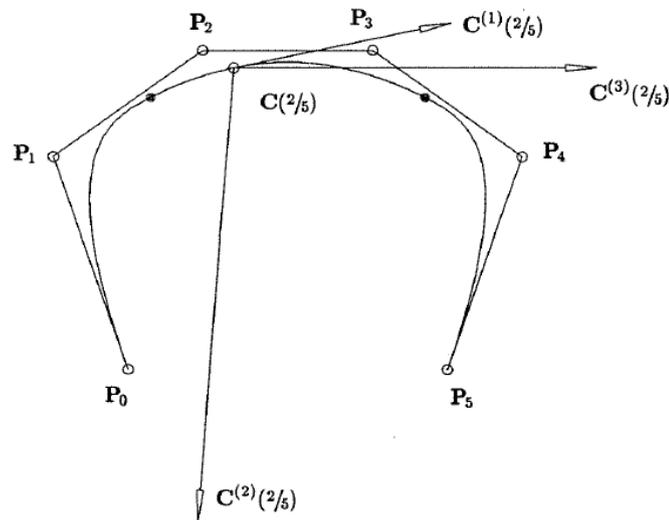


Figura 3.40 - B-spline cúbica em $U=\{0, 0, 0, 0, 1/4, 3/4, 1, 1, 1, 1\}$ com a derivada primeira, segunda e terceira computadas em $u=2/5$ (PIEGL; TILLER, 1997)

A primeira derivada dos pontos inicial e final de uma curva B-spline são dadas por, respectivamente:

$$C'(0) = \frac{p}{u_{p+1}}(P_1 - P_0) \quad (3.36)$$

$$C'(1) = \frac{p}{1-u_{m-p-1}}(P_n - P_{n-1}) \quad (3.37)$$

Como apresentado nas propriedades da B-spline, a continuidade de $C(u)$ depende da continuidade de $N_{i,p}(u)$. No interior do intervalo de nós (u_i, u_{i+1}) a curva é infinitamente diferenciável e é $(p-k)$ vezes diferenciável nos nós com multiplicidade k . Dessa forma, uma

curva de quarto grau pode ser derivada 3 vezes em relação a seu parâmetro (FERREIRA, 2011).

Há algumas formas de alterar o perfil da curva B-spline: alterando os pontos de controle; mudando o vetor nó, que conseqüentemente muda as funções de base; alterando a ordem da curva e a definição da derivada. Para uma mudança nos pontos de controle, a B-spline garante controle local, no qual apenas os pontos vizinhos a ele sofrerão alteração.

3.6.7 Interpolação com B-spline

Sabe-se que quando se define uma B-spline ela não passa pelos pontos de controle. Porém, se for necessário que a curva passe por pontos pré-determinados é preciso calcular um conjunto de pontos de controle e um vetor nó adequado a partir dos pontos dados, para poder traçar a curva (FERREIRA, 2011).

Considerando que seja conhecido um conjunto de pontos $\{Q_k\}$, ($k = 0, \dots, j$) e deseja interpolar esse pontos com uma curva B-spline de grau p . Se para cada Q_k da curva existe um nó \bar{u}_k correspondente, o vetor nó $U = \{u_0, \dots, u_m\}$ pode ser escolhido para o cálculo das funções base. Assim, um sistema de equações lineares pode ser resolvido por:

$$Q_k = C(\bar{u}_k) = \sum_{i=0}^n N_{i,p}(\bar{u}_k) P_i \quad (3.38)$$

Os pontos de controle P_i são as $(n+1)$ incógnitas.

A partir dos pontos de controle obtidos calcula-se a B-spline normalmente com a Eq. (3.32). Dessa forma, a curva resultante passará por todos os pontos $\{Q_i\}$.

Um dos problemas a ser resolvido é a escolha de \bar{u}_k e U , visto que suas escolhas afetam a forma e a parametrização da curva. A seguir são apresentadas algumas metodologias para a definição dos nós \bar{u}_k :

• **Nós igualmente espaçados:**

$$\begin{aligned} \bar{u}_0 &= 0 & \bar{u}_j &= 1 \\ \bar{u}_k &= \frac{k}{j} & k &= 1, \dots, j-1 \end{aligned} \quad (3.39)$$

Esse método não é recomendado, pois pode causar formas indesejadas da B-spline.

• **Método do comprimento de corda:**

Seja ℓ o comprimento total de corda obtido entre os pontos Q_k , dado por:

$$\ell = \sum_{k=1}^j |Q_k - Q_{k-1}| \quad (3.40)$$

Então,

$$\bar{u}_0 = 0 \quad \bar{u}_j = 1$$

$$\bar{u}_k = \bar{u}_{k-1} + \frac{|Q_k - Q_{k-1}|}{\ell} \quad k = 1, \dots, j - 1 \quad (3.41)$$

Este é o método mais utilizado e geralmente adequado, pois permite obter uma boa parametrização da curva tendo uma curva mais comportada.

• **Método Centrípeto**

Considerando que,

$$d_c = \sum_{k=1}^j \sqrt{|Q_k - Q_{k-1}|} \quad (3.42)$$

Então,

$$\bar{u}_0 = 0 \quad \bar{u}_j = 1$$

$$\bar{u}_k = \bar{u}_{k-1} + \frac{\sqrt{|Q_k - Q_{k-1}|}}{d_c} \quad k = 1, \dots, j - 1 \quad (3.43)$$

Cada parâmetro de interpolação \bar{u}_k ($k = 0, \dots, j$) corresponde a um ponto Q_k por onde a B-spline deve passar. Para obter a B-spline, é necessário calcular os pontos de controle P_i ($i = 0, \dots, n$) que definem o fecho convexo, obtidos pela Eq. (3.38). Para o traçado da B-spline é necessário então, definir o vetor nó $U = \{u_0, \dots, u_m\}$ que permite obter as funções base (FERREIRA, 2011).

Dois métodos têm sido utilizados para a definição do vetor nó para uma curva “clamped”. O primeiro consiste em uma distribuição uniforme, ou seja:

$$\begin{aligned} u_0 = \dots = u_p = 0 & & u_{m-p} = \dots = u_m = 1 \\ u_{k+p} = \frac{k}{n-p+1} & & k = 1, \dots, n-p \end{aligned} \quad (3.44)$$

O segundo método é denominado *averaging* e que, a partir do parâmetro de interpolação \bar{u}_k , pode ser obtido por:

$$\begin{aligned} u_0 = \dots = u_p = 0 & & u_{m-p} = \dots = u_m = 1 \\ u_{k+p} = \frac{1}{p} \sum_{i=k}^{k+p-1} \bar{u}_i & & k = 1, \dots, n-p \end{aligned} \quad (3.45)$$

Com esse método os nós refletem a distribuição de \bar{u}_k .

A parametrização do vetor \bar{U} , assim como a ordem, possuem influência na forma da curva. Isso pode ser observado na Fig. 3.41 em que é realizada a interpolação para os três tipos de parametrização mais conhecido, sendo que o vetor U é obtido por *averaging*. Na parametrização de \bar{U} com nós igualmente espaçados a curva torna-se menos suave; em regiões que os pontos estão próximos, a curva tende a ser “encolhida”, o que provoca curvaturas mais fechadas; em regiões que possuem pontos mais distantes a curva tende a ser “esticada”. A parametrização centrípeta garante uma curva suave, porém com a parametrização por comprimento de corda a curva possui um melhor comportamento (FERREIRA, 2011).

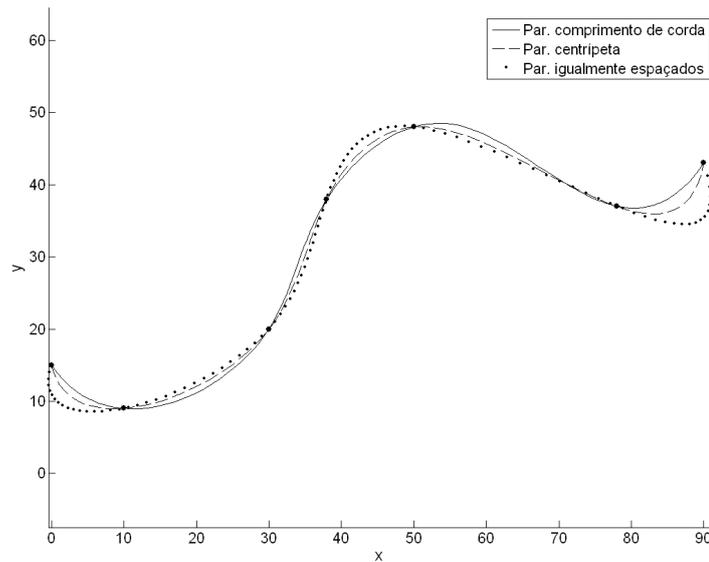


Figura 3.41 - Interpolação com diferentes metodologias de parametrização do vetor \bar{U} (FERREIRA, 2011)

Um elemento muito importante no formato da curva é a escolha do vetor nó U , o qual é responsável pela obtenção das funções de base. Na Figura 3.42 são apresentadas as curvas obtidas pelos mesmos pontos da Fig. 3.41. Neste exemplo, o vetor \bar{U} é obtido com parametrização por comprimento de corda, e o vetor U é calculado igualmente espaçado e utilizando a técnica *averaging*.

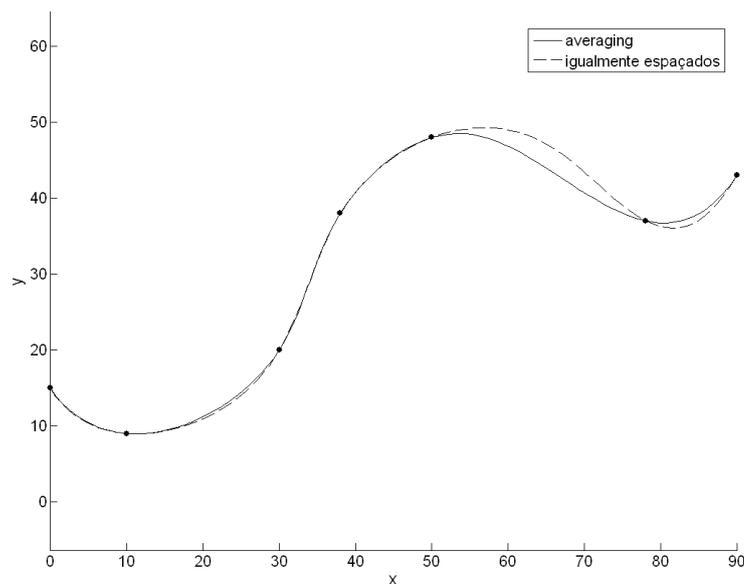


Figura 3.42 - Interpolação com diferentes metodologias de obtenção do vetor U (FERREIRA, 2011)

Na Figura 3.43 é apresentado um exemplo de interpolação dos pontos $\{Q_k\}=\{(0,0), (3,4), (-1,4), (-4,0), (-4,-3)\}$. O vetor \bar{U} foi obtido com a parametrização por comprimento de corda e o vetor de nós U calculado pelo método *averaging*.

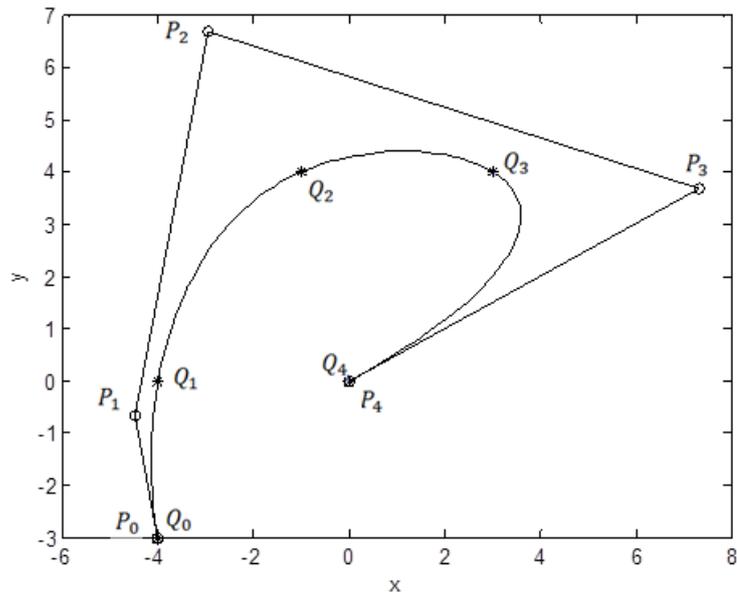


Figura 3.43 - Interpolação por B-spline cúbica usando parametrização por comprimento de corda e vetor nó obtido pelo método de *averaging*.

3.6.8 Aproximação com B-spline

No método de aproximação a curva não passa necessariamente pelos pontos dados. A seguir é apresentado um desses métodos.

- Aproximação usando mínimos quadrados:

Assim como no método de interpolação é necessário encontrar os valores dos pontos de controle para definir o polígono e assim obter a curva desejada.

Assume que $p \geq 1$, $n \geq p$, e Q_0, \dots, Q_j ($j > n$) são dados, e assim busca-se a curva de grau p definida pela Eq. (3.32), sendo Q_k aproximados pelo método dos mínimos quadrados, isto é (PIEGL; TILLER, 1997), (SMITH et al., 1974) e (LYCHE; MORKEN, 2008):

$$e = \sum_{k=0}^j |Q_k - C(\bar{u}_k)|^2 \quad (3.46)$$

é um mínimo com respeito a $n+1$ variáveis, P_i ; os $\{\bar{u}_k\}$ são parâmetros pré calculados.

Para os casos em que os pontos Q_k possuem peso, a Eq. (3.46) fica:

$$e = \sum_{k=0}^j \{w_k |Q_k - C(\bar{u}_k)|^2\} \quad (3.47)$$

Os pesos w_k são valores atribuídos aos pontos Q_k forçando a curva B-spline a aproximar ou distanciar desses pontos. Quanto maior o valor dos pesos, mais próxima dos pontos será a curva.

A Eq. (3.47) é desenvolvida no Apêndice V chegando à Eq. (3.48), na qual os pontos de controle são calculados.

$$(N^T W N) P = R \quad (3.48)$$

onde N é uma matriz $(j+1) \times (n+1)$ de escalares das funções de base calculadas para \bar{u}_k , $k=0, \dots, j$.

$$N = \begin{bmatrix} N_{0,p}(\bar{u}_0) & \cdots & N_{n,p}(\bar{u}_0) \\ \vdots & \ddots & \vdots \\ N_{0,p}(\bar{u}_j) & \cdots & N_{n,p}(\bar{u}_j) \end{bmatrix} \quad (3.49)$$

R é um vetor de $n+1$ pontos

$$R = \begin{bmatrix} N_{0,p}(\bar{u}_0)S_0 + \cdots + N_{0,p}(\bar{u}_j)S_j \\ \vdots \\ N_{n,p}(\bar{u}_0)S_0 + \cdots + N_{n,p}(\bar{u}_j)S_j \end{bmatrix} \quad (3.50)$$

onde: $S_k = w_k Q_k$

W é uma matriz diagonal $(j+1) \times (j+1)$ com os pesos associados a cada ponto Q_k ($k=0, \dots, j$)

$$W = \begin{bmatrix} w_0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w_j \end{bmatrix} \quad (3.51)$$

e por fim, P é o vetor com os pontos de controle:

$$P = \begin{bmatrix} P_0 \\ \vdots \\ P_n \end{bmatrix}$$

Para resolver as Eqs. (3.49) e (3.50), um vetor nó $U = \{u_0, \dots, u_m\}$ e parâmetros $\{\bar{u}_k\}$ são requeridos. Os parâmetros $\{\bar{u}_k\}$ podem ser calculados usando o método do comprimento de corda, Eq. (3.41), e o vetor nó pelo método *averaging*, Eq. (3.45) ou pelo método proposto por (PIEGL; TILLER, 1997).

Seja d_a um número real positivo, denota por $i = \text{int}(d_a)$ o maior inteiro tal que $i \leq d_a$. Para uma curva “*clamped*” é necessário um total de $n + p + 2$ nós, com $n - p$ nós internos, e $n - p + 1$ vãos dos nós internos. Seja:

$$d_a = \frac{j+1}{n-p+1} \quad (3.52)$$

Então os nós internos são definidos por:

$$i = \text{int}(kd_a); \quad \alpha = kd_a - i$$

$$u_{p+k} = (1 - \alpha)\bar{u}_{i-1} + \alpha\bar{u}_i \quad k = 1, \dots, n - p \quad (3.53)$$

Para uma curva fechada, a quantidade de nós é $n + p$, respeitando a repetição dos pontos de controle; sendo $n - p - 2$ nós internos e $n - p - 1$ vãos dos nós internos. Assim d_f é definido por:

$$d_f = \frac{j+1}{n-p-1} \quad (3.54)$$

A definição dos nós internos para esse tipo de curva é dada por:

$$i = \text{int}(kd_f); \quad \alpha = kd_f - i$$

$$u_k = (1 - \alpha)\bar{u}_{i-1} + \alpha\bar{u}_i \quad k = 1, \dots, n + p - 3 \quad (3.55)$$

Para $k=0$, tem-se $u_0 = 0$.

3.7 NURBS (*Non-Uniform Rational B-spline*)

As Curvas NURBS (*Non-Uniform Rational B-Splines*), são assim nomeadas por se tratarem de curvas definidas por funções de base racionais que utilizam vetores de nós não uniformes. Dentre suas vantagens, destacam-se as técnicas para modelar círculos e cônicas em geral. Para tanto, é introduzido o conceito de peso. As curvas NURBS têm associado a cada ponto de controle um valor real que determina o peso daquele ponto de controle. Dessa forma, quanto maior o peso de um ponto de controle, mais influência este ponto exerce sobre a parte da curva no qual atua (VIEIRA, 2005).

NURBS é uma ferramenta industrial padrão para representação e design de geometrias simples e complexas. As razões dessa vasta utilização são:

- a forma matemática comum oferecida pelo método para representar formas analíticas padrões e formas livres;
- fornece uma alta flexibilidade no design as formas;
- o tempo de processamento é razoavelmente pequeno;
- são generalizações das curvas e superfícies de Bézier e B-splines.

De acordo com Piegl e Tiller (1997) a NURBS de grau p é definida como:

$$C(u) = \frac{\sum_{i=0}^n N_{i,p}(u)\omega_i P_i}{\sum_{i=0}^n N_{i,p}(u)\omega_i} \quad a \leq u \leq b \quad (3.56)$$

onde P_i são os pontos que formam o polígono de controle, ω_i são os pesos e $N_{i,p}(u)$ são as funções de base de grau p definidas por um vetor de nós.

A NURBS possui uma função de base racional, obtida por meio de uma relação entre funções de base de B-spline, que é dada por:

$$R_{i,p}(u) = \frac{N_{i,p}(u)\omega_i}{\sum_{i=0}^n N_{i,p}(u)\omega_i} \quad (3.57)$$

Assim, a Eq.(3.56) pode ser reescrita da seguinte forma:

$$C(u) = \sum_{i=0}^n R_{i,p}(u)P_i \quad (3.58)$$

A função base racional $R_{i,p}(u)$ possui as seguintes propriedades:

Propriedade 1- Não negatividade: $R_{i,p}(u) \geq 0$ para todo i, p e $u \in [0; 1]$;

Propriedade 2- Partição de unidade: $\sum_{i=0}^n R_{i,p}(u) = 1$ para todo $u \in [0; 1]$;

Propriedade 3- $R_{0,p}(0) = R_{n,p}(1) = 1$;

Propriedade 4- Para $p > 0$, todo $R_{i,p}(u)$ atinge exatamente um máximo no intervalo $u \in [0; 1]$;

Propriedade 5 - Suporte Local: $R_{i,p}(u) = 0$ para $u \notin [u_i, u_{i+p+1}]$. Além disso, em qualquer intervalo de nós (vão) dado, $R_{i,p}(u)$ será não nula no máximo em $p+1$ vãos.

Propriedade 6- Todas as derivadas de $R_{i,p}(u)$ existem no interior do vão de nós. No nó, $R_{i,p}(u)$ é $p-k$ vezes diferenciável, onde k é a multiplicidade do nó.

Propriedade 7- Se $\omega_i = 1$ para todo i , então $R_{i,p}(u) = N_{i,p}(u)$ para todo i .

Propriedade 8- Para uma curva “clamped” $C(0) = P_0$ e $C(1) = P_n$;

Propriedade 9- Propriedade do fecho convexo: se $u \in [u_i; u_{i+1}]$, então $C(u)$ está dentro do fecho convexo dos pontos de controle P_{i-p}, \dots, P_i ;

Propriedade 10- Todas as derivadas de $C(u)$ existem no interior do vão de nós. No nó, $C(u)$ é $p-k$ vezes diferenciável, onde k é a multiplicidade do nó.

Propriedade 11- Aproximação Local: Se o ponto de controle P_i é movido, ou o peso ω_i é trocado, isso afeta somente a porção da curva no intervalo $u \in [u_i; u_{i+p+1}]$.

As curvas NURBS herdam todas as propriedades das curvas B-spline. Porém, além da edição local de pontos de controle, existe agora a possibilidade de alteração de seus pesos, o que implica uma nova ferramenta para modelagem de curvas.

A propriedade 11 é muito importante para a alteração de forma da curva. Usando a curva NURBS, podem ser utilizados tanto o ponto de controle, como o peso, para a alteração

da forma da curva. A Figura 3.44 mostra o efeito da modificação do peso ω_3 relacionado ao ponto de controle P_3 . Assumindo um intervalo u $[u_i, u_{i+p+1})$, com o aumento de ω_i a curva tende a se aproximar do ponto P_i , já com a redução do fator peso a curva se afasta.

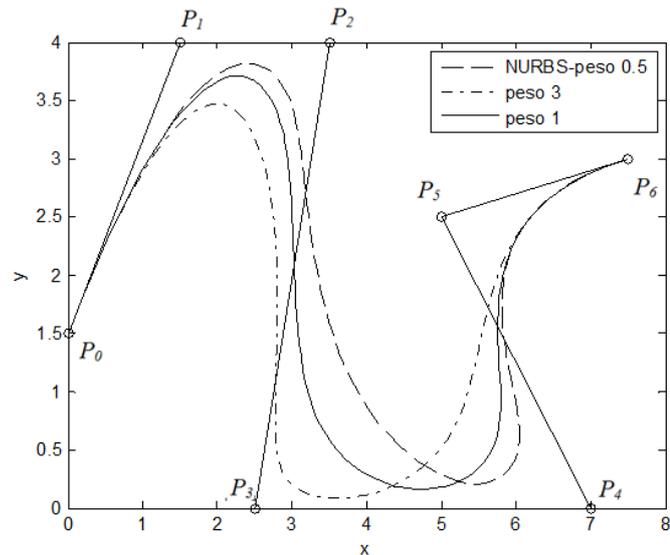


Figura 3.44 - Mudança do peso w_3 relacionado ao ponto de controle P_3

Além disso, o movimento da curva $C(u)$, para um valor fixo de u , é definido como uma linha reta. Os pontos correspondentes a cada ω_i são colineares, isso pode ser observado com a linha tracejada entre $C(u, \omega=0)$ e P_3 , Fig. 3.45.

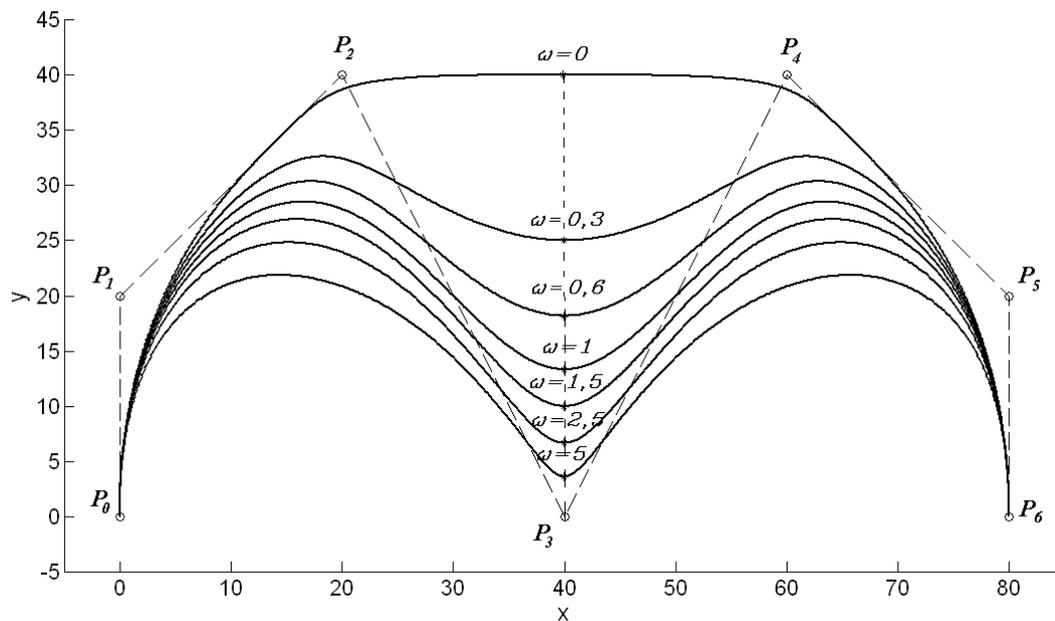


Figura 3.45 - Influência do peso no comportamento da curva (FERREIRA, 2011)

A reta formada pelos pontos correspondentes a pesos diferentes passa por P_3 , e para $0 \leq \omega \leq \infty$ o ponto da curva sempre estará sobre a reta. Para $\omega = \infty$, a curva passará sobre P_3 , e para $\omega = 0$ o ponto P_3 não possui influência sobre a curva.

Há três formas de alterar a forma da curva NURBS: mudança do ponto de controle, alteração do vetor nó e mudança do vetor peso. Alteração da curva por meio da mudança do vetor nó é muito complicada, visto que não é intuitivo, assim os métodos mais utilizados são mudança do vetor peso e dos pontos de controle.

Para uma NURBS com peso igual a 1, a curva se restringe a uma B-spline.

3.8 Conclusão

Neste capítulo foram apresentadas as curvas paramétricas, com destaque para as curvas B-spline e NURBS, que têm sido muito úteis nas conversões de pontos lidos em curvas e superfícies devido à reprodução fiel da superfície e a possibilidade de controle da curva.

As B-splines, originárias das funções de Bézier e das splines, consistem de segmentos de curvas cujos coeficientes polinomiais não dependem de todos os pontos de controle. Desta forma, ao mover um ponto de controle, a área afetada da curva é menor e o tempo de cálculo dos coeficientes é reduzido (FONSECA; CARVALHO, 2007). Com isso, ao reproduzir uma

seção transversal da superfície é possível alterar algum ponto a fim de representar melhor o objeto lido sem alterar o restante da curva.

A B-spline por se apresentar de maneiras diferentes, como aberta, “*clamped*” e fechada, é possível reproduzir diferentes tipos de objetos, com superfícies de seções abertas ou fechadas. Ao utilizar a B-spline através da aproximação é possível também alterar o perfil da curva utilizando os pesos relacionados aos pontos previamente dados, modificando a curva nas regiões necessárias para que a mesma aproxime ou afaste dos pontos a fim de obter uma representação mais real do objeto.

As curvas NURBS (*Non Uniform Rational B-spline*) possuem as mesmas características que as B-splines com um acréscimo de pesos associados a cada ponto de controle. Assim curvas NURBS possuem grande flexibilidade, pois além dos pontos de controle a curva pode ser alterada com a mudança desses pesos, aproximando ou afastando a curva do ponto de controle. Se todos os pesos forem iguais a 1, uma curva NURBS reduz para uma curva B-spline.

A diferença entre os pesos de uma NURBS e de uma aproximação por B-spline com peso é que na primeira os pesos aproximam a curva dos pontos de controle, e na segunda os pesos aproximam a curva do ponto que representa a coordenada do objeto real. Diferença essa, que torna a curva B-spline um atrativo para a sua utilização na reconstrução de uma superfície. Além disso, computacionalmente é mais viável utilizar uma B-spline do que uma NURBS, devido ao fato de que a NURBS é uma razão de funções de base.

Assim, uma B-spline, devido às suas características e propriedades já discutidas anteriormente, se coloca como método de aplicação confiável para a reconstrução de superfícies tridimensionais através de seções transversais.

CAPÍTULO IV

DIGITALIZAÇÃO DE CURVAS E SUPERFÍCIES

4.1 Introdução

É evidente o interesse crescente para o uso de ferramentas computacionais e softwares CAD/CAM para projetar e produzir objetos tridimensionais. Em alguns casos o objeto físico existe e deve ser reproduzido. Muitas pesquisas têm sido feitas a fim de obter um método eficiente para reconstruir superfícies tridimensionais para ser aplicado à diversas áreas tais como medicina, engenharia biomédica, processamento de dados geográficos, paleontologia, na engenharia reversa, entre outros. Neste texto, entende-se por “objeto” um sólido o qual se quer reproduzir sua superfície.

A primeira etapa de reconstruir um modelo é adquirir dados do objeto. Isso pode ser feito usando equipamentos manuais e/ou automáticos, denominados “*scanner*” ou digitalizador que fornece uma coleção de pontos que permite descrever a superfície original. No processo de aquisição de dados, milhares ou milhões de dados podem ser adquiridos para a digitalização do objeto. A fidelidade do modelo digitalizado é altamente dependente da densidade de distribuição de pontos e da resolução do sistema. Em geral, o conjunto de dados é uma nuvem de pontos, não organizados, com densidade de amostragem não uniforme, com lacunas e ruídos (BAJAJ et al., 1995; LARSSON; KJELLANDER, 2006; MOCCOZET et al., 2004; MÜLLER et al., 2000; REMONDINO, 2003; VÁRADY et al., 1996).

A segunda etapa consiste no tratamento dos dados adquiridos, onde são aplicados algoritmos que reduzem a nuvem de pontos a pontos significativos para a etapa do processamento dos dados. Existem diversas técnicas de processamento dos dados para a reconstrução de superfície a partir de nuvens de pontos. Várias pesquisas vêm sendo feitas,

principalmente no uso de técnicas através de fatias (seções transversais) do objeto, onde cada seção é definida por curvas B-splines.

Neste capítulo será apresentado o digitalizador desenvolvido no Laboratório de Automação e Robótica – LAR, da FEMEC/UFU, explicando os passos de digitalização até a utilização das curvas propostas.

4.2 Sistemas de digitalização tridimensionais

A digitalização de objetos tridimensionais é uma ferramenta muito útil no desenvolvimento de produtos devido à crescente utilização das técnicas da engenharia reversa no processo de manufatura. A reprodução da geometria dos produtos proporciona uma considerável vantagem relacionada com a redução dos tempos e custos com um alto grau de precisão (EVBUOMWAN; SIVALOGANATHAN; JEBB, 1996).

Existem vários métodos e sistemas de digitalização de objetos tridimensionais. Fazer uma escolha de qual sistema ou qual método é o mais apropriado para fins práticos às vezes resulta em uma tarefa difícil, já que o sistema selecionado deve ler uma variedade considerável de formas.

Um digitalizador tridimensional é um equipamento que analisa um objeto para obter dados de sua forma e/ou de sua cor, que podem ser usados para construir modelos digitais tridimensionais. O propósito dos digitalizadores tridimensionais é criar uma nuvem de pontos a partir da “leitura” da superfície do objeto físico, que serão utilizados para reconstruir a forma do objeto lido (reconstrução) (BAJAJ et al., 1995; MOCCOZET et al., 2004). Os passos e as técnicas para a reconstrução são denominados de engenharia reversa (LARSSON; KJELLANDER, 2006; MÜLLER et al., 2000; VÁRADY et al., 1996).

A representação geométrica tridimensional dos objetos ou de um ambiente é empregada em diversas áreas tais como na robótica, visão artificial, prototipagem rápida, desenho por computador e medicina. Os métodos de aquisição dos dados dos objetos podem ser por sensoriamento passivo ou ativo. A técnica passiva não interage com o objeto ao passo que o método ativo tem contato com objeto ou projeta alguma forma de energia sobre ele. Pesquisas em visão computacional têm realizado estudos voltados aos métodos passivos que extraem formas de uma ou mais imagens digitalizadas (CURLISS, 1997). Neste projeto é utilizado o método ativo, com sensor sem contato, para aquisição dos dados da superfície. A Figura 4.1

representa uma taxonomia dos métodos ativos de aquisição de formas, cujo estudo completo não faz parte desse trabalho.

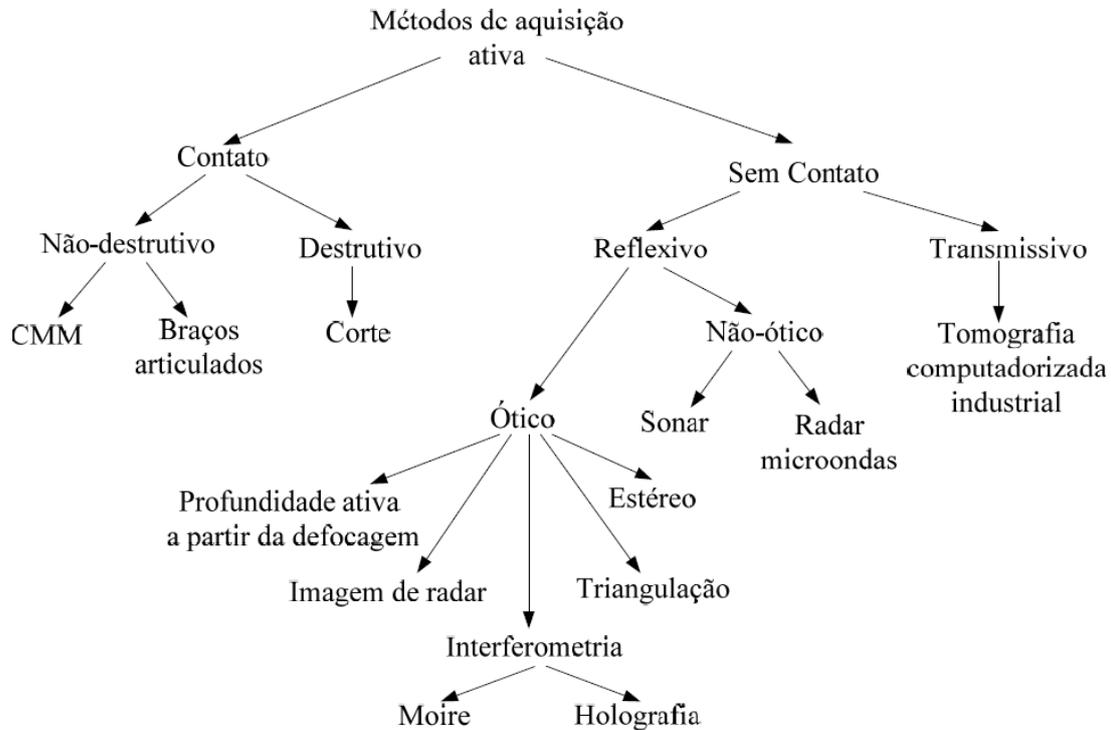


Figura 4.1 - Taxonomia dos métodos ativos de aquisição de formas (CURLESS, 1997)

4.3 O Digitalizador Desenvolvido no LAR

O principal objetivo do digitalizador desenvolvido consiste em permitir a reprodução gráfica de um objeto tridimensional sem zonas mortas que são comuns nos digitalizadores tipo “*scanner*”. Estas zonas mortas obrigam a interferência de um operador para a reconstrução dessa parte do objeto. Outra característica que se procura com o digitalizador é obter as coordenadas da superfície do objeto com resolução suficiente para permitir sua fabricação. Os sistemas tipo “*scanner*” reproduzem a forma do objeto mas não garante sua reprodução dimensional real. Nesse sentido, o digitalizador desenvolvido utiliza duas estruturas que operam de forma cooperativa, imitando o procedimento realizado pela mão direita e a mão esquerda, onde uma manipula o objeto e a outra, o sensor de leitura da superfície (DÍAZ, 2011). Em geral os digitalizadores industriais utilizam um sensor laser que fornece um plano de luz para as leituras. No digitalizador desenvolvido o feixe laser consiste em uma luz unifilar.

Uma característica do digitalizador é a aquisição ordenada e não excessiva de pontos da superfície. Isto permite reduzir o tempo de processamento dos dados para a reprodução da superfície. O digitalizador tridimensional é composto por um sistema de leitura e uma estrutura mecânica. O sistema de leitura é formado por um emissor de feixe laser e uma unidade fotossensível do tipo CCD como receptor, embutidos numa unidade compacta. O receptor capta a luz refletida pela superfície do objeto, permitindo medir a distância entre a superfície e o sensor. O sistema leitor é conhecido como “sensor laser” e é baseado no princípio da triangulação ativa espacial para poder adquirir as medidas (EVERETT, 1995).

A estrutura mecânica é uma estrutura cooperativa onde uma parte tem como objetivo manipular o sensor de leitura e a outra manipular o objeto. A estrutura de manipulação do sensor tem três graus de liberdade (3 gdl), com duas articulações de rotação (R) e uma prismática (P), estrutura tipo RRP, e a de manipulação do objeto tem dois gdl, com uma articulação prismática e uma de rotação, estrutura tipo PR, cujo diagrama esquemático está representado na Fig. 4.2.

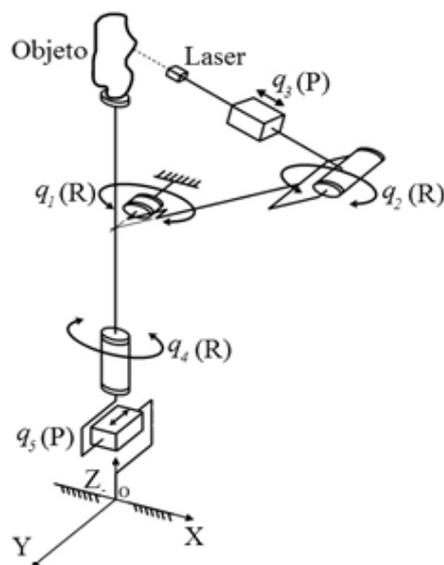


Figura 4.2 - Diagrama da cadeia cinemática do digitalizador desenvolvido no LAR (estrutura tipo RRP+PR) (DÍAZ, 2011)

A combinação das duas estruturas permite o feixe laser “medir” a sua distância até à superfície do objeto na condição em que seu feixe de luz coincida com a normal à superfície, no ponto de leitura. Sensores laser que fornecem excelente resolução, e que a direção da

normal não seja tão importante para o processo de leitura, necessitam operar muito próximo à superfície, com grande possibilidade de colisão. Considerando que, em geral o objeto a ser lido e, posteriormente fabricado, não necessitaria resoluções muito abaixo da ordem de 0,1mm em função dos equipamentos utilizados na sua fabricação, optou-se por utilizar um sensor para leitura a maior distância (no projeto foi considerado cerca de 30 mm), evitando colisões, visto que seria muito difícil prevêê-las e evitá-las automaticamente.

Assim, as duas estruturas têm que trabalhar em conjunto para “ajustar” a normal de cada ponto a ser lido na direção do laser. Para isso, a estrutura PR tem por objetivo reposicionar uma normal à superfície do objeto para o plano de leitura do laser, que corresponde ao plano da estrutura RRP.

Sendo as estruturas cinematicamente simples, seu modelo matemático permite relacionar as coordenadas articulares relativas às leituras do laser com as coordenadas da superfície do objeto, além de sua construção ser robusta e compacta (DÍAZ, 2011). Na Figura 4.3 está apresentado o digitalizador utilizado.

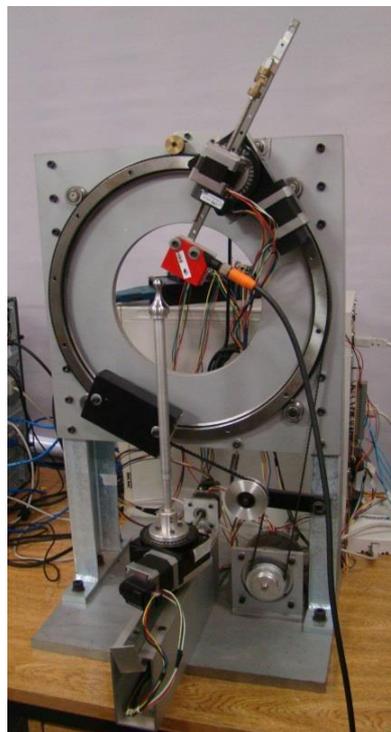


Figura 4.3 - Estrutura mecânica do digitalizador desenvolvido no LAR

4.4 Redução dos pontos lidos

A reconstrução gráfica tridimensional do objeto é feita a partir de suas seções transversais. Em princípio, quanto mais próximas forem essas seções, melhor será a reprodução da superfície do objeto e maior o custo computacional. Para o processo de leitura, as coordenadas lidas pelo sensor e armazenadas como válidas, correspondem à “melhor leitura” do sensor. Esta “melhor leitura” é função das características do sensor, da estrutura e está associada à direção normal à superfície, do ponto lido (DÍAZ, 2011).

Embora o sensor de distância laser, forneça a distância entre ele e o objeto, na realidade seu processador trata os dados adquiridos de forma estatística para definir esta distância. Quando da leitura dos dados da superfície o que se obtém é uma nuvem de pontos conforme representado na Fig. 4.4, no qual a largura da nuvem de pontos é função do equipamento de leitura e devido a ruídos na aquisição de dados.

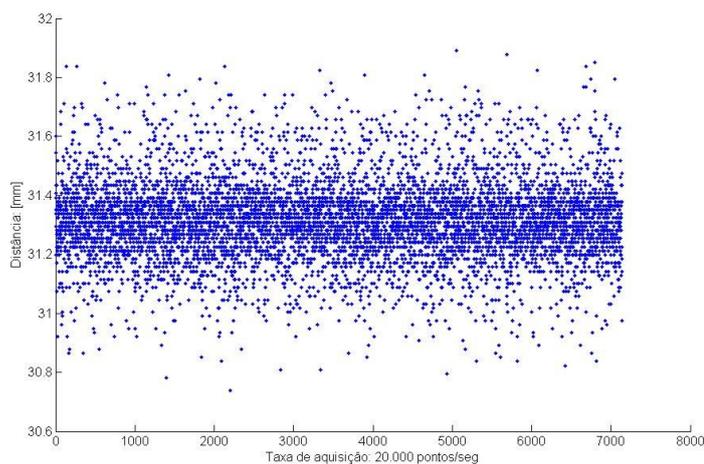


Figura 4.4 - Exemplo de nuvem de pontos adquirido pelo sensor laser para uma distância fixa

A obtenção de uma curva B-spline a partir dessa nuvem de pontos é computacionalmente inviável pelo tempo de processamento. A fim de simplificar o processo de reconstrução e para reduzir o tempo de processamento de dados, foi desenvolvida uma metodologia no Laboratório de Automação e Robótica da FEMEC/UFU para reduzir o número dos pontos adquiridos com o intuito de preservar ao máximo os contornos da secção transversal.

Aplicado em um plano definido pelos eixos x e y , o método consiste na substituição de um conjunto de pontos existentes em uma determinada região por seu centro de massa, considerando que cada ponto possui massa unitária. Como o centro de massa é calculado em uma região definida, as dimensões de cada região é definida pela precisão requerida na

reconstrução da superfície (FONSECA; CARVALHO, 2007). A coordenada z é definida como constante durante todo o processo.

De simulações a melhor geometria da região é um paralelepípedo a qual é definida pelas dimensões “dx” e “dy”. Como a geometria é simples, é fácil identificar os pontos internos e elaborar o algoritmo reduzindo o tempo de processamento de dados. Além disso, é fácil obter a mudança de valores das dimensões, a fim de obter o melhor conjunto de pontos que permite uma boa reprodução da superfície (AQUINO et al, 2009).

Uma vez que ambas as dimensões da região e do número de pontos na região são conhecidos, o centro de massa de cada região pode ser obtido a partir de suas coordenadas x_{CM} e y_{CM} como:

$$x_{CM} = \frac{\sum_{i=1}^n x_i}{n} \quad y_{CM} = \frac{\sum_{i=1}^n y_i}{n} \quad (4.1)$$

Onde “n” representa o número de pontos dentro da região; x_i e y_i são as coordenadas dos pontos i .

A Figura 4.5 representa um conjunto de pontos adquiridos no plano em que as regiões são definidas por dx e dy, e a Fig. 4.6 corresponde ao seu de massa (CM).

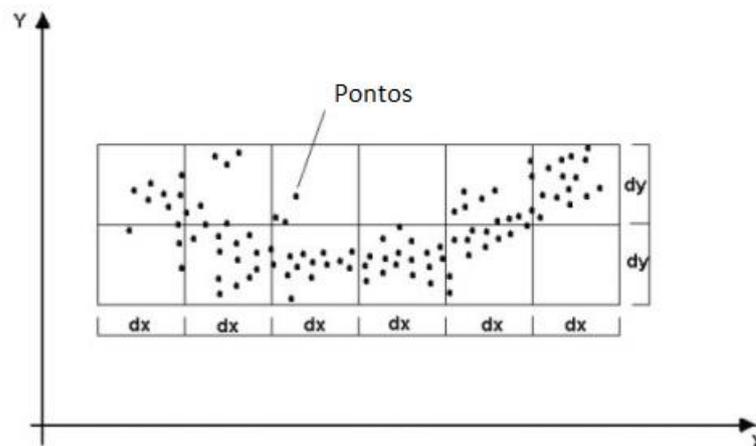


Figura 4.5 - Pontos adquiridos e regiões (AQUINO et al, 2009)

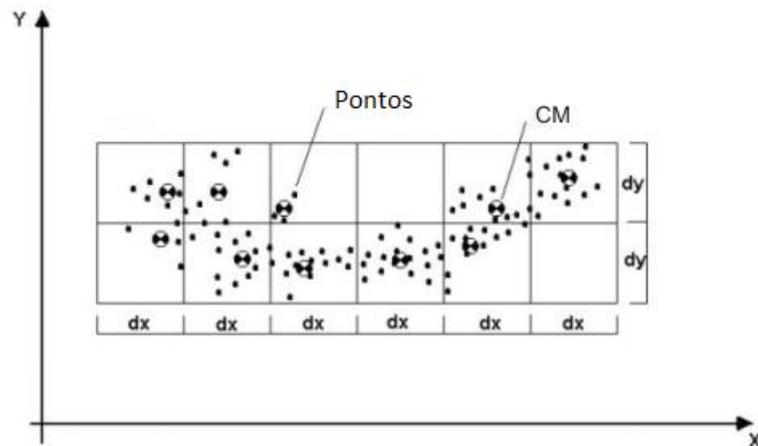


Figura 4.6 - Regiões e seus centros de massa (AQUINO et al., 2009)

O resultado proveniente do método anterior fornece os pontos mais significativos na representação da forma do objeto. Porém, é necessário determinar o contorno da seção transversal definindo a sequência de ligação entre os pontos (centros de massa) obtidos (FONSECA; CARVALHO, 2007).

A ordem nos quais os pontos são ligados pode resultar em inúmeros contornos distintos que, muitas das vezes, não descreve com fidelidade o contorno do objeto digitalizado. Então, para evitar tal problema, foram escolhidos alguns parâmetros, e combinações entre eles, para serem analisados e comparados. É importante salientar que para a utilização de B-splines é necessário conhecer a sequência de pontos que definem a curva (o contorno).

Os parâmetros utilizados foram: a distância entre o CM de referência e cada CM pertencente a uma macrorregião, e o número de pontos (densidade) utilizados na definição de cada CM obtido. O processo é iniciado escolhendo o centro de massa da região com maior densidade de pontos. Isso porque, provavelmente, este centro de massa seja um ponto da superfície. Através desta referência, uma macrorregião é definida englobando os centros de massa vizinhos. A Figura 4.7 ilustra o início do processo e a forma com que a macrorregião é definida.

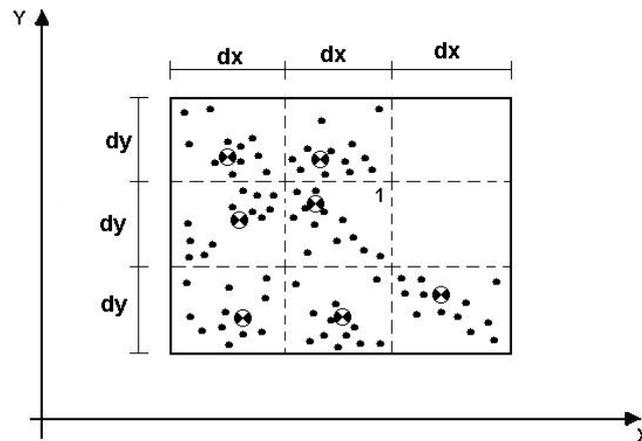


Figura 4.7 – Definição da macrorregião para o início do processo do método de detecção de contorno, a partir da região 1 (FONSECA; CARVALHO, 2007)

A partir de então, os dois parâmetros de base são determinados para cada CM da vizinhança. O maior valor obtido pelo produto entre a quantidade de pontos e a distância entre os CM, define o próximo centro de massa que passará a ser a referência, sendo desprezados os CM restantes presentes na macrorregião. Desta forma, o processo é repetido até que ocorra o fechamento da curva. Ao final do método são obtidos os pontos, em ordem de conectividade, que definem o contorno da curva (CARVALHO; FONSECA, 2005).

Após a realização dos procedimentos de redução de pontos e detecção de contorno, ficam definidos os pontos que representam a seção transversal do objeto. Assim, é possível traçar uma curva para cada seção transversal calculando os pontos de controle e os demais parâmetros para definição da superfície.

4.5 Conclusão

Nos últimos anos, as técnicas de digitalização e reconstrução de formas complexas de objetos tridimensionais têm apresentado um grande desenvolvimento sendo aplicadas em diversas áreas. Para tal, escolher o tipo de digitalizador a ser utilizado é uma tarefa difícil, mas ao mesmo tempo importante.

O digitalizador desenvolvido no LAR obtém os pontos que definem as seções transversais dos objetos representados através de uma “massa” de dados que formam, na realidade, uma nuvem de pontos, nem sempre organizados. A obtenção de uma curva B-spline a partir dessa nuvem de pontos é computacionalmente inviável, para isso é utilizado um

algoritmo computacional que utiliza uma metodologia para reduzir a quantidade de dados utilizando seus centros de massa. Uma grande vantagem dessa metodologia é que os ruídos da aquisição dos dados não afetam o resultado da curva como nos demais processos. Outro aspecto importante é que os dados não precisam ser adquiridos de forma ordenada e alguns vazios não precisam ser “preenchidos” manualmente (com dados criados pelo operador). Isso permite automatizar o processo de reconstrução da superfície.

Com os CM definidos, é utilizada uma metodologia, para definir o sequenciamento dos CM que mais provavelmente fazem parte da superfície.

As atividades dessa dissertação partem do ponto em que os CM já estão sequenciados. A partir daí é possível então reconstruir a superfície, através das seções transversais do objeto utilizando as curvas B-spline. Para obter a curva da seção transversal deve-se definir como obter os pontos de controle e o vetor nó, pois o método utilizado vai afetar diretamente o comportamento da curva e, conseqüentemente, o perfil do objeto.

CAPÍTULO V

APLICAÇÃO DE B-SPLINE FECHADA COM PESO NA RECONSTRUÇÃO DAS SEÇÕES TRANSVERSAIS

5.1 Introdução

Neste capítulo são apresentados os resultados da aplicação da curva B-spline na reconstrução de seções transversais da superfície de um objeto.

É utilizada a B-spline devido sua grande vantagem no controle local, pois assim é possível ajustá-la em locais necessários sem que isso modifique o restante da curva, pontos em que a curva já possui resultado desejável. Além disso, uma curva B-spline possui um tempo de processamento mais rápido e resultado mais direto, por não se apresentar de forma racional como a curva NURBS, conforme pode ser observado nas Eqs. (3.32) e (3.56).

O projeto é desenvolvido usando uma B-spline cúbica fechada capaz de reproduzir seções fechadas de um objeto. Tal curva é utilizada em 3º grau porque são naturalmente adequadas para muitas aplicações de engenharia, pois possuem tempo de processamento de cálculos reduzido, apresentam continuidade na sua derivada, e com ela é possível obter um perfil “S” entre dois pontos consecutivos, garantindo a suavidade na curva.

Para reproduzir a seção transversal, a curva B-spline deve aproximar o máximo possível dos pontos oriundos da digitalização. Pontos esses que, depois de tratados, melhor representam as coordenadas do objeto. Para isso, a B-spline é aproximada pelo método dos mínimos quadrados com peso, para que o erro entre a curva original e a curva obtida seja o menor possível. Os pesos são relacionados a cada ponto e definidos pela relação entre os raios das circunferências formadas a cada três pontos da curva. Quanto maior o valor do peso, mais a curva se aproxima do ponto.

Para um melhor resultado, ainda são utilizados ajustes na curva baseados em características específicas da mesma, como a definição da posição da tangente do primeiro ponto.

A conceituação teórica das curvas B-spline foi apresentada no Capítulo III e a seguir são apresentados o método desenvolvido e exemplos de simulações aplicando a B-spline fechada com peso.

5.2 Desenvolvimento do método

Para a aplicação do método é considerado que o conjunto de pontos $Q_k (k = 0, \dots, j)$, que definem a seção do objeto, é conhecido.

O método pode ser dividido em duas etapas. A primeira, que consiste em obter uma B-spline cúbica aberta do tipo “*clamped*” a partir dos pontos conhecidos e, a segunda etapa, onde a B-spline cúbica fechada é obtida a partir da curva aberta.

A primeira etapa do método é definir uma B-spline cúbica aberta do tipo “*clamped*”. Para isso, primeiramente, é encontrado um vetor nó $\bar{U} = \{\bar{u}_0, \dots, \bar{u}_j\}$ pelo método do comprimento de corda, Eq. (3.41). Cada parâmetro $\bar{u}_k (k = 0, \dots, j)$, corresponde a um ponto $Q_k (k = 0, \dots, j)$ conhecido.

O próximo passo é definir a quantidade de pontos de controle. Essa quantidade é definida, primeiramente, pela quantidade de pontos $Q_k (k = 0, \dots, j)$ que invertem a curvatura, descrito no item 5.3.1. Com a quantidade de pontos de controle é possível definir o vetor nó $U = \{u_0, \dots, u_m\}$, utilizando as Eqs. (3.52) e (3.53). O número de nós, $m+1$, o número de pontos de controle, $n+1$ e o grau, p , são relacionados da seguinte forma, Eq.(3.33):

$$m = p + n + 1$$

O próximo passo é definir os pesos, $w_k (k = 0, \dots, j)$, relacionados a cada ponto $Q_k (k=0, \dots, j)$. Os pesos são definidos pela relação entre os raios das circunferências formadas por três pontos consecutivos, conforme descrito no item 5.3.2.

Com esses parâmetros já calculados, é então usado o método dos mínimos quadrados, Eq. (3.48), para encontrar os valores dos pontos de controle. Assim, tem-se os pontos de controle, e o vetor nó que definem a B-spline cúbica aberta do tipo “*clamped*”.

A segunda etapa consiste em obter uma curva fechada. Para isso é necessário definir um novo vetor nó da forma “*unclamped*” (definição no item 3.6.3). Esse vetor nó é definido de acordo com as Eqs. (3.54) e (3.55).

Como característica de uma curva fechada, o vetor nó possui os $p+3$ últimos vãos iguais aos $p+3$ primeiros vãos. Assim, com o vetor nó já obtido é necessário que se acrescente $p+3$ nós de modo que eles possuam vãos de tamanhos iguais aos $p+3$ primeiros vãos desse mesmo vetor nó. Para $p=3$, tem-se:

$$U = \{u_0 \underset{d_0}{\smile} u_1 \underset{d_1}{\smile} u_2 \underset{d_2}{\smile} u_3 \underset{d_3}{\smile} u_4 \underset{d_4}{\smile} u_5 \underset{d_5}{\smile} u_6, \dots, u_m \underset{d_0}{\smile} u_{m+1} \underset{d_1}{\smile} u_{m+2} \underset{d_2}{\smile} u_{m+3} \underset{d_3}{\smile} u_{m+4} \underset{d_4}{\smile} u_{m+5} \underset{d_5}{\smile} u_{m+6}\}$$

Além da repetição dos nós, é necessário também que aos pontos de controle obtidos pela Eq. (3.48), sejam acrescentados $p+1$ pontos, de modo que esses pontos sejam iguais aos $p+1$ primeiros. Para $p=3$, tem-se:

$$P = \{P_0, P_1, P_2, P_3, P_4, P_5, \dots, P_n, P_0, P_1, P_2, P_3\}$$

O novo vetor nó e os pontos de controle, após a repetição, possuem dimensões que respeitam o relação, Eq. (3.34):

$$m = p + n$$

Assim, com os pontos de controle com repetição e o novo vetor nó já definidos, é possível traçar a curva B-spline cúbica fechada que se aproxima dos pontos que representam as coordenadas do objeto. No entanto, somente a consideração dos pesos em cada ponto conhecido não forneceu bons resultados no ajuste de algumas curvas. Para atingir os resultados esperados foram ainda considerados: um método para seleção do ponto inicial para aplicação do método; correção de alguns pontos de controle, considerando-os sobre a tangente do ponto inicial e alteração, quando necessário, da quantidade de pontos de controle. Esses métodos são detalhados no item 5.3 e exemplos de sua aplicação no item 5.4.

5.3 Metodologia de ajuste de curva

Nesta seção são apresentadas e analisadas as etapas da B-spline cúbica para representar o conjunto de pontos conhecidos, $Q_k (k = 0, \dots, j)$, que descrevem a seção do objeto.

5.3.1 Definição da quantidade de pontos de controle em função do número de inversões da curva

Os pontos de controle P_i são as $(n+1)$ incógnitas, porém para calculá-los é necessário que se saiba qual a sua quantidade. Neste trabalho, inicialmente, a quantidade de pontos de controle foi definida pela quantidade de pontos $Q_k (k = 0, \dots, j)$ que invertem a curvatura.

Dado o conjunto de pontos que definem a curva, $Q_k (k = 0, \dots, j)$, são selecionados três pontos consecutivos (Q_0, Q_1 e Q_2) para definir o vetor normal, \vec{G} , a eles, considerando a seção do objeto paralelo a um sistema de coordenadas $OXYZ$, conforme Fig. 5.1.

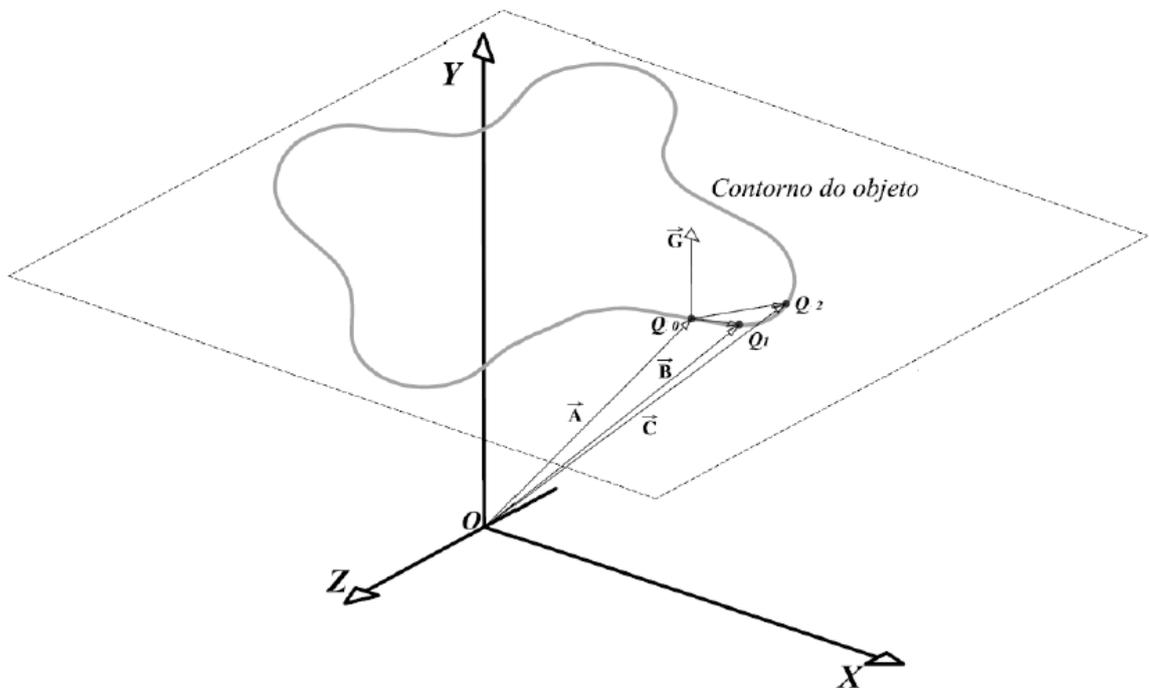


Figura 5.1 – Pontos para definição do vetor normal (DÍAZ, 2011)

Da Figura 5.1 pode-se escrever:

$$\overrightarrow{OQ_0} = \vec{A}; \quad \overrightarrow{OQ_1} = \vec{B}; \quad \overrightarrow{OQ_2} = \vec{C} \quad (5.1)$$

$$\vec{G} = \overrightarrow{Q_0 Q_1} \times \overrightarrow{Q_0 Q_2} = (\vec{B} - \vec{A}) \times (\vec{C} - \vec{A}) \quad (5.2)$$

O vetor \vec{G} é o produto vetorial dos outros dois vetores, ou seja, a normal a eles. Ele é calculado a cada três pontos, para todos os pontos dados. Dependendo da posição dos pontos, \vec{G} pode estar apontando para cima (positivo) ou para baixo (negativo). A mudança do sinal do vetor indica o ponto de inversão de curvatura. A quantidade de pontos de controle é, então, definida por essa quantidade de pontos de inversão.

A Figura 5.2 apresenta os pontos de inversão em um dado conjunto de pontos $Q_k(k=0, \dots, j)$.

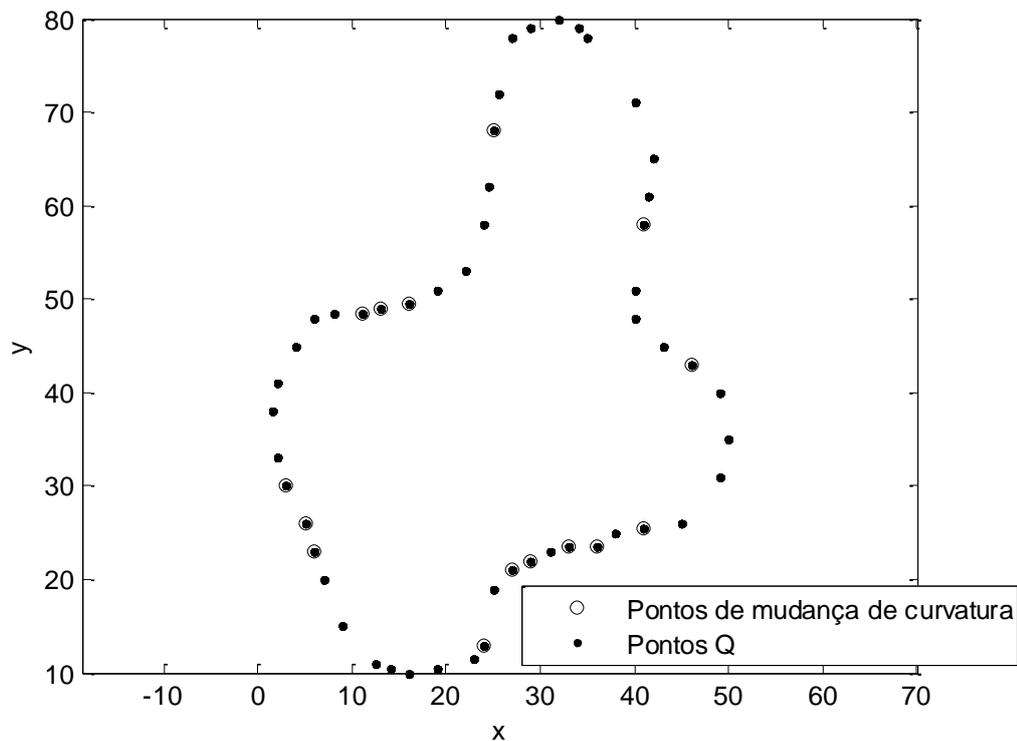


Figura 5.2 – Pontos de inversão de curvatura

A quantidade de pontos de controle influencia no resultado da curva obtida, ou seja, quanto mais pontos de controle mais a curva se aproxima dos pontos dados, $Q_k (k = 0, \dots, j)$. Portanto, quando os pontos digitalizados representam uma seção mais uniforme, sem muitas inversões de curvaturas, a quantidade de pontos de inversão é pequena, e conseqüentemente a quantidade de pontos de controle também. Assim a curva traçada pode não apresentar um resultado considerado ótimo, como pode ser observado na Fig. 5.3.

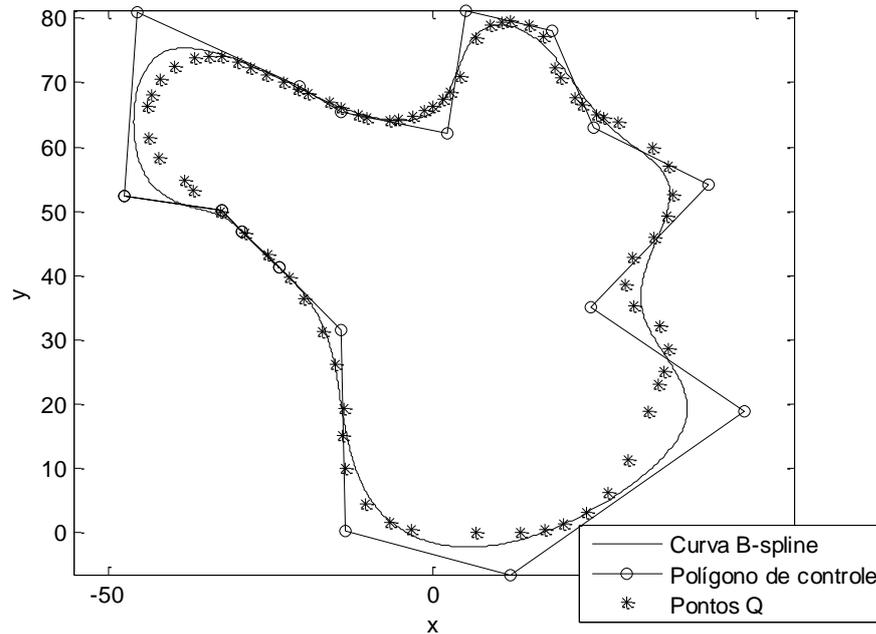


Figura 5.3 – Quantidade de pontos de controle definido pelos pontos de inversão de curvatura

5.3.2 Definição dos pesos

Os pesos são definidos pela relação entre os raios das circunferências formadas a cada três pontos consecutivos de Q_k ($k = 0, \dots, j$).

Tem-se Q_k , Q_{k+1} e Q_{k+2} definindo um triângulo. O encontro das mediatrizes dos lados $\overline{Q_k Q_{k+1}}$ e $\overline{Q_{k+1} Q_{k+2}}$ define o centro da circunferência que passa por esses três pontos, o raio então é definido pela distância do centro a um dos pontos, como na Fig. 5.4. Assim, o raio r_{k+1} corresponde ao ponto Q_{k+1} .

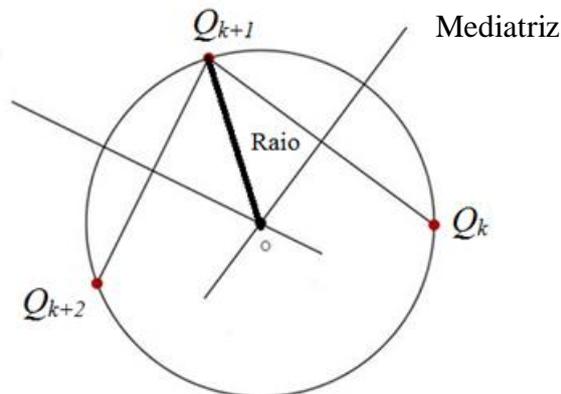


Figura 5.4 – Circunferência e raio definidos por três pontos sequenciais de Q_k

Para $j+1$ pontos, tem-se $j+1$ valores de raios, $r=\{r_0, \dots, r_j\}$. Em situações em que os pontos se encontram alinhados os valores dos raios tendem a infinito. Para realizar as operações a seguir, esses valores são descartados.

Tendo os valores dos raios são obtidos a média, \bar{r} , e o desvio padrão, σ , respectivamente:

$$\bar{r} = \frac{1}{j} \sum_{k=0}^j r_k \quad (5.3)$$

$$\sigma = \frac{\sum_{k=0}^j (r_k - \bar{r})^2}{j - 1} \quad (5.4)$$

Assim, define-se o raio máximo, $rmax$, pela Eq. (5.5):

$$rmax = \bar{r} + \sigma \quad (5.5)$$

A definição do raio máximo, Eq. (5.5), foi feita a partir de simulações considerando o raio máximo como sendo a soma da média dos raios e variando a cada metade do desvio padrão, chegando como melhor resultado a Eq. (5.5).

O peso, w_k , é então definido pela razão entre o raio máximo e cada raio encontrado, $r=\{r_0, \dots, r_j\}$, Eq. (5.6):

$$w_k = \frac{rmax}{r_k} \quad k = 0, \dots, j \quad (5.6)$$

Assim, tem-se um peso relacionado a cada raio. Como o peso e o raio são inversamente proporcionais, evidente pela Eq. (5.6), quanto menor o raio, maior o peso relacionado a ele. Ou seja, nos pontos em que o raio é muito pequeno, é automaticamente atribuído a eles um peso maior para que force a curva a aproximar ao máximo à esses pontos.

Então, para que a curva se aproxime do ponto é necessário que se aumente o valor do peso nele. Em locais em que os pontos estão alinhados faz-se necessário que a curva também se aproxime ao máximo, portanto, o valor do peso nesses pontos também deve ser o maior possível. Para isso é definido o peso máximo como o maior valor dos pesos obtidos na Eq.

(5.6), que é o valor do peso definido para o menor raio. Esse valor, então, é também atribuído aos pontos em que estão mais alinhados, sendo eles considerados os pontos em que o raio seja maior que o raio máximo e quando o raio é infinito.

A seguir um exemplo da relação entre os raios e os pesos.

Seja:

$Q = \{(30,10) (31.5,13) (33,16) (35,20) (40,23) (45,26) (50,30) (48,34) (45,38) (42,44) (40,46) (38,48) (37,50) (35,51) (33,53) (30,55) (29,59) (28.5,64) (27,68) (26,70) (24,64) (22.5,58) (20,50) (19,48) (18.5,47) (16.5,42) (15,40) (13,37) (11,33) (10,30) (12,26) (13.5,23) (15,20) (16,17) (17,15) (19,13) (21,11) (22,10) (24,6) (25,3) (27,3) (28,6) (29,8)\}$

Pela relação do encontro das mediatrizes tem-se os valores dos raios, em milímetro, encontrados:

$r = \{\text{infinito infinito infinito } 9.2195 \text{ infinito } 45.5748 \quad 4.3690 \quad 26.3688 \quad 32.5960$
 $14.9164 \text{ infinito } 7.9057 \quad 3.5355 \quad 7.9057 \quad 16.3248 \quad 5.3600 \quad 31.5062 \quad 17.9920 \quad 31.0456$
 $3.5355 \quad 81.4900 \quad 125.7927 \quad 33.0941 \text{ infinito } 39.1352 \quad 15.0520 \quad 55.0142 \quad 32.5000 \quad 26.9258$
 $5 \text{ infinito infinito } 22.9810 \quad 19.0394 \quad 7.9057 \text{ infinito infinito } 9.2195 \quad 26.9258 \quad 2.2361$
 $2.2361 \quad 19.0394 \text{ infinito}\}$

Os valores “infinito” encontrados significam que os três pontos (o ponto anterior, o ponto analisado e o ponto posterior) estão alinhados, pois o raio de uma reta é infinito. Como já mencionado, para cálculo da média e desvio padrão esses valores são ignorados. Assim, tem-se:

$$\bar{r} = 24.4295 \text{ e } \sigma = 25.4479$$

De acordo com a Eq. (5.5), $r_{max} = 49.8774 \text{ mm}$. Assim, o peso obtido pela Eq. (5.6) para cada valor do raio, exceto para raio “infinito”, pela Eq. (5.6) é:

$w_k = \{\text{--- --- --- } 5.4100 \text{ --- } 1.0944 \quad 11.4163 \quad 1.8915 \quad 1.5302 \quad 3.3438 \text{ --- } 6.3090$
 $14.1075 \quad 6.3090 \quad 3.0553 \quad 9.3054 \quad 1.5831 \quad 2.7722 \quad 1.6066 \quad 14.1075 \quad 0.6121 \quad 0.3965$

1.5071 --- --- 1.2745 3.3137 0.9066 1.5347 1.8524 9.9755 --- --- 2.1704 2.6197
6.3090 --- --- 5.4100 1.8524 22.3059 22.3059 2.6197 ---}

e o peso máximo (maior valor do peso encontrado) é 22.3059, que é o valor associado ao ponto de menor raio.

Percebe-se que os valores de r_{21} , r_{22} e r_{27} são maiores que o valor encontrado para o raio máximo, portanto, para esses raios e para os raios “infinito”, o valor do peso definido é igual ao peso máximo, para que force a curva a aproximar mais desses pontos. Assim, finalmente, tem-se o vetor de pesos relacionados aos pontos Q_k :

$w_k = \{22.3059 \ 22.3059 \ 22.3059 \ 5.4100 \ 22.3059 \ 1.0944 \ 11.4163 \ 1.8915 \ 1.5302$
 $3.3438 \ 22.3059 \ 6.3090 \ 14.1075 \ 6.3090 \ 3.0553 \ 9.3054 \ 1.5831 \ 2.7722 \ 1.6066$
 $14.1075 \ 22.3059 \ 22.3059 \ 1.5071 \ 22.3059 \ 1.2745 \ 3.3137 \ 22.3059 \ 1.5347 \ 1.8524$
 $9.9755 \ 22.3059 \ 22.3059 \ 2.1704 \ 2.6197 \ 6.3090 \ 22.3059 \ 22.3059 \ 5.4100 \ 1.8524$
 $22.3059 \ 22.3059 \ 2.6197 \ 22.3059\}$

As Figuras 5.5 e 5.6 ilustram a diferença de uma curva B-spline aproximada a pontos sem e com peso, respectivamente. Na Figura 5.6 os pesos são definidos pelo exemplo anterior.

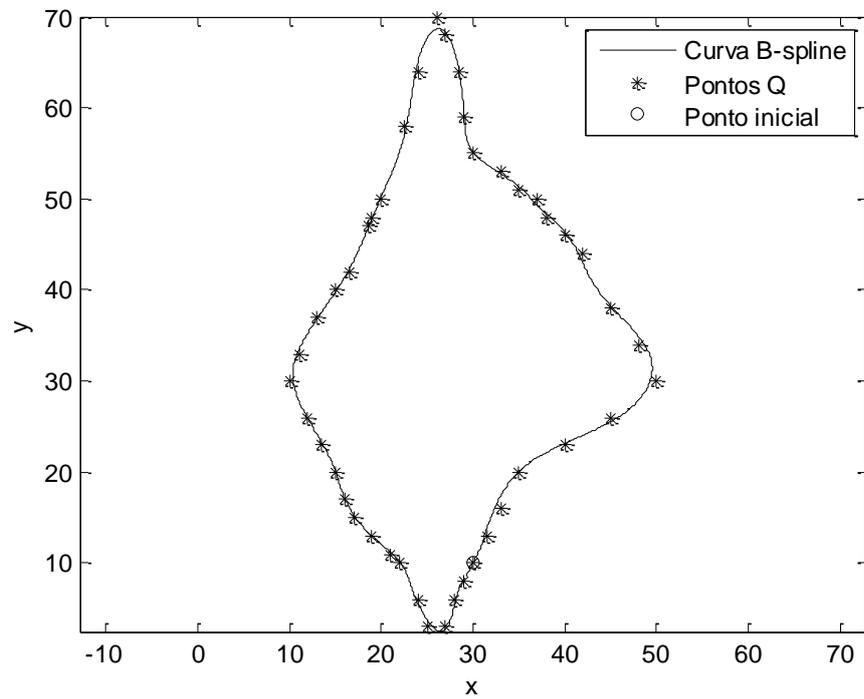


Figura 5.5 – B-spline cúbica aproximada com os pontos sem considerar os pesos

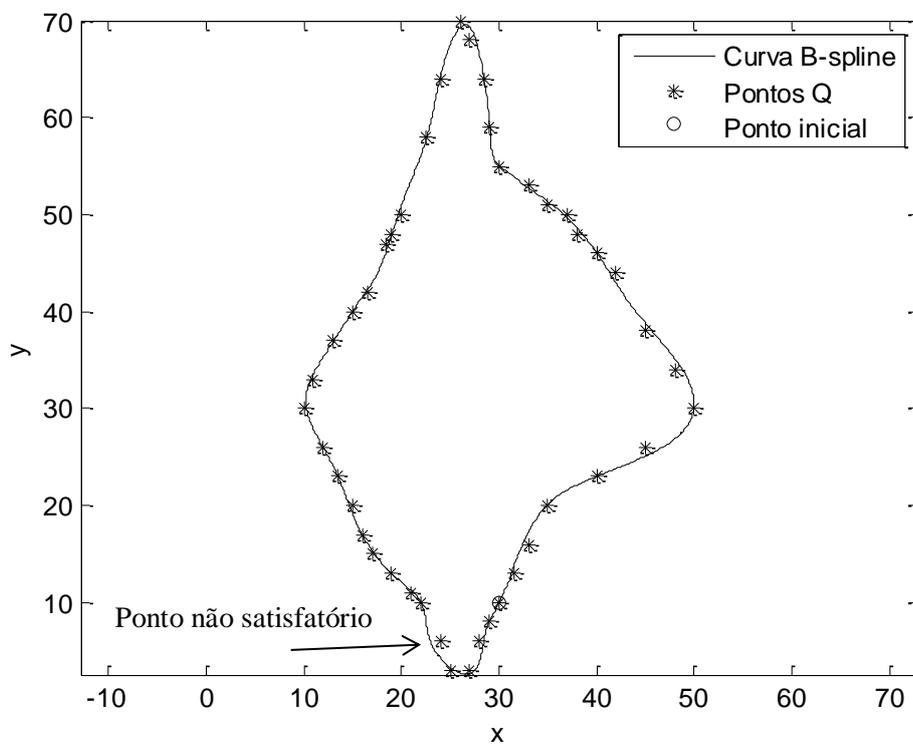


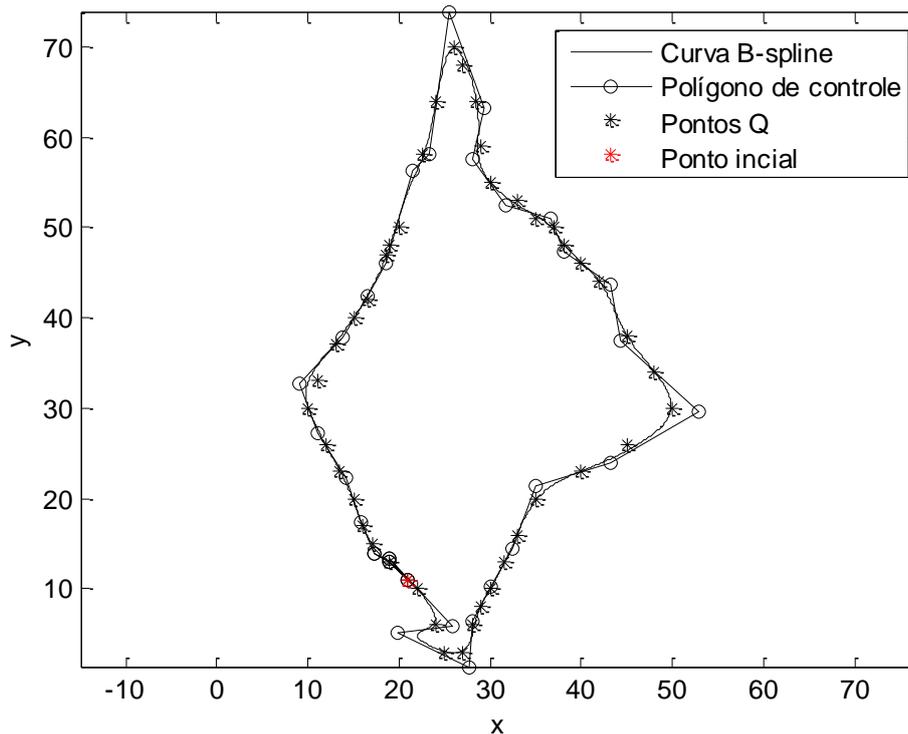
Figura 5.6 – B-spline cúbica aproximada considerando os pontos com peso

Pode-se perceber na Fig. 5.6 que a curva se aproxima mais dos pontos nas extremidades devido ao peso adotado a eles. No entanto, ainda não foi considerado satisfatório para vários exemplos, como pode ser observado no ponto Q_k indicado na Fig. 5.6.

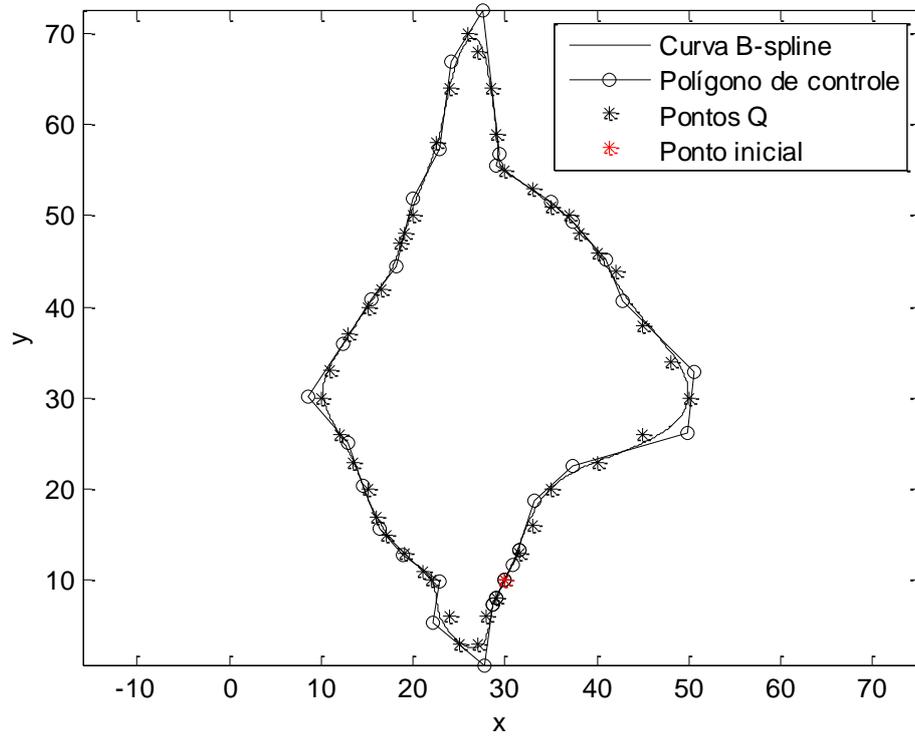
5.3.3 Definição do ponto inicial

Tendo os pontos oriundos da digitalização é necessário que se defina qual o ponto de início para o traçado da curva, pois dependendo do ponto a curva obtida é apresentada de forma diferente. Assim, deve-se definir o ponto inicial para que se obtenha uma curva que melhor represente a seção transversal original.

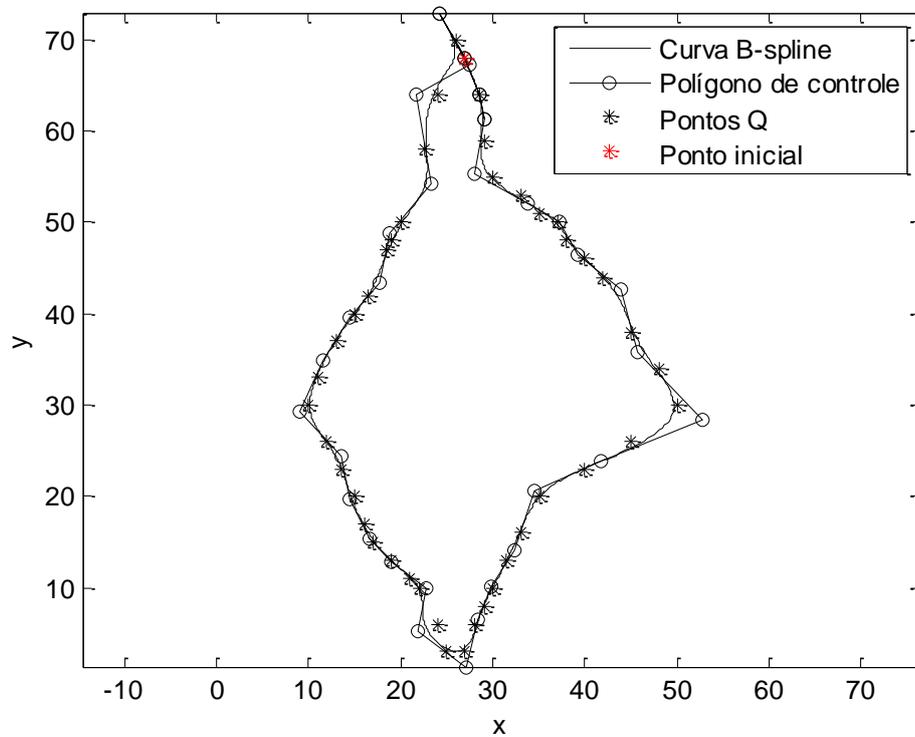
As Figuras 5.7a, 5.7b e 5.7c mostram exemplos da mesma curva para diferentes pontos iniciais, considerando a quantidade de pontos de controle definidos pela quantidade de inversão da curva, item 5.3.1, e os pesos calculados conforme item 5.3.2.



(a)



(b)



(c)

Figura 5.7 – (a), (b) e (c) Curvas B-spline para diferentes pontos iniciais.

Percebe-se que para cada ponto é obtida uma curva diferente, e em certos casos a curva não possui um bom resultado, principalmente nos pontos próximos aos pontos iniciais.

Assim, foram analisadas todas as curvas começando em cada ponto Q_k ($k = 0, \dots, j$) para verificar qual a melhor relação dos pontos iniciais em que a curva obtida possuía o melhor resultado. Verificou-se que as melhores curvas eram obtidas quando os pontos iniciais estavam mais alinhados com o ponto anterior e posterior ao ponto analisado.

O alinhamento foi definido pelo ângulo interno de um triângulo formado a cada três pontos de Q_k . Seja a distribuição de pontos conforme a Fig. 5.8 e tendo como exemplo os pontos em vermelho.

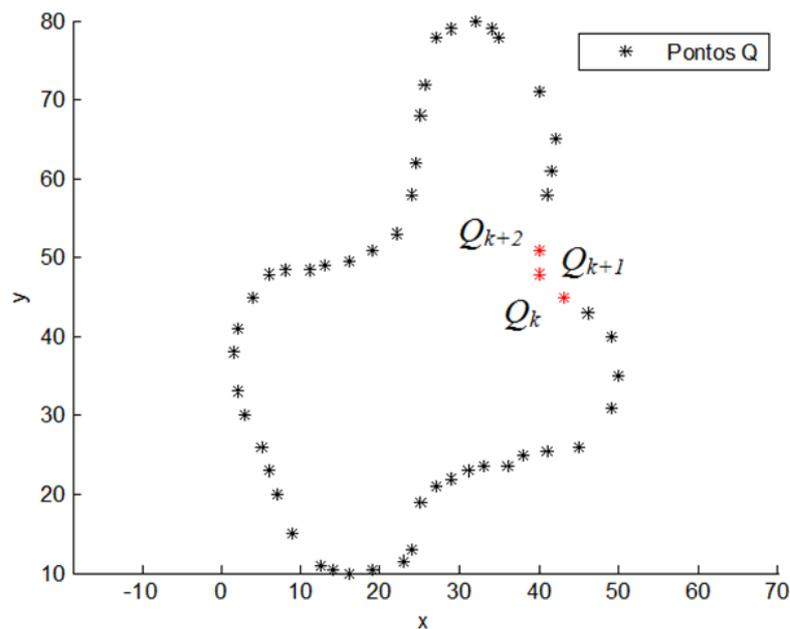


Figura 5.8 – Pontos Q_k para a definição do ponto inicial

Tem-se os pontos Q_k , Q_{k+1} e Q_{k+2} formando um triângulo de lados a , b e c , e ângulos A , B e α conforme Fig. 5.9.

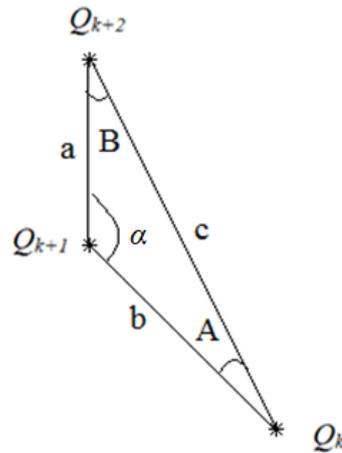


Figura 5.9 – Triângulo formado por Q_k , Q_{k+1} e Q_{k+2} . Cálculo do ângulo formado pelos segmentos $\overline{Q_k Q_{k+1}}$ e $\overline{Q_{k+1} Q_{k+2}}$

Usa-se a Lei dos Cossenos para obter o valor do ângulo α :

$$c^2 = a^2 + b^2 - 2ab \cos(\alpha) \quad (5.7)$$

Manipulando a Eq. (5.7) tem-se:

$$\alpha = \arccos\left(\frac{-c^2 + a^2 + b^2}{2ab}\right) \quad (5.8)$$

É feito esse processo a cada três pontos consecutivos. Assim, para $j+1$ pontos, tem-se $j+1$ valores de ângulos. Quanto mais próximo de 180° os valores dos ângulos, mais próximos de uma reta estão os pontos.

Para definir qual o valor do ângulo que possa ser considerado próximo a 180° , primeiramente é feito a média e o desvio padrão dos ângulos (α):

$$\bar{\alpha} = \frac{1}{j} \sum_{i=0}^j \alpha_i \quad (5.9)$$

$$\sigma = \frac{\sum_{i=0}^j (\alpha_i - \bar{\alpha})^2}{j - 1} \quad (5.10)$$

Assim, adotou-se que todos os ângulos maiores que $\bar{\alpha} + \sigma/2$ são considerados próximos de 180° , portanto, estão mais alinhados. Para definir a relação anterior ($\bar{\alpha} + \sigma/2$), foram feitas simulações a fim de obter a melhor relação para que a mesma apresentasse um resultado mais satisfatório.

Seja o seguinte conjunto de pontos que definem uma curva:

$Q = \{(30,10) (31.5,13) (33,16) (35,20) (40,23) (45,26) (50,30) (48,34) (45,38) (42,44) (40,46) (38,48) (37,50) (35,51) (33,53) (30,55) (29,59) (28.5,64) (27,68) (26,70) (24,64) (22.5,58) (20,50) (19,48) (18.5,47) (16.5,42) (15,40) (13,37) (11,33) (10,30) (12,26) (13.5,23) (15,20) (16,17) (17,15) (19,13) (21,11) (22,10) (24,6) (25,3) (27,3) (28,6) (29,8)\}$

Usando a Eq. (5.8) a cada três pontos consecutivos, obtém-se os ângulos, em graus, para cada ponto:

$\alpha = \{180 \ 180 \ 180 \ 147.5288 \ 180 \ 172.3039 \ 102.0948 \ 169.6952 \ 169.6952 \ 161.5651 \ 180 \ 161.5651 \ 143.1301 \ 161.5651 \ 168.6901 \ 137.7263 \ 171.6743 \ 165.1545 \ 173.9910 \ 45 \ 175.6013 \ 176.6822 \ 170.7890 \ 180 \ 175.2364 \ 164.9315 \ 176.8202 \ 172.8750 \ 171.8699 \ 135 \ 180 \ 180 \ 171.8699 \ 171.8699 \ 161.5651 \ 180 \ 180 \ 161.5651 \ 171.8699 \ 108.4349 \ 108.4349 \ 171.8699 \ 180\}$

Pelas Eqs. (5.9) e (5.10), tem-se:

$$\bar{\alpha} = 162.2945 \text{ e } \sigma = 26.9321$$

Assim, $\bar{\alpha} + \sigma/2 = 175.7606$. Portanto, todos os ângulos maiores que 175.7606° são considerados próximos a 180° , e os pontos relacionados a esses ângulos são considerados alinhados. Tais pontos, são:

$Q_1=(30,10)$, $Q_2=(31.5,13)$, $Q_3=(33,16)$, $Q_5=(40,23)$, $Q_{11}=(40,46)$, $Q_{22}=(22.5,58)$, $Q_{24}=(19,48)$, $Q_{27}=(15,40)$, $Q_{31}=(12,26)$, $Q_{32}=(13.5,23)$, $Q_{36}=(19,13)$, $Q_{37}=(21,11)$, $Q_{43}=(29,8)$.

e estão representados na Fig. 5.10.

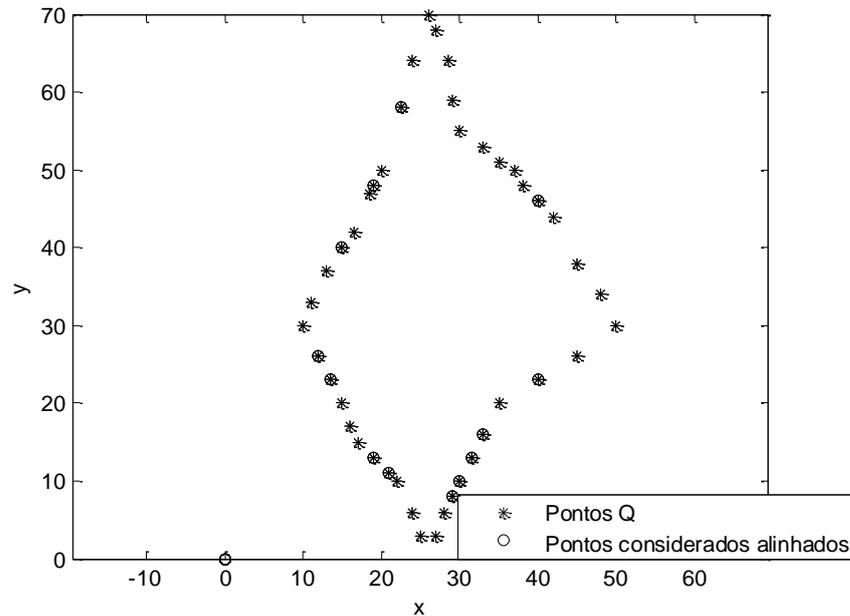


Figura 5.10 - Pontos considerados alinhados

Dentre os pontos considerados alinhados, ainda é necessário definir qual deles deve ser o ponto inicial para que se obtenha a melhor curva.

Para cada ponto desse foi verificada a direção da sua tangente, que é definida pela derivada do ponto usando a Eq. (5.11)

$$C'(0) = \frac{p}{u_{p+1}} (Q_1 - Q_0) \quad (5.11)$$

Para cada ponto considerado alinhado é traçada uma B-spline cúbica fechada conforme descrito no item 5.2. Nesta curva ficam definidos os pontos de controle P_1 e P_2 (segundo e terceiro pontos de controle), bem como a tangente definida por Q_0 . Esses dados são utilizados para o cálculo das distâncias d_{t1} e d_{t2} , conforme esquematizado na Fig. 5.11.

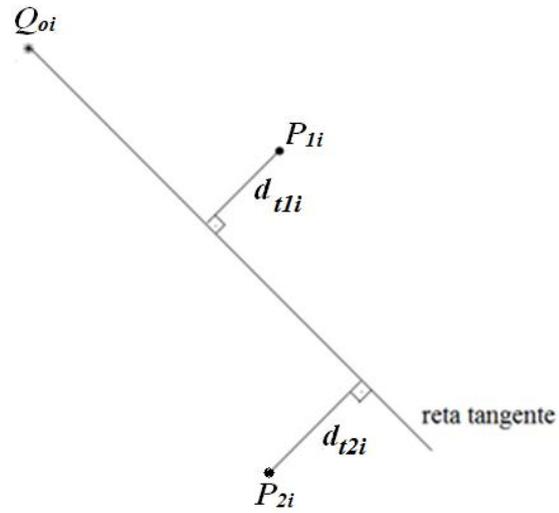


Figura 5.11 – Distâncias entre os pontos de controle e a reta tangente

O ponto em que a soma dessas duas distâncias, $d_t = d_{t1} + d_{t2}$, for a menor, é considerado o ponto inicial, pois a B-spline estará mais alinhada com os pontos que definem a curva.

A Tabela 5.1 mostra os pontos definidos como alinhados do exemplo anterior, as distâncias d_{t1} e d_{t2} e a sua soma.

Tabela 5.1 – Pontos considerados alinhados e distâncias d_{t1} e d_{t2}

Pontos considerados alinhados	$d_{t1}(mm)$	$d_{t2}(mm)$	$d_t = d_{t1} + d_{t2}(mm)$
(30,10)	2.8865	1.3245	4.2110
(31.5,13)	5.0157	2.6420	7.6577
(33,16)	17.6699	8.9258	26.5957
(40,23)	14.9533	6.0299	20.9832
(40,46)	10.8445	2.9515	13.7961
(22.5,58)	40.3196	3.8563	44.1758
(19,48)	1.6492	3.2765	4.9257
(15,40)	12.7493	5.4691	18.2184
(12,26)	4.6527	1.8931	6.5458
(13.5,23)	1.2870	0.8595	2.1465
(19,13)	12.7321	3.3305	16.0627
(21,11)	6.2034	8.6913	14.8947
(29,8)	1.3480	0.7959	2.1439

A menor soma das distâncias é a referente ao ponto (29,8). Portanto, esse é o ponto inicial para o exemplo dado, Fig. 5.12.

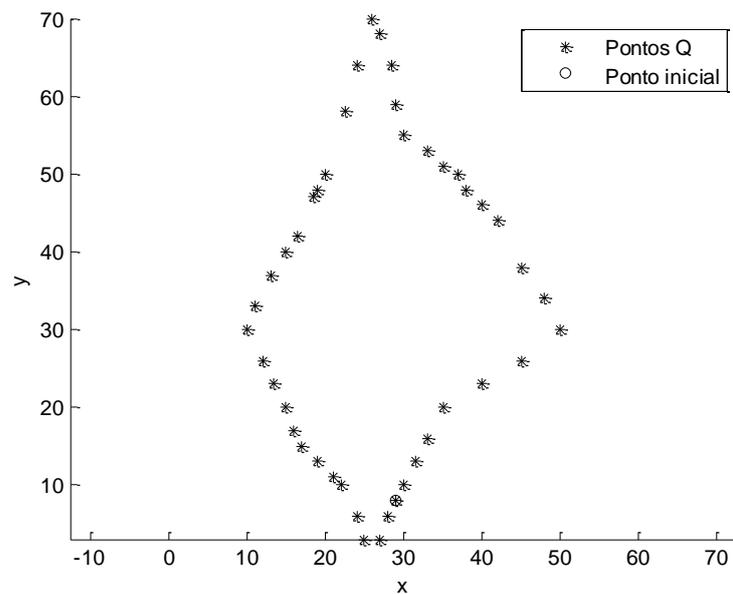


Figura 5.12 – Ponto inicial considerando o alinhamento entre os pontos dados

Após obtido o melhor ponto inicial, várias curvas obtidas apresentavam uma alteração próximo a ele, Fig 5.13, cujo zoom da região inicial é apresentado na Fig. 5.14.

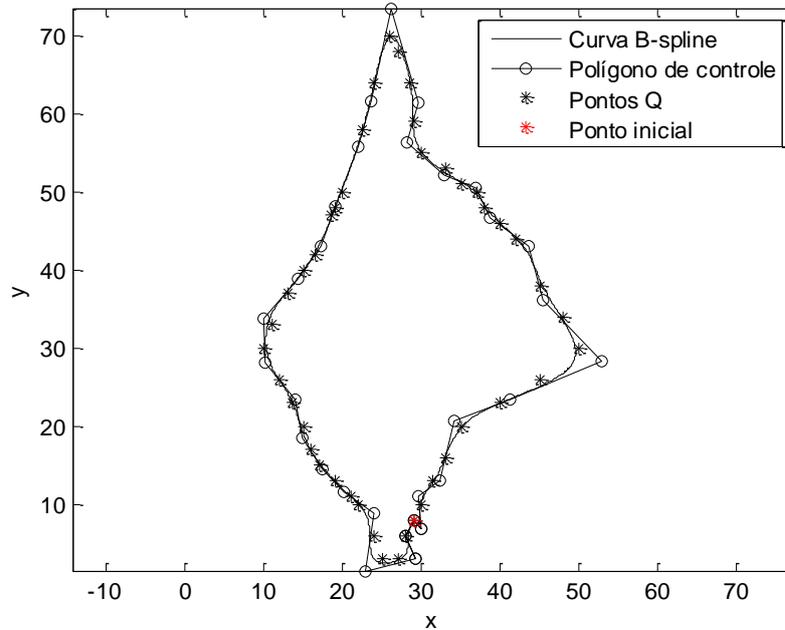


Figura 5.13 – Curva B-spline com alteração na região inicial considerando a direção da tangente

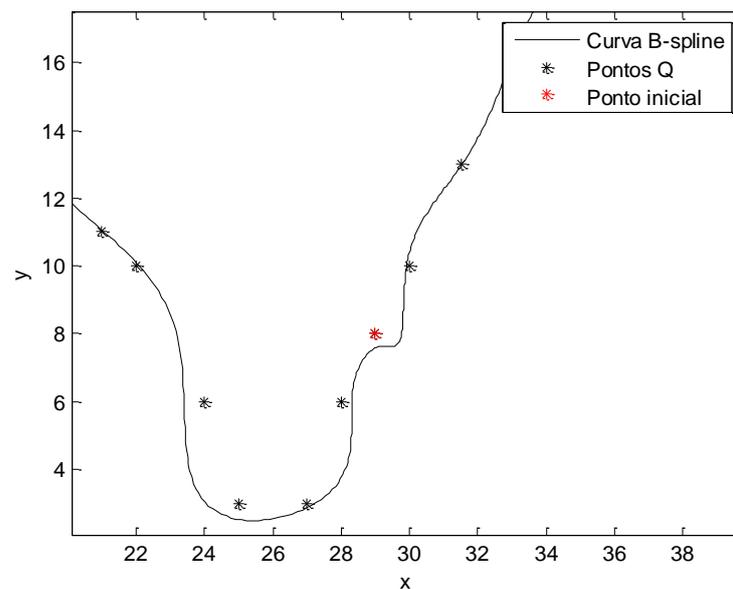


Figura 5.14 – Zoom da região inicial considerando a direção da tangente

Para resolver o problema dessa alteração da curva, foi utilizada a mesma tangente do primeiro ponto, definida pela Eq. (5.11). Tendo a direção dessa tangente foi imposto que os pontos de controle P_1 e P_2 , já calculados, deveriam estar sobre ela. Como a distância entre esses pontos (reta normal à tangente que passa por cada ponto de controle) e a reta tangente já estavam calculadas, foram encontrados os pontos de interseção das mesmas. Esses pontos são, então, os novos pontos de controle. A Figura 5.15 ilustra a curva com os pontos de controle modificados e a Fig. 5.16 mostra o zoom na região.

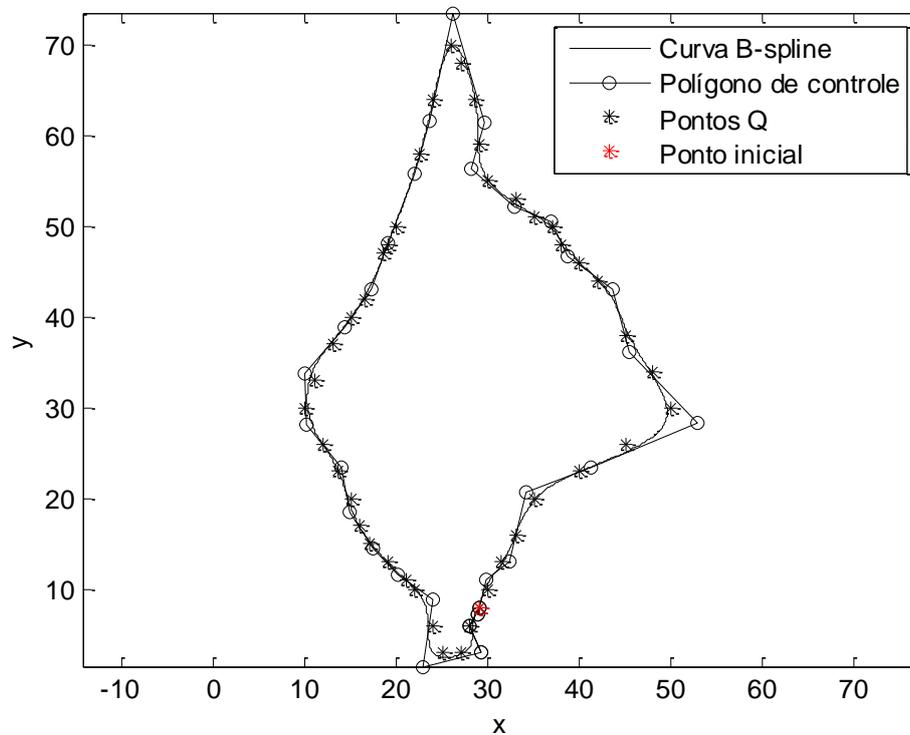


Figura 5.15 – Curva B-spline com os pontos de controle P_1 e P_2 modificados

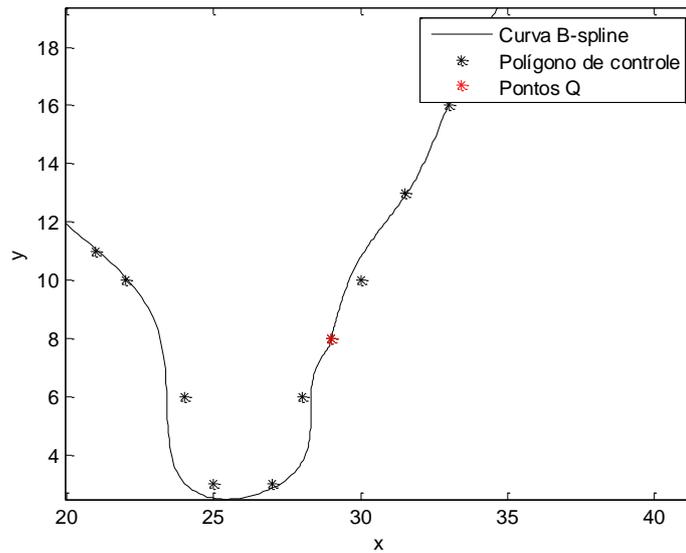


Figura 5.16 – Zoom da região inicial com os pontos de controle P_1 e P_2 modificados

É perceptível, pela Fig. 5.16, que a imposição dos pontos de controle P_1 e P_2 sobre a direção da tangente melhorou a curva B-spline na região inicial, eliminando a alteração verificada na Fig. 5.14.

5.3.4 Ajuste da quantidade dos pontos de controle

Mesmo com os ajustes apresentados nos itens anteriores, várias curvas apresentam desvios que não podem ser considerados adequados na representação da seção do objeto a partir de seus pontos, conforme pode ser observado na Fig. 5.3.

Foi verificado por simulações que, quando a quantidade de pontos de inversão é inferior a 50% dos pontos dados, a curva geralmente apresenta estes desvios. Então, para melhorar o resultado nesses casos, é adotado que a quantidade de pontos de controle é a metade da quantidade de pontos Q_k ($k = 0, \dots, j$).

As Figuras 5.17 e 5.18 mostram a mesma curva para uma quantidade de pontos de controle definido pelos pontos de inversão de curvatura e para a quantidade igual a 50% da quantidade de pontos Q_k ($k = 0, \dots, j$), respectivamente.

A quantidade de pontos dados, Q_k ($k = 0, \dots, j$), é 74 e para o exemplo da Fig. 5.3 e a quantidade de pontos de inversão de curvatura é 17, o que é menor que a metade de pontos Q_k ($k = 0, \dots, j$). Para esses casos, como antes mencionado, é adotado que a quantidade de pontos de controle deve ser imposta a 50%, sendo então, no exemplo da Fig. 5.18 uma

quantidade de 37. A Figura 5.3 é apresentada novamente, junto com a Fig. 5.18 para melhor visualização.

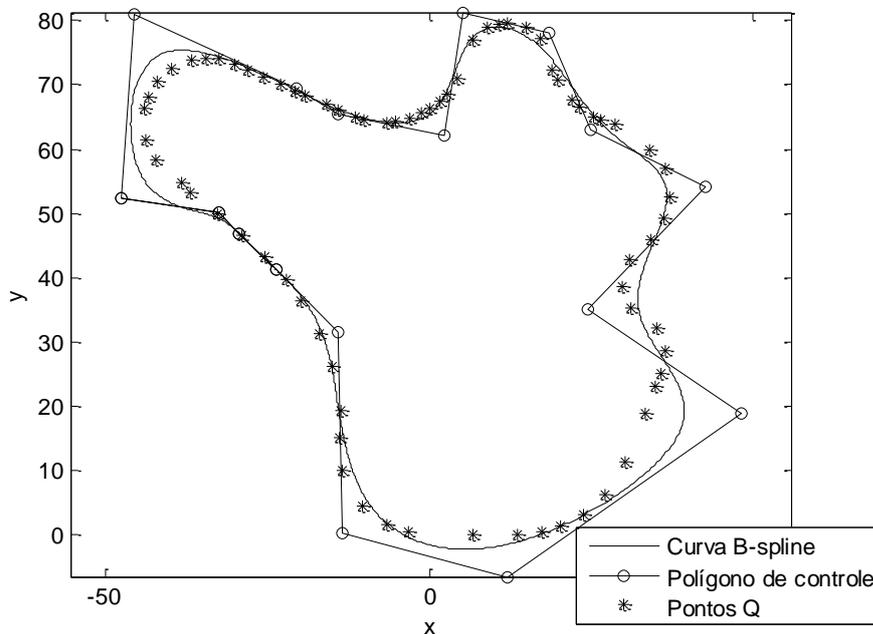


Figura 5.17 – Repetição da Figura 5.3

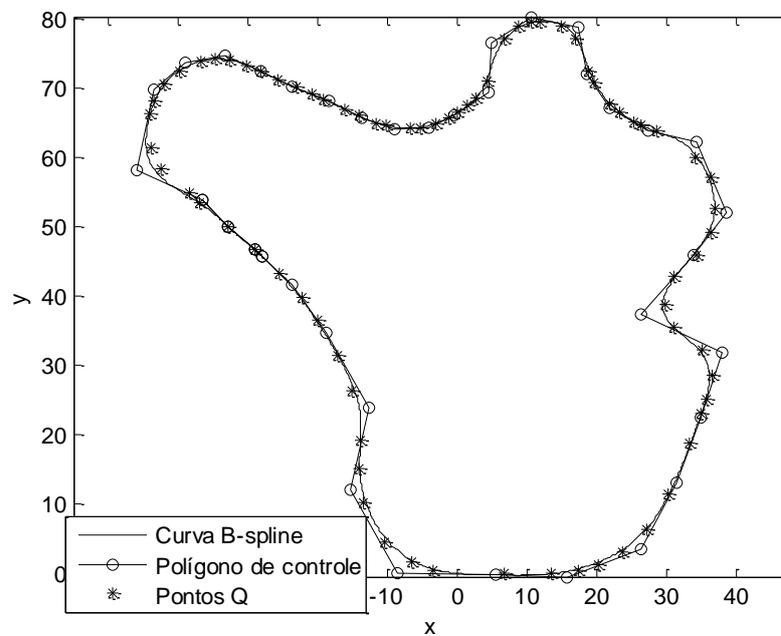


Figura 5.18 – Curva obtida considerando a quantidade de pontos de controle igual a 50% dos pontos dados, os pesos e a definição do ponto inicial conforme apresentado

Pode-se perceber que a curva traçada tem resultado muito melhor para uma quantidade de pontos de controle maior.

5.4 Aplicação da B-spline nas seções transversais

Para efetuar a aproximação da B-spline fechada pelo método dos mínimos quadrados com peso é necessário que se tenha os pontos que representam as coordenadas do objeto, a quantidade de pontos de controle, os pesos associados a cada ponto, a definição do ponto inicial e dos ajustes da curva, e o grau da curva. Com tudo já definido é possível então traçar a curva a fim de obter uma representação mais fiel possível da seção transversal do objeto.

A seguir são apresentados exemplos da curva B-spline aplicada em diferentes seções transversais. Em todos os casos o grau adotado para a curva é 3.

Para todos os exemplos a seguir, as coordenadas dos pontos Q_k ($k = 0, \dots, j$), os pesos e os pontos de controle encontrados estão no Apêndice VI, e o programa desenvolvido para os cálculos e plotagem da curva está no Apêndice VII.

5.4.1 Exemplo 1

Neste exemplo a curva B-spline deve se aproximar dos pontos representados na Fig.5.19. Em todos os exemplos o ponto inicial já foi definido de acordo com o item 5.3.3.

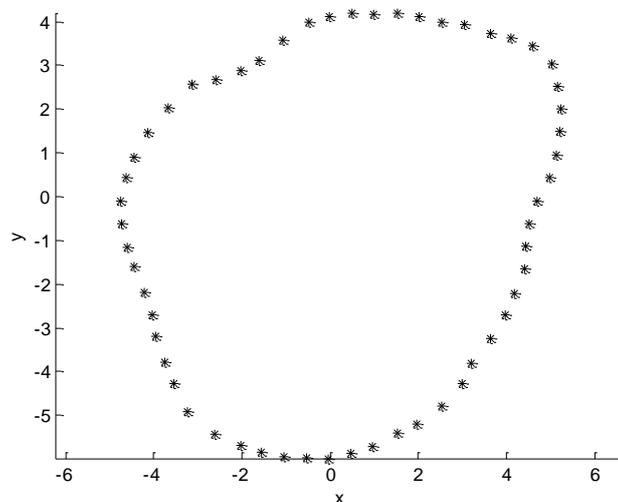


Figura 5.19 – Pontos conhecidos que definem a seção do objeto - exemplo 1

A Figura 5.20 ilustra a B-spline cúbica para os pontos dados sem os ajustes necessários. Ou seja, os pontos estão sem pesos associados a eles, a quantidade de pontos de controle é inferior a 50% da quantidade de $Q_k (k = 0, \dots, j)$, e os pontos P_1 e P_2 não foram impostos sobre a tangente.

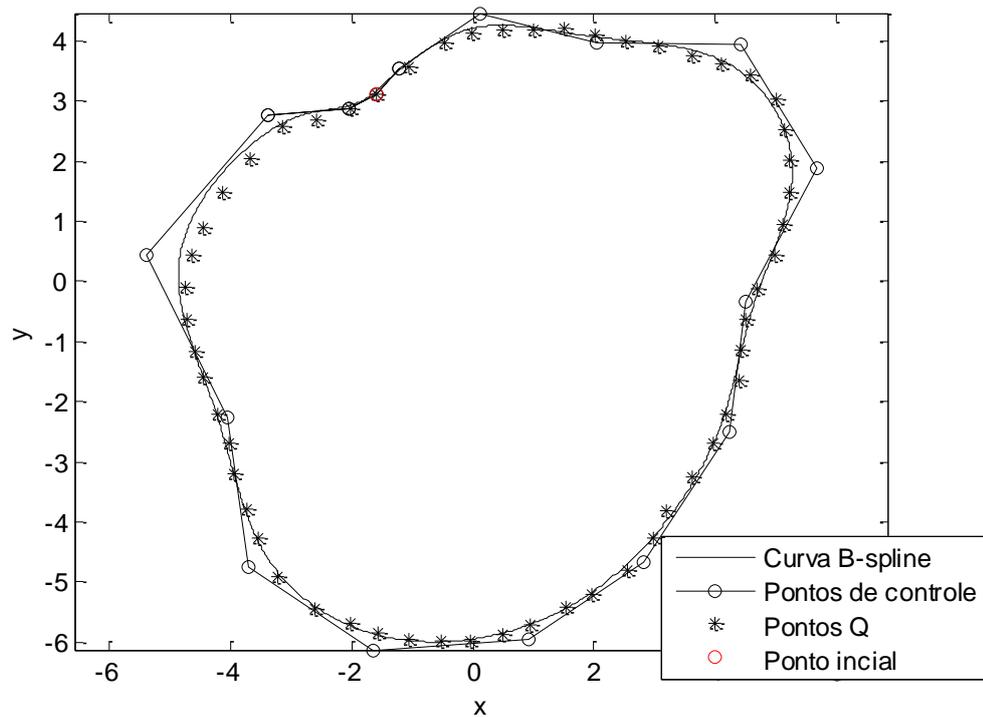
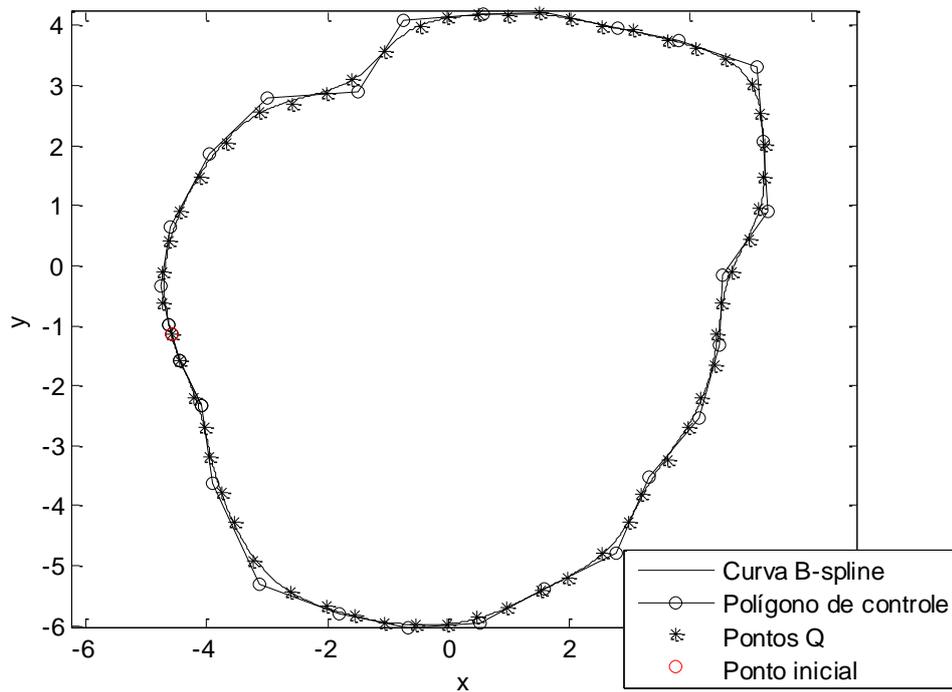
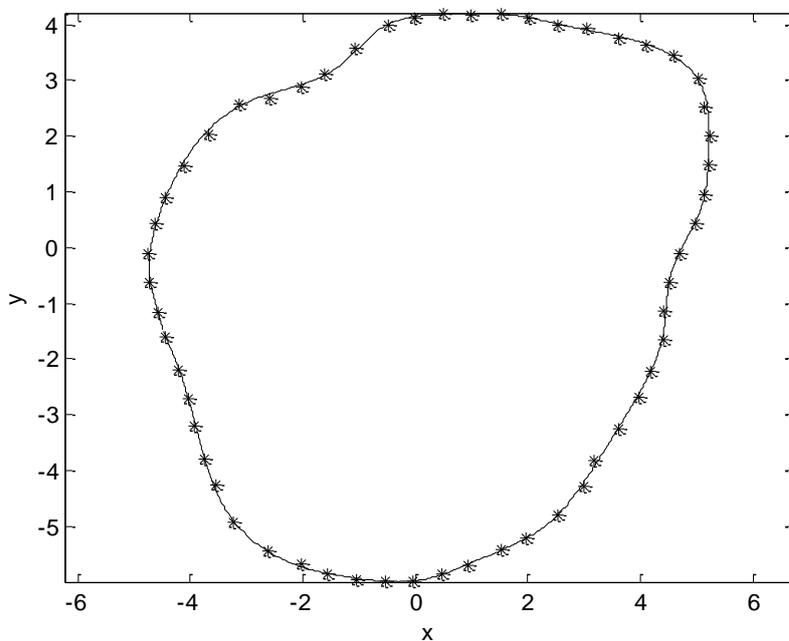


Figura 5.20 – Curva B-spline sem ajustes (exemplo1)

A curva B-spline final traçada está representada nas Figs. 5.21a e 5.21b. Nesse caso a curva foi obtida após serem realizados todos os ajustes necessários para uma melhora do resultado da mesma. Ajustes esses que são: pesos associados aos pontos, pontos de controle definidos como no mínimo 50% dos valores de $Q_k (k = 0, \dots, j)$, e os pontos de P_1 e P_2 definidos sobre a tangente.



(a)



(b)

Figura 5.21 - Curva B-spline cúbica final para os dados do exemplo 1. (a) Curva B-spline com os pontos que a definem e os pontos de controle; (b) Curva B-spline com pontos e sem os pontos de controle

Pelas Figuras 5.21a e 5.21b, verifica-se que a curva B-spline obtida tem um resultado bastante satisfatório, principalmente em comparação com a Fig. 5.20, pois ela aproximou-se ao máximo dos pontos dados, representando de forma fiel a seção.

5.4.2 Exemplo 2

Neste exemplo os pontos utilizados estão representados na Fig. 5.22:

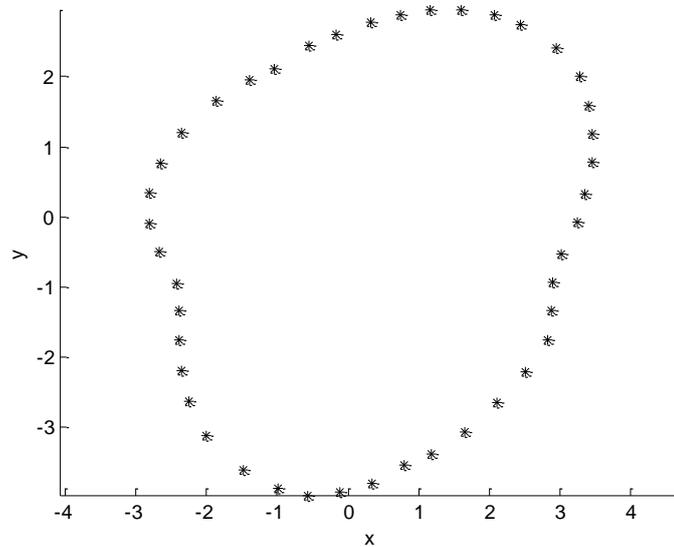


Figura 5.22 – Pontos conhecidos que definem a seção do objeto - exemplo 2

A curva da Fig. 5.23 representa a B-spline cúbica sem ajustes traçada para o exemplo 2.

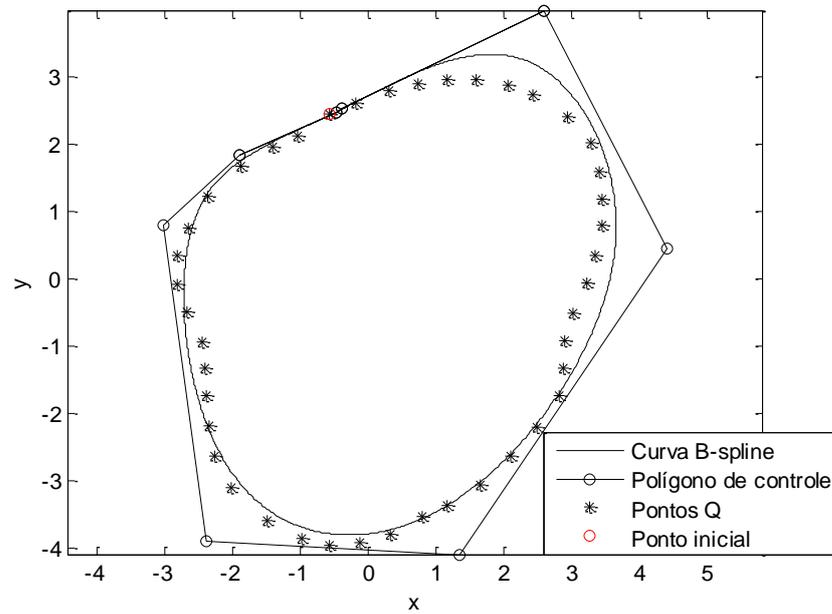
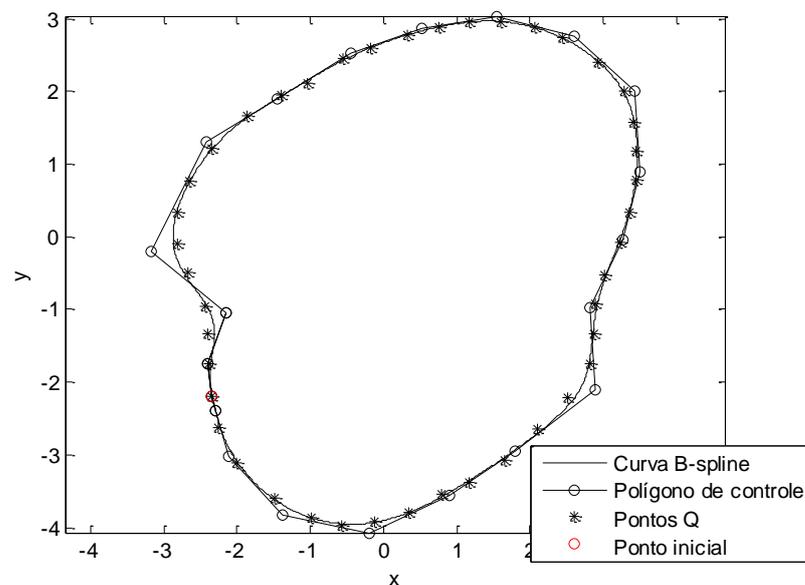


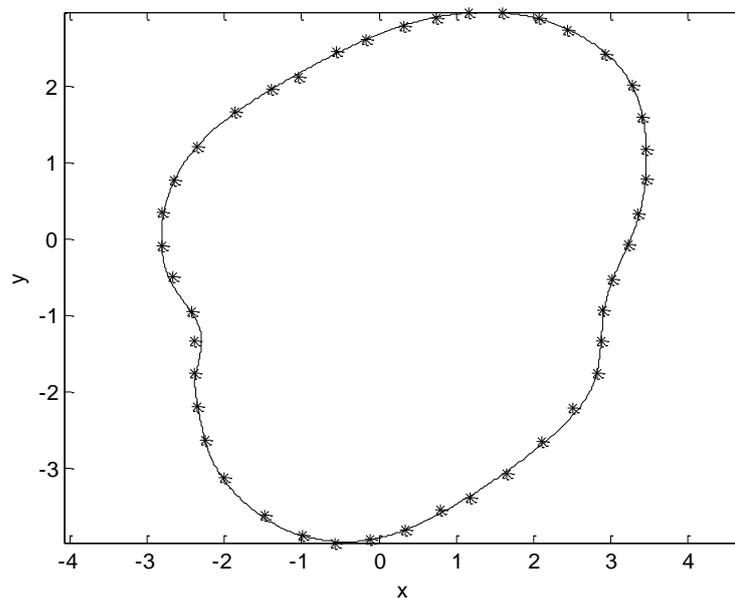
Figura 5.23 - Curva B-spline sem ajustes (exemplo2)

Pela Figura 5.23, pode-se perceber que a curva B-spline sem ajustes não possui um bom resultado, já que em determinadas regiões a curva não acompanha os pontos, tendo um perfil diferente do desejado.

Assim, a curva B-spline, com os ajustes necessários, é traçada e representada nas Figs.5.24a e 5.24b.



(a)



(b)

Figura 5.24 - Curva B-spline cúbica final para os dados do exemplo 2. (a) Curva B-spline com pontos que a definem e os pontos de controle; (b) Curva B-spline com pontos e sem os pontos de controle

As Figuras 5.24a e 5.24b mostram que a curva B-spline com ajustes teve uma melhora significativa comparada à curva da Fig. 5.23, representando melhor o perfil do objeto.

5.4.3 Exemplo 3

Neste exemplo os pontos utilizados são os da Fig. 5.25:

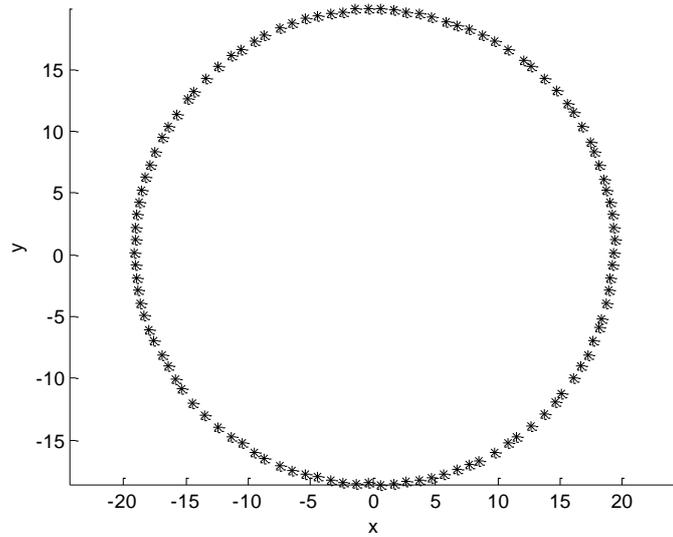


Figura 5.25 – Pontos conhecidos que definem a seção do objeto - exemplo 3

A curva B-spline cúbica, sem os ajustes, é representada na Fig. 5.26.

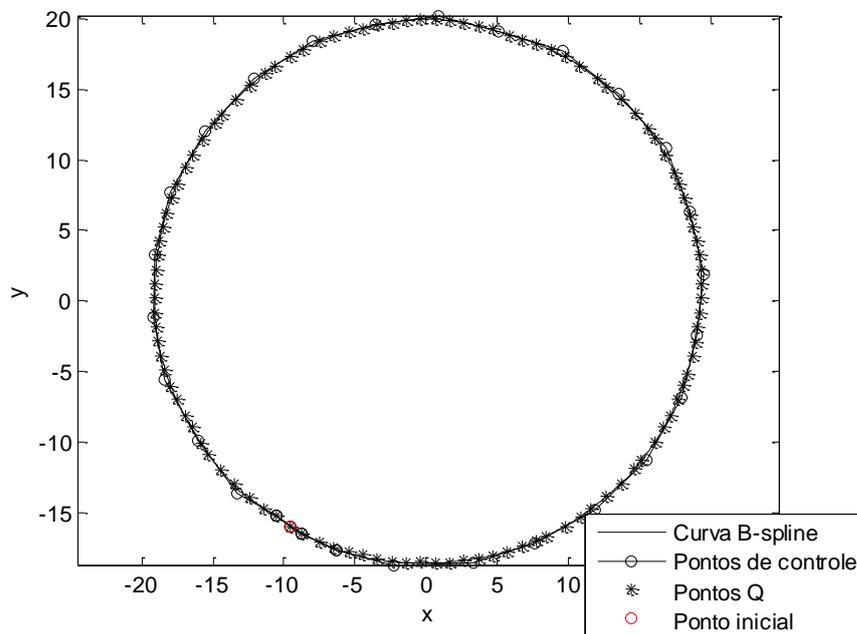
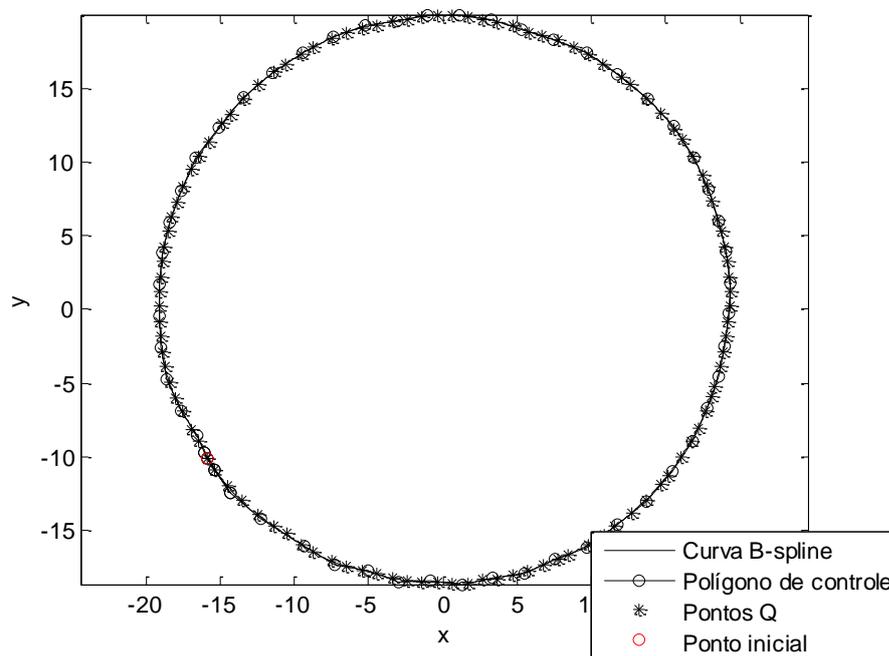
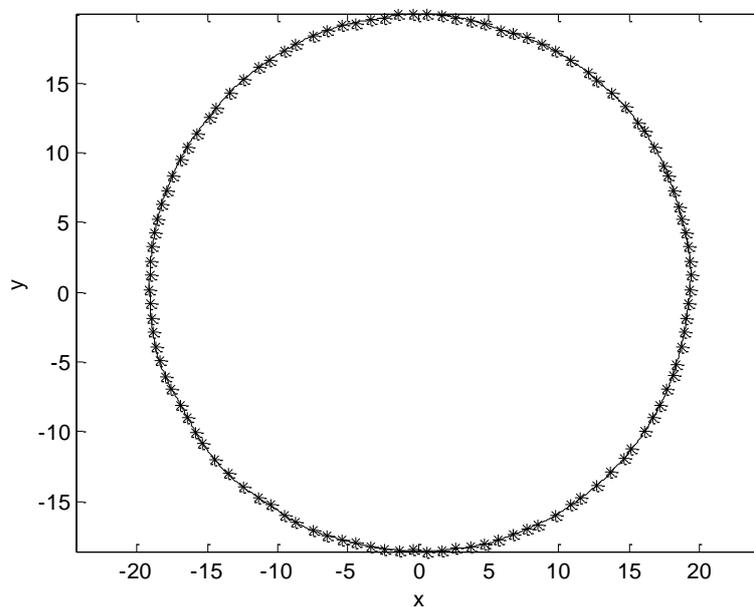


Figura 5.26 - Curva B-spline sem ajustes (exemplo 3)

E a curva B-spline cúbica, com os ajustes necessários, traçada para esse exemplo está representada nas Figs. 5.27a e 5.27b.



(a)



(b)

Figura 5.27 – Curva B-spline cúbica final para os dados do exemplo 3. (a) Curva B-spline com os pontos que a definem e os pontos de controle; (b) Curva B-spline com pontos e sem os pontos de controle

Como visto na Fig. 5.27, a curva B-spline também possui bons resultados em seções circulares.

5.4.4 Exemplo 4

Neste exemplo os pontos utilizados são os da Fig. 5.28:

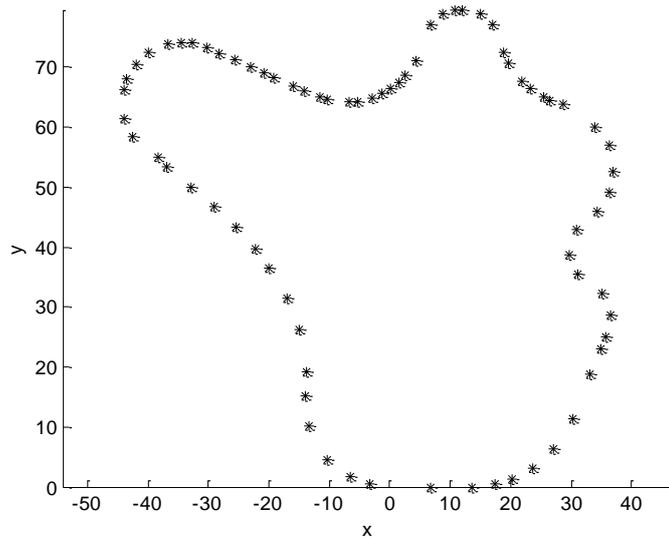


Figura 5.28 – Pontos conhecidos que definem a seção do objeto - exemplo 4

A curva B-spline cúbica, sem os ajustes, para esse exemplo é representada na Fig. 5.29.

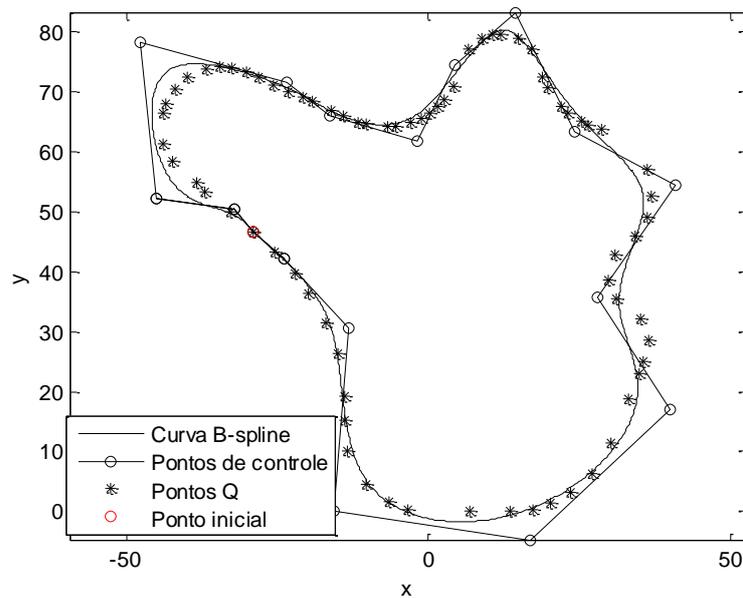
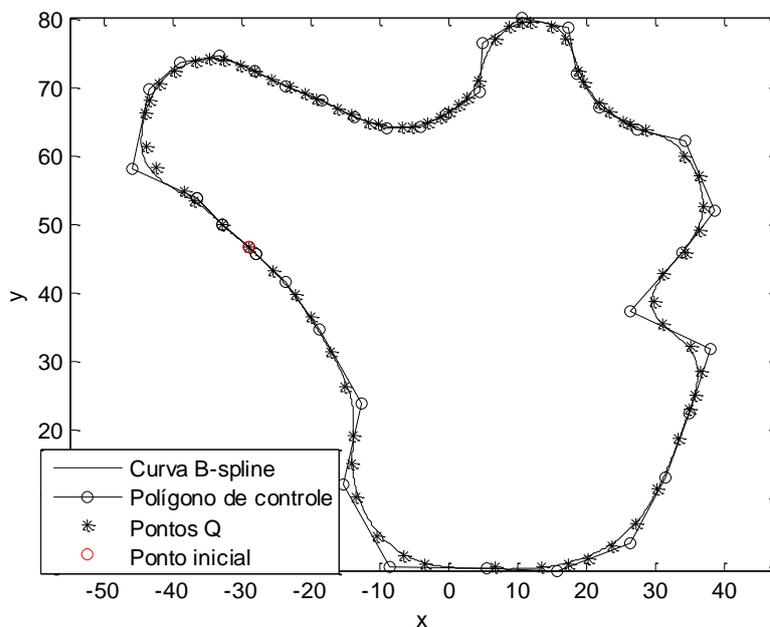
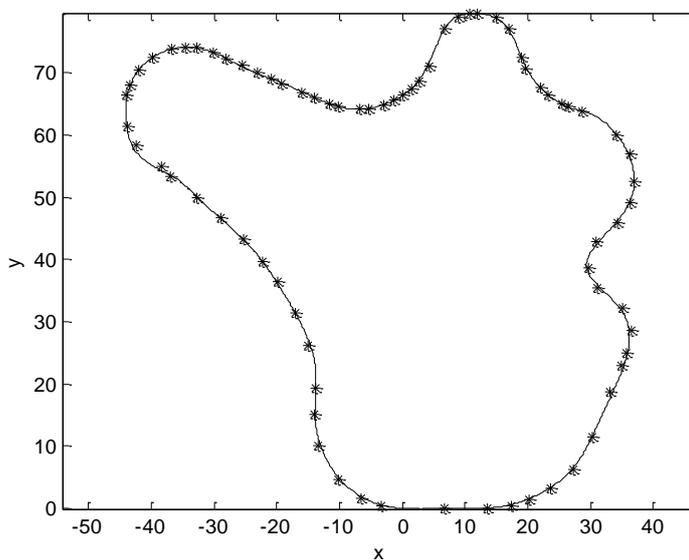


Figura 5.29 - Curva B-spline sem ajustes (exemplo 4)

E a curva B-spline cúbica final traçada para esse exemplo está representada nas Figs. 5.30a e 5.30b.



(a)



(b)

Figura 5.30 – Curva B-spline cúbica final para os dados do exemplo 4. (a) Curva B-spline com os pontos que a definem e os pontos de controle; (b) Curva B-spline com pontos e sem os pontos de controle

Na Figura 5.30 é possível perceber, mais uma vez, que a curva B-spline traçada depois dos ajustes feitos, possui um melhor resultado. E é perceptível que, para todos os exemplos dados, a curva B-spline obtida teve resultado satisfatório, aproximando ao máximo dos pontos dados.

5.5 Resumo da metodologia de ajuste de curvas fechadas utilizando B-spline com pesos

A metodologia proposta, para o ajuste de curvas fechadas a partir de um conjunto de pontos conhecidos, que definem uma seção de um objeto, utilizando B-spline com pesos, pode ser resumida nas seguintes etapas:

1 – Traçado de uma B-spline aberta do tipo “*clamped*” considerando:

- $Q_k (k = 0, \dots, j)$ como pontos conhecidos;
- Cálculo de $\bar{u}_k (k = 0, \dots, j)$ pelo método do comprimento de corda, Eq. (3.41);
- Definição da quantidade de pontos de controle pela quantidade de pontos $Q_k (k = 0, \dots, j)$ que invertem a curvatura, item 5.3.1;
- Cálculo de $U = \{u_0, \dots, u_m\}$ pelo método das Eqs. (3.52) e (3.53);
- Definição dos pesos $w_k (k = 0, \dots, j)$ pela relação entre os raios das circunferências formadas a cada três pontos consecutivos de $Q_k (k = 0, \dots, j)$, item 5.3.2;
- Cálculo dos pontos de controle pelo método dos mínimos quadrados, Eq. (3.48).

2 – Fechamento da B-spline cúbica aberta considerando:

- Cálculo de um novo vetor nó do tipo “*unclamped*” pelas Eqs. (3.54) e (3.55);
- Acréscimo de $p+3$ nós de modo que os seus vãos sejam iguais aos $p+3$ primeiros vãos do vetor nó definido pelas Eqs. (3.54) e (3.55);
- Repetição dos $p+1$ primeiros pontos de controle encontrados pela Eq. (3.48);
- Traçado da curva cúbica B-spline fechada.

3 – Ajustes na curva:

- Definição do ponto inicial da curva através dos pontos considerados mais alinhados, item 5.3.3;
- Imposição de alguns pontos de controle sobre a tangente do ponto inicial, item 5.3.3.

- Definição dos pontos de controle como 50% dos pontos $Q_k (k = 0, \dots, j)$ para os casos em que a quantidade de pontos de inversão de curvatura é menor que esse valor, item 5.3.4;

Depois de todas as etapas concluídas é traçada a curva B-spline cúbica fechada com peso.

5.6 Conclusão

Escolher o tipo de curva e o método a ser utilizado é o primeiro passo para a reprodução de uma seção. A B-spline foi escolhida por possuir características como: controle local da curva, no qual a curva pode ser ajustada apenas em pontos necessários e resultado mais direto com tempo de processamento mais rápido. O método dos mínimos quadrados com peso possibilita a curva B-spline aproximar o máximo possível dos pontos oriundos da digitalização, pois tal método implica que o erro entre a curva obtida e a curva original deva ser o menor possível. Além disso, o peso relacionado a cada ponto que representa a coordenada do objeto, oferece à curva uma flexibilidade a mais, pois com esses pesos é possível aproximar ou afastar a curva dos pontos, possibilitando ao construtor definir de qual maneira ele pretende fazer isso.

Para a reprodução de uma seção transversal fechada é preciso definir a curva B-spline fechada, calculando de forma adequada os pontos de controle e os nós, de tal forma que ela possua suavidade no fechamento não deixando evidente onde está esse local.

Traçar a B-spline cúbica diretamente não gera o melhor resultado. Por isso, é necessário definir vários fatores. Além do peso, já mencionado, deve-se definir qual o melhor ponto para se iniciar o traçado da curva; qual a quantidade mais adequada de pontos de controle e, depois de traçada a curva, ainda fazer ajustes como a imposição de alguns pontos de controle sobre a direção da tangente do primeiro ponto, como nesse trabalho. Todos esses requisitos são utilizados para que se obtenha um melhor resultado da curva, ou seja, para que ela se aproxime ao máximo dos pontos dados.

Em todos os exemplos apresentados nesse capítulo, a curva obtida, depois de todos os ajustes e definições consideradas, teve um resultado satisfatório. Em todos os casos a curva representou de forma fiel a seção transversal.

CAPÍTULO VI

CONCLUSÕES

6.1 Conclusões

A digitalização de perfis tridimensionais de objetos e sua reconstrução têm apresentado um grande desenvolvimento e podem ser aplicadas em diversas áreas tais como na fabricação (fundição, estereolitografia, etc.), na engenharia reversa (reconstrução de componentes de máquinas que não possuem desenhos), na inspeção (permitindo verificar se o objeto está conforme o projetado), simulação virtual (efeitos especiais no cinema, jogos), na cópia de peças disponíveis em museus, na medicina (reconstrução de partes do corpo humano), na exploração científica e no mercado de consumo. Essa digitalização origina uma nuvem de pontos, em geral desorganizados que devem ser reduzidos, organizados e tratados com o intuito de preservar ao máximo o contorno do objeto. Para reconstruir a superfície através dessas amostragens são requeridos algoritmos eficientes e confiáveis. Essa reconstrução pode ser feita a partir da reconstrução das seções transversais que depois de unidas permitem representar a superfície do objeto.

Para a reconstrução dessas seções transversais é necessário definir qual o método a ser utilizado. Nesse trabalho foi utilizada a curva B-spline fechada associada a método de aproximação por mínimos quadrados com peso.

A B-spline é utilizada devido suas várias características, dentre elas o controle local da curva, que permite ajustar a curva somente em determinados pontos sem alteração do restante da curva, e é cúbica devido ao fato de ser mais adequada para muitas aplicações. A curva é fechada para reproduzir seções fechadas de um determinado objeto e os pesos são implementados para a aproximação ou afastamento da curva em determinados pontos.

O método utilizado possui como ajustes, além dos pesos, a definição da quantidade de pontos de controle, do ponto inicial e a definição de alguns desses pontos de controle sobre a tangente do ponto inicial para evitar distorções da curva nessa região. Tal método pode ser utilizado para qualquer conjunto de pontos dados obtendo resultados satisfatórios sem a intervenção do usuário. Ou seja, dado um conjunto de pontos que representam a seção transversal do objeto, a curva é traçada e todos os ajustes necessários são feitos de forma automática, definindo qual o ponto inicial, a tangente, os pesos adotados e a quantidade de pontos de controle sem a necessidade de uma avaliação e/ou modificação do usuário. O método ainda pode ser utilizado para qualquer tipo de grau da curva, mas nesse caso foi adotado o grau 3 devido às suas vantagens antes mencionadas.

A curva B-spline cúbica utilizada representou de forma fiel a seção transversal do objeto, mostrando através dos exemplos, que ela é um reconstrutor de forma eficiente.

6.2 Sugestões para trabalhos futuros

Como sugestão para trabalhos futuros está definir a superfície do objeto através das seções transversais reproduzidas nesse projeto.

Para a reprodução da superfície é feita a ligação entre os pontos que estão em diferentes planos produzindo uma malha bidirecional. Para obter as curvas longitudinais poderão ser utilizados os mesmos pontos de controle obtidos para as seções transversais do objeto. Esse procedimento facilita o processamento matemático e computacional.

Em princípio, se as seções transversais construídas correspondem “fielmente” às seções transversais do objeto, a superfície também o será. No entanto, sabe-se que isso depende da distância utilizada entre as seções transversais. Nesse sentido podem ser feitas análises para a redução de pequenas distorções que porventura ocorrerem, fazendo-se ajustes no procedimento de ligação entre os pontos de controle e no distanciamento das seções transversais.

Além disso, pode-se sugerir como trabalho futuro definir a métrica para a verificação quantitativa do erro entre os pontos digitalizados, que representam a seção original, e a curva B-spline obtida.

REFERÊNCIAS BIBLIOGRÁFICAS

ALVES, A. S. **Metrologia Geométrica**. Fundação Calouste Gulbenkian, 1996. 270 p.

ARANA FERNANDEZ, E.; BUITRAGO, E.; BENET, F.; TOBARRA E.; Tomografía Computarizada: Introducción a las Aplicaciones Dentales. **RCOE**. Valencia, España, v.11, n.3, p. 311-322, may. 2006.

AQUINO, D. M.; FONSECA, J. N. Jr.; CARVALHO, J. C. M.; Three-Dimensional Surface Reconstruction Using NURBS. In: 20th INTERNATIONAL CONGRESS OF MECHANICAL ENGINEERING (COBEM 2013), 2009, Porto Alegre, Rio Grande do Sul. p. 15 -20.

BAJAJ, C. L.; BERNARDINI, F.; XU, G.; Automatic Reconstruction of Surfaces and Scalar Fields from 3D Scans. In: PROCEEDINGS OF SIGGRAPH95 ON COMPUTER GRAPHICS PROCEEDINGS, 1995, Los Angeles, California., **ACM SIGGRAPH**, p. 109-118.

BARBARINI, L. H. M.; **Síntese de Cascos de Embarcações Através de Métodos de Otimização Aplicados a Curvas B-spline**. 2007. 142 f. Dissertação de Mestrado em Engenharia Mecatrônica - Escola Politécnica da Universidade de São Paulo, São Paulo, São Paulo.

CALIÒ, F.; MIGLIO, E.; MORONI, G.; RASELLA, M.; An Improved B-spline Approach for the Surfaces Reconstruction from Data Measured by CMM. In: ASPE 2003 SUMMER TOPICAL MEETING ON COORDINATE MEASURING MACHINE, 2003, Charlotte, North Carolina USA, p. 21-26.

CARVALHO, J. C. M.; FONSECA Jr, J. N.; Three – Dimensional Representation a Surface Definition Using NURBS. In: 1st INT. CONGRESS UNIVERSITY – INDUSTRY COOPERATION – UNINDU, 2005, Taubaté.

CESAR, R. M. Jr.; **Reconstrução Tridimensional por Ajuste de Superfícies Paramétricas**. 1994. 142 f. Dissertação de Mestrado em Engenharia Elétrica - Faculdade de Engenharia Elétrica – Universidade Estadual de Campinas, Campinas.

CHEN, X. D.; MA, W.; PAUL, J. C.; Cubic B-spline curve approximation by curve unclamping. **Computer – Aided Design**, v. 42, n. 6, p. 523 – 534, jun. 2010.

COHEN, F. S.; HUANG, Z.; YANG, Z.; Invariant Matching and Identification of Curves Using B-Splines Curve Representation. **IEEE Transactions on Image Processing**, v. 4, n. 1, p. 1 – 10, jan. 1995.

CURLESS, B. L.; **New Methods for Surface Reconstruction from Range Images**. 1997, 189f. Tese de Doutorado em Engenharia Elétrica - Departamento de Engenharia Elétrica. Universidade de Stanford, Califórnia.

DÍAZ, J. E. R.; **Modelagem Cinemática e Dinâmica de uma Estrutura RRP+PR**. 2011. 164f. Tese de Doutorado em Engenharia Mecânica – Departamento de Engenharia Mecânica – Universidade Federal de Uberlândia.

DE BOOR, C.; **A Pratical Guide to Spline**. 1. ed. N.Y: Springer, 2000. 241 p.

DOUROS, I.; DEKKER, L.; BUXTON, B. F.; An Improved Algorithm for Reconstruction of the Surface of the Human Body from 3D Scanner Data Using Local B-spline Patches. **IEE Computer Society**, 1999. Disponível em: <
<http://www.computer.org/csdl/proceedings/mpeople/1999/0362/00/03620029-abs.html>> .
 Acessado em: 02 fev. 2012.

ECK, M.; HOPPE, H.; Automatic Reconstruction of B-Spline Surfaces of Arbitrary Topological Type. In: PROCEEDINGS OF THE 23RD ANNUAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES (SIGGRAPH 96), 1996, Nova York, USA. p. 325-334.

EVBUOMWAN, N. F.; SIVALOGANATHAN, S.; JEBB, A. A survey of Design Philosophies, Models, Methods and System. **Journal of Engineering Manufacture**. London, v. 210, n. 4, p. 301-320, 1996.

EVERETT, H. R.; **Sensors for Mobile Robots: Theory and Application**. Wellesley, MA: A K Peters, 1995. 282p.

FERREIRA, W. R. B., **Planejamento de Trajetórias Robóticas Utilizando B-splines**. 2011. 139 f. Dissertação de Mestrado em Engenharia Mecânica - Universidade Federal de Uberlândia. Uberlândia.

FISCHER, A.; MANOR A.; Utilizing Image Processing Techniques for 3D Reconstruction of Laser-Scanned Data. **CIRP Annals - Manufacturing Technology**. v. 48, n. 1, p. 99-102, 1999. Disponível em:

< <http://www.sciencedirect.com/science/article/pii/S0007850607631400> > Acessado em : 10 mai. 2013.

FONSECA, J. N. Jr.; CARVALHO, J.C.M; Modelagem Computacional para Reconstrução Gráfica de Objetos, 2007. Disponível em: < www.seer.ufu.br/index.php/horizontecientifico/article/viewFile/.../2796 >. Acessado em: 28 ago. 2012.

FRANÇA, J. G. D. M.; GAZZIRO, M. A.; IDE, A. N.; SAITO, J. H.; A 3d Scanning System Based on Laser Triangulation and Variable Field of View. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP), 2005, Genoa. v. 1. p. I-425-428.

FURFERI, R.; GOVERNI, L.; PALAI, M.; VOLPE, Y.; From Unordered Point Cloud to Weighted B-Spline - a novel PCA-based method. In: 5th WSEAS INTERNATIONAL

CONFERENCE ON COMPUTER ENGINEERING AND APLICATIONS (CEA), 2011, p. 146-151.

GÁLVEZ, A.; IGLESIAS, A.; PEY, J. P.; Iterative two-step genetic-algorithm-based method for efficient polynomial B-spline surface reconstruction. **Information Sciences**. v. 182, n. 1, p. 56-76, jan. 2012.

GLEICHER, M.; **A curve Tutorial for Introductory Computer Grafics**, Department of Computer Science, University of Wiscosin, MaYERS, R. H., MONTGOMERY, D. C., **Response Surface Methodology: Process and Product Optimization Using Designed Experiments**, Wiley Series in Probability and Statistics, EUA, 2004.

GOIS, J. P.; **Reconstrução de Superfícies a Partir de Nuvens de Pontos**. 2004. 131 f. Dissertação de Mestrado em Ciência da Computação e Matemática Computacional– Instituto de Ciências Matemáticas e de Computação – ICMC-USP, São Carlos.

HE, Y.; QIN, H.; Surface Reconstruction with Triangular B-splines. In: **GEOMETRIC MODELING AND PROCESSING**, 2004, Beijing. p. 279-287.

HOPPE, H.; **Surface reconstruction from unorganized points**. 1994. 116 f. Tese de Doutorado - Department of Computer Science and Engineering, University of Washington, Washington.

HSIAO, S. W.; CHEN, R. Q.; A study of surface reconstruction for 3D mannequins based on feature curves. **Computer-Aided Design**. v. 45, n. 11, p. 1426-1441, nov. 2013.

HUANG, Y.; QIAN, X.; Dynamic B-spline surface reconstruction: Closing the sensing-and-modeling loop in 3D digitization. **Computer-Aided Design**. v. 39, n. 11, p. 987-1002, nov. 2007.

HUANG, M. C.; TAI, C. C.; The Pre-Processing of Data Points for Curve Fitting in Reverse Engineering. **The International Journal of Advanced Manufacturing Technology**, v. 16, n. 9, p. 635-642, jul. 2000.

ISAWA, K.; NAKAYAMA, S.; IKEDA, M.; TAKAGI, S.; TOSAKA, S.; KASAI, N.; Robotic 3D SQUID Imaging System for Practical Nondestructive Evaluation Applications. **Physica C: Superconductivity**. Japan, v. 432, n. 3-4, p. 182-192, nov. 2005.

KINERI, Y.; WANG, M.; LIN, H.; MAEKAWA, T.; B-spline surface fitting by iterative geometric interpolation/approximation algorithms. **Computer-Aided Design**. v. 44, n. 7, p. 697-708, jul. 2012.

LARSSON, S.; KJELLANDER, J. A. P.; A Laser Profile Scanner and a Robot as Platform for Unattended Acquisition of Unknown 3D Objects. In: PROCEEDINGS OF SIMsafe, 2006, Karlskoga. p. 1-13.

LIANG, K. M.; KHOO, B. E., RAJESWARI, M.; A Novel And Fast Shape Description Method Using Non-Uniform Rational B-Spline (NURBS). In: NATIONAL CONFERENCE ON COMPUTER GRAPHICS AND MULTIMEDIA (COGRAMM), 2002a, Melaka, Malásia.

LIANG, K. M.; RAJESWARI, M.; KHOO, B. E.; Similarity Measure Determination From NURBS-Warping Method. In: 7th INTERNATIONAL CONFERENCE ON CONTROL, AUTOMATION, ROBOTICS AND VISION (ICARCV), 7, 2002b, Singapura. p.1222 – 1227.

LYCHE, T.; MORKEN, K.; Spline Methods Draft. Department of Informatics Centre of Mathematics for Applications, Apostila sobre Spline, University of Oslo, 235 p, 2008.

MA, W.; KRUTH, J. P.; Parameterization of Randomly Measured Points for Least Squares Fitting of B-spline Curves and Surfaces. **Computer-Aided Design**. v. 27, n. 9, p. 663-675, sep. 1995.

MALISKA, C. R.; **Transferência de Calor e Mecânica de Fluidos Computacional**. 2. ed. LTC – Livros Técnicos e Científicos, 1995. 460p.

MARI, J. F.; **Reconstrução de Superfície 3D a Partir de Nuvens de Pontos Usando Rede Neurais Auto-Organizáveis**. 2007. 126 f. Dissertação de Mestrado em Ciência da Computação – Universidade Federal de São Carlos, São Carlos.

MOCZOZET, L.; DELLAS, F.; MAGNENAT-THALMANN, N.; BIASOTTI, S.; MORTARA, M.; FALCIDIENO, B.; MIN, P.; VELTKAMP, R.; Animatable Human Body Model Reconstruction from 3D Scan Data using Templates. In: PROC. CAP TECH 2004, 2004, Zermatt, Switzerland. **Capture Techniques for Virtual Environments**, p. 73-79.

MOTTA, R. C.; Curvas e Superfícies. Apostila sobre Curvas, Universidade de São Paulo, 2010.

MOUSTAKIDES, G.; BRIASSOULIS, D.; PSARAKIS, E.; DIMAS, E.; 3D Image Acquisition and NURBS Based Geometry Modelling of Natural Objects. **Advances in Engineering Software**. v. 31, n. 12, p. 955-969, dec. 2000.

MULLER, H.; Surface Reconstruction – An Introduction. In: SCIENTIFIC VISUALIZATION CONFERENCE, 1997, Dagstuhl, Germany. p. 239.

MÜLLER, H. A.; JAHNKE, J. H.; SMITH, D. B.; STOREY, M.; TILLEY, S. R.; WONG, K.; Reverse Engineering: A Roadmap. In: CONFERENCE ON THE FUTURE OF SOFTWARE ENGINEERING, 2000, Limerick, Ireland, p. 47-60.

NWODOH, T.; NNAJI, B.; POPPLESTONE, R.; LACH, E.; Three-dimensional Model Acquisition for Medical Robotics-assisted Burn Debridement System. **Robotics and Computer-integrated Manufacture**. Massachusetts, USA, v.13 n. 4, p. 309-318, dec. 1997.

PAVLIDIS, T.; **Algorithms for Graphics and Image Processing**. Springer, 1982. 466p.

PARK, H.; LEE, J. H.; B-spline Curve Fitting Based on Adaptive Curve Refinement Using Dominant Points. **Computer-Aided Design**. v. 39, n. 6, p. 439-451, jun. 2007.

PIEGL, L.; TILLER, W.; Curve and surface Constructions Using Rational B-splines. **Computer-Aided Design**. v. 19, n. 9, p. 485-498, nov. 1987.

PIEGL, L.; TILLER, W.; **The NURBS Book**. 2. ed. N.Y: Springer, 1997. 641 p.

REMONDINO, F. From Point Cloud to Surface: The Modeling and Visualization Problem. In: WORKSHOP ON VISUALIZATION AND ANIMATION OF REALITY-BASED 3D MODELS, 2003, Switzerland.

ROGERS, D. F.; **An Introduction to NURBS: With Historical Perspective**. 1. ed. San Diego: Morgan Kaufmann Publishers, 2001. 309 p.

SILVA, F. P.; DUARTE, L. C.; ROLDO, L.; KINDLEIN, W. JR.; Digitalização Tridimensional Móvel e sua aplicação Design do Produto. **Design e Tecnologia**, Pgdesign, v. 01, n. 01, p. 60-65, 2010.

SMITH Jr, R. E.; PRICE, J. M.; Howser, L. M.; A Smoothing Algorithm Using Cubic Spline Functions. National Aeronautics and Space Administration, Washington, 87 p, 1974.

SON S.; PARK H.; LEE K.H.; Automated Laser Scanning System for Reverse Engineering and Inspection. **International Journal Of Machine Tools And Manufacture**, Elsevier Science, v. 42, n. 8, p. 889-897, jun. 2002.

STODDART, A.; HILTON, A.; ILLINGWORTH, J.; Slime : A new deformable surface. In: 5th BRITISH MACHINE VISION CONFERENCE, 1994, York. p. 285 -294.

VÁRADY, T.; MARTIN, R. R.; COX, J.; Reverse Engineering of Geometric Models—an Introduction. **Computer-Aided Design**. Elsevier Science, v.29, n.44, p. 255-269, may. 1996.

VIEIRA, T. M. A.; Reconstrução de superfícies paramétricas triangulares usando uma técnica de SPR adaptada. II JORNADA NACIONAL DE INICIAÇÃO CIENTÍFICA, 2005. Universidade Federal de Alagoas.

WANGENHEIM, A. V.; WAGNER, H. M.; *Curvas Paramétricas*. Apostila sobre Curvas Paramétricas, Universidade Federal de Santa Catarina, 2013.

YAMAGUCHI, F.; **Curves and Surfaces in Computer Aided Geometric Design**, Springer-Verlag, 1988. 378p.

ZHENG, W.; BO, P.; LIU, Y.; WANG, W.; Fast B-spline curve fitting by L-BFGS. **Computer Aided Geometric Design**. v.29, n.7, p. 448-462, out. 2012.

APÊNDICE

Apêndice I – Função que Encontra o Vão em que o Nó se Encontra

```

%Encontra vão do nó

%% Entradas:
% grau=grau da função
% u=nó
% U= vetor nó
% no=número de nós

function mid=funencontravao2(no,u,U,grau)

if u>=U(no)
    u=no;
    mid = u-grau-1;

else
    low = grau-1;
    high = no-2;
    mid = fix((low+high)/2);

    while ((u<U(mid))||(u>=U(mid+1)))
        if(u<U(mid))
            high = mid;

        else
            low = mid;

        end
        mid = fix((low+high)/2);
    end
end
mid;

```

Apêndice II – Programa que Calcula as Funções de Base não Nulas

```

% Calculo das funções de base não nulas
%%% Entradas:
% grau=grau da função
% u=nó
% U= vetor nó
% vao=vão do nó
% no = número de nós

function N=funfuncbase2(no,grau,u,U,vao)

for n=1:1:no
if n==vao
N(n,1)=1;      % para N(i,0)
else
N(n,1)=0;
end
end

for p=2:1:grau+1
for i=1:1:no-p

x=((u-U(i))/(U(i+p-1)-U(i)));
if (u-U(i))==0 && (U(i+p-1)-U(i))==0
x=0;
end
if (U(i+p-1)-U(i))==0
x=0;
else
x=((u-U(i))/(U(i+p-1)-U(i)));
end

y=((U(i+p-1+1)-u)/(U(i+p-1+1)-U(i+1)));
if (U(i+p-1+1)-u)==0 && (U(i+p-1+1)-U(i+1))==0
y=0;
end
if (U(i+p-1+1)-U(i+1))==0
y=0;
else
y=((U(i+p-1+1)-u)/(U(i+p-1+1)-U(i+1)));
end

N(i,p)=x*N(i,p-1)+y*N(i+1,p-1);

end
end

```

% as funções de base não nulas são $N(i-p,p) \dots N(p,p)$

% neste programa os valores estão na matriz, deslocados de 1, ou seja, para $N(0,0) = N(1,1)$, e assim por diante.

Apêndice III – Função que multiplica as funções base não nulas pelos seus respectivos pontos de controle e plota a curva

```

%% Programa que plota a curva B-spline
%% Entradas:
% p=grau da função
% U= vetor nó
% grau=grau da curva;

no=length(U);
u=linspace(0,1,100);
Cx=0;
Cy=0;
for j = 1:100
for i=1:1:grau+1
    vao = funencontravao(no,grau,u(j),U);
    N = funfuncbase(no,grau,u(j),U,vao);

    Cx=Cx+N(vao-grau+i-1,grau+1)*x(vao-grau+i-1);
    Cy=Cy+N(vao-grau+i-1,grau+1)*y(vao-grau+i-1);

end

    a(j) = Cx;
    b(j) = Cy;
    Cx = 0;
    Cy = 0;
end

figure
plot(a,b,'k',x,y,'k-o')

```

Apêndice IV – Programa que calcula uma curva fechada

```

% para acrescentar p+2 nós iguais aos primeiros.
% vetor nó não uniforme

% Entradas:
% x=coordenadas em x dos pontos de controle
% y=coordenadas em y dos pontos de controle
% grau= grau da curva

n=length(x);
m=n+grau;
U1=zeros(1,m);

for i=1
    U(i)=0;
end
for i=m
    U(i)=1;
end

for i=2:m-1
    U(i)=(i-1)/m;
end
U;

no=length(U);

u=linspace(U(grau+1),U(no-grau),100);
Cx=0;
Cy=0;
for j = 1:100
for i=1:1:grau+1
    vao = funencontravao(no,u(j),U);
    N = funfuncbase(no,grau,u(j),U,vao);
    Cx=Cx+N(vao-grau+i-1,grau+1)*x(vao-grau+i-1);
    Cy=Cy+N(vao-grau+i-1,grau+1)*y(vao-grau+i-1);

end

    a(j) = Cx;
    b(j) = Cy;
    Cx = 0;
    Cy = 0;
end

plot(a,b,'r-',x,y,'k-o')

```

Apêndice V – Desenvolvimento da Eq. (3.47)

$$e = \sum_{k=0}^j \{w_k |Q_k - C(\bar{u}_k)|^2\} \quad (3.47)$$

De acordo com a Eq. (3.38) tem-se que $C(\bar{u}_k) = \sum_{i=1}^n N_{i,p}(\bar{u}_k) P_i$. Assim:

$$\begin{aligned} e &= \sum_{k=0}^j \{w_k |Q_k - C(\bar{u}_k)|^2\} = \sum_{k=0}^j \left\{ w_k \left| Q_k - \sum_{i=0}^n N_{i,p}(\bar{u}_k) P_i \right|^2 \right\} \\ &= \sum_{k=0}^j \left\{ w_k \left| Q_k Q_k - 2Q_k \sum_{i=0}^n N_{i,p}(\bar{u}_k) P_i + \left(\sum_{i=0}^n N_{i,p}(\bar{u}_k) P_i \right) \left(\sum_{i=0}^n N_{i,p}(\bar{u}_k) P_i \right) \right|^2 \right\} \\ &= \sum_{k=0}^j \left\{ w_k Q_k Q_k - w_k 2Q_k \sum_{i=0}^n N_{i,p}(\bar{u}_k) P_i + w_k \left(\sum_{i=0}^n N_{i,p}(\bar{u}_k) P_i \right) \left(\sum_{i=0}^n N_{i,p}(\bar{u}_k) P_i \right) \right\} \end{aligned}$$

Denominando $w_k Q_k = S_k$, tem-se:

$$= \sum_{k=0}^j \left\{ Q_k S_k - 2S_k \sum_{i=0}^n N_{i,p}(\bar{u}_k) P_i + w_k \left(\sum_{i=0}^n N_{i,p}(\bar{u}_k) P_i \right) \left(\sum_{i=0}^n N_{i,p}(\bar{u}_k) P_i \right) \right\}$$

Derivando a equação anterior em relação a P_i :

$$\sum_{k=0}^j \left\{ -2S_k N_{i,p}(\bar{u}_k) + N_{i,p}(\bar{u}_k) w_k \sum_{i=0}^n N_{i,p}(\bar{u}_k) P_i + N_{i,p}(\bar{u}_k) w_k \sum_{i=0}^n N_{i,p}(\bar{u}_k) P_i \right\} = 0$$

Logo,

$$\sum_{k=0}^j \left\{ -2S_k N_{i,p}(\bar{u}_k) + 2N_{i,p}(\bar{u}_k) w_k \sum_{i=0}^n N_{i,p}(\bar{u}_k) P_i \right\} = 0$$

$$\sum_{k=0}^j \left\{ -S_k N_{i,p}(\bar{u}_k) + N_{i,p}(\bar{u}_k) w_k \sum_{i=0}^n N_{i,p}(\bar{u}_k) P_i \right\} = 0$$

$$-\sum_{k=0}^j S_k N_{i,p}(\bar{u}_k) + \sum_{k=0}^j \sum_{i=0}^n N_{i,p}(\bar{u}_k) w_k N_{i,p}(\bar{u}_k) P_i = 0$$

$$\sum_{k=0}^j \sum_{i=0}^n w_k N_{i,p}(\bar{u}_k) N_{i,p}(\bar{u}_k) P_i = \sum_{k=0}^j S_k N_{i,p}(\bar{u}_k)$$

Denominando:

$$S_k N_{i,p}(\bar{u}_k) = R$$

Tem-se então uma equação do tipo:

$$(N^T W N) P = R$$

Apêndice VI – Exemplos de aplicação do método desenvolvido

Exemplo 1:

Quantidade de pontos que definem a curva: 56

Coordenadas dos pontos Q_k :

$Q = \{(-4.5878, -1.1534) (-4.7241, -0.6256) (-4.7464, -0.0982) (-4.6317, 0.4256) (-4.4518, 0.9038) (-4.1245, 1.4726) (-3.6762, 2.0329) (-3.1373, 2.5674) (-2.5953, 2.6823) (-2.0285, 2.8856) (-1.6040, 3.1142) (-1.0684, 3.5769) (-0.4702, 3.9767) (-0.0267, 4.1298) (0.4925, 4.1838) (0.9954, 4.1788) (1.5177, 4.2082) (2.0224, 4.1145) (2.5333, 3.9973) (3.0554, 3.9283) (3.6262, 3.7447) (4.1059, 3.6318) (4.5830, 3.4458) (5.0228, 3.0333) (5.1456, 2.5304) (5.2274, 2.0109) (5.2158, 1.4838) (5.1322, 0.9461) (4.9774, 0.4431) (4.6984, -0.1103) (4.5136, -0.6197) (4.4251, -1.1348) (4.4074, -1.6484) (4.1786, -2.2145) (3.9642, -2.6925) (3.6274, -3.2482) (3.1883, -3.8202) (2.9882, -4.2788) (2.5387, -4.7937) (1.9720, -5.2097) (1.5328, -5.4132) (0.9488, -5.7106) (0.4823, -5.8636) (-0.0345, -5.9847) (-0.5386, -5.9762) (-1.0505, -5.9560) (-1.5621, -5.8405) (-2.0205, -5.6842) (-2.6143, -5.4396) (-3.226, -4.9193) (-3.5522, -4.2661) (-3.7432, -3.7849) (-3.9454, -3.1971) (-4.0241, -2.7015) (-4.2058, -2.1997) (-4.4549, -1.5922)\}$

Os pesos relacionados a cada ponto, e definidos de acordo com o item 5.3.2, são:

$w_k = \{9.6287 \quad 3.7087 \quad 4.5749 \quad 2.6078 \quad 2.6325 \quad 2.1035 \quad 1.4698 \quad 8.1356 \quad 2.2168$
 $2.6123 \quad 3.4692 \quad 1.6361 \quad 4.0787 \quad 4.3604 \quad 2.1021 \quad 1.2260 \quad 4.3763 \quad 9.6287 \quad 1.6950$
 $3.0216 \quad 1.3874 \quad 2.6456 \quad 6.4451 \quad 9.6287 \quad 1.5140 \quad 3.1997 \quad 2.3375 \quad 2.5514 \quad 2.7802$
 $1.9413 \quad 3.1558 \quad 2.4802 \quad 5.8585 \quad 9.6287 \quad 1.9868 \quad 1.5170 \quad 3.7650 \quad 4.8823 \quad 2.9979$
 $3.1760 \quad 9.6287 \quad 2.5441 \quad 1.6081 \quad 4.5129 \quad 9.6287 \quad 3.3351 \quad 1.9982 \quad 1.0478 \quad 4.0938$
 $4.9481 \quad 1.2920 \quad 9.6287 \quad 2.9310 \quad 3.4740 \quad 9.6287 \quad 1.6134\}$

Quantidade de pontos de controle obtida: 28.

Como para uma curva fechada é necessário a repetição dos $p+1$ primeiros pontos, a quantidade final é de 32, sendo eles:

$P = \{(-4.0762, -2.3429) (-4.4573, -1.5909) (-4.5878, -1.1534) (-4.6239, -0.9751) (-4.7532, -0.3370) (-4.6159, 0.6405) (-3.9642, 1.8545) (-2.9982, 2.7933) (-1.5042, 2.8874) (-0.7359, 4.0892) (0.5688, 4.1924) (1.6172, 4.2321) (2.7968, 3.9488) (3.8109, 3.7519) (5.1101, 3.3191) (5.2211, 2.0733) (5.2783, 0.8991) (4.5469, -0.1462) (4.4765, -1.3113) (4.1516, -2.5374) (3.3060, -3.5241) (2.7830, -4.8037) (1.5748, -5.3844) (0.5208, -5.9575) (-0.6747, -6.0219) (-1.8085, -5.8138) (-3.1274, -5.2983) (-3.8981, -3.6236) (-4.0762, -2.3429) (-4.4573, -1.5909) (-4.5878, -1.1534) (-4.6239, -0.9751)\}$

Exemplo 2:

Quantidade de pontos que definem a curva: 43

Coordenadas dos pontos Q_k :

$Q = \{(-2.3467, -2.1920) (-2.2541, -2.6321) (-2.0072, -3.1119) (-1.4858, -3.5969) (-0.9823, -3.8659) (-0.5717, -3.9626) (-0.1180, -3.9223) (0.3377, -3.7983) (0.7986, -3.5407) (1.1699, -3.3704) (1.6514, -3.0656) (2.1016, -2.6367) (2.5045, -2.2052) (2.8214, -1.7428) (2.8757, -1.3381) (2.8962, -0.9226) (3.0192, -0.5221) (3.2322, -0.0759) (3.3485, 0.3334) (3.4494, 0.7822) (3.4541, 1.1771) (3.4053, 1.5815) (3.2808, 2.0112) (2.9345, 2.4027) (2.4394, 2.7332) (2.0669, 2.8800) (1.6032, 2.9595) (1.1645, 2.9520) (0.7473, 2.8876) (0.3206, 2.7804) (-0.1747, 2.5976) (-0.5593, 2.4511) (-1.0361, 2.1174) (-1.3980, 1.9485) (-1.8687, 1.6601) (-2.3524, 1.2091) (-2.6547, 0.7587) (-2.8128, 0.3406) (-2.8131, -0.0894) (-2.6672, -0.4904) (-2.4325, -0.9528) (-2.3963, -1.3388) (-2.3820, -1.7435)\}$

Os pesos relacionados a cada ponto são:

$w_k = \{3.1361 \quad 5.8924 \quad 6.0112 \quad 4.3881 \quad 5.6947 \quad 7.9304 \quad 4.1632 \quad 5.3122 \quad 1.8545$
 $2.9958 \quad 3.6047 \quad 1.0502 \quad 2.8525 \quad 10.4364 \quad 2.2252 \quad 6.4858 \quad 3.5266 \quad 4.0012 \quad 1.3741$
 $5.3351 \quad 3.5927 \quad 4.1338 \quad 9.8746 \quad 5.0233 \quad 4.6759 \quad 5.1387 \quad 4.4859 \quad 3.4533 \quad 2.3490$
 $2.4264 \quad 10.4364 \quad 5.4064 \quad 3.8663 \quad 2.5941 \quad 3.6048 \quad 4.1531 \quad 5.0561 \quad 8.9403 \quad 8.8698$
 $2.7852 \quad 9.0062 \quad 1.6055 \quad 1.0985\}$

Quantidade de pontos de controle obtida: 21. Os pontos de controle encontrados, já com a repetição são:

$P = \{(-2.1571, -1.0493) (-2.3966, -1.7420) (-2.3467, -2.1920) (-2.2947, -2.3857) (-2.1226, -3.0264) (-1.3751, -3.8196) (-0.1986, -4.0697) (0.9054, -3.5503) (1.7894, -2.9468) (2.8835, -2.1133) (2.8146, -0.9652) (3.2641, -0.0399) (3.5029, 0.8860) (3.4358, 2.0123) (2.5969, 2.7532) (1.5473, 3.0366) (0.5236, 2.8612) (-0.4523, 2.5257) (-1.4613, 1.8933) (-2.4248, 1.3131) (-3.1776, -0.1963) (-2.1571, -1.0493) (-2.3966, -1.7420) (-2.3467, -2.1920) (-2.2947, -2.3857)\}$

Exemplo 3:

Quantidade de pontos que definem a curva: 108

Coordenadas dos pontos Q_k :

$Q = \{(-15.8253, -10.0889) (-16.4657, -8.9452) (-16.9096, -8.1234) (-17.5552, -6.9365) (-17.9746, -6.0560) (-18.3852, -4.8955) (-18.7115, -3.8851) (-18.8750, -2.8616) (-18.9707, -1.8436) (-19.0492, -0.8284) (-19.1181, 0.1870) (-19.0994, 1.2018) (-19.0323, 2.2193) (-18.9260, 3.2388) (-18.7573, 4.2551) (-18.5261, 5.2672) (-18.2664, 6.2618) (-17.9155, 7.3044) (-17.4926, 8.3217) (-16.9355, 9.4965) (-16.4740, 10.3591) (-15.7621, 11.4023) (-14.8625, 12.5790) (-14.3534, 13.2047) (-13.4257, 14.3191) (-12.4218, 15.2581) (-11.3363, 16.1003) (-10.6251, 16.6254) (-9.5229, 17.3386) (-8.7026, 17.8010) (-7.4790, 18.3853) (-6.4674, 18.7983) (-5.4400, 19.1128) (-4.4202, 19.2916) (-3.4219, 19.5739) (-2.3967, 19.6504) (-1.3927, 19.8927) (-0.3738, 19.9279) (0.6432, 19.9677) (1.6592, 19.8638) (2.6677, 19.6688) (3.6842, 19.4881) (4.6965, 19.2574) (5.8284, 18.8111) (6.7224, 18.5418) (7.7407, 18.2389) (8.8102, 17.7822) (9.7666, 17.2859) (10.7972, 16.6536) (12.0636, 15.7293) (12.6585, 15.2106) (13.7297, 14.3135) (14.6742, 13.2658) (15.6004, 12.1855) (16.1060, 11.5187) (16.7982, 10.3598) (17.4683, 9.0748) (17.7792, 8.3333) (18.1408, 7.3114) (18.4986, 6.1327) (18.7330, 5.2731) (18.9832, 4.2574) (19.1871, 3.2420) (19.2964,$

2.2191) (19.3537, 1.2008) (19.3230, 0.1850) (19.1942, -0.8278) (19.0382,-1.8461) (18.8928, -2.8677) (18.6916, -3.8845) (18.3248, -5.2026) (18.0821, -5.9138) (17.6809, -6.9259) (17.2013, -8.0911) (16.7006, -8.9438) (16.0832, -9.9964) (15.1440, -11.2641) (14.6448, -11.8855) (13.6938, -12.9054) (12.6892, -13.8339) (11.5382, -14.7504) (10.8539, -15.2519) (9.7717, -15.9604) (8.4496, -16.6647) (7.7366, -16.9233) (6.7369, -17.3634) (5.6909, -17.7609) (4.6987, -18.0295) (3.6956, -18.2681) (2.6649, -18.3231) (1.6577, -18.5578) (0.6378, -18.5690) (-0.3756, -18.4666) (-1.3893, -18.4962) (-2.4046, -18.4230) (-3.4223, -18.2282) (-4.4301, -17.9470) (-5.4391, -17.7253) (-6.5014, -17.4467) (-7.4755, -17.0896) (-8.7465, -16.4385) (-9.5133, -15.9799) (-10.5360, -15.2337) (-11.3767, -14.7418) (-12.4444, -13.9050) (-13.5058, -12.9300) (-14.4833, -11.9739) (-15.3493, -10.8486)}

Os pesos relacionados a cada ponto são:

$w_k = \{10.6033 \quad 3.1910 \quad 50.3131 \quad 10.9663 \quad 22.4758 \quad 5.7447 \quad 34.8274 \quad 14.9135 \quad 3.8814$
 $2.1794 \quad 20.1259 \quad 11.0983 \quad 8.8420 \quad 13.9997 \quad 13.7984 \quad 7.0921 \quad 15.4580 \quad 14.9545 \quad 9.6663$
 $10.0857 \quad 22.8062 \quad 9.3343 \quad 6.3001 \quad 2.3389 \quad 20.9552 \quad 15.9129 \quad 5.0211 \quad 13.3527$
 $12.8553 \quad 14.0176 \quad 11.2212 \quad 19.8654 \quad 27.7716 \quad 23.3465 \quad 46.1808 \quad 37.4077 \quad 46.7778$
 $1.0787 \quad 32.8197 \quad 20.6887 \quad 3.4946 \quad 11.0310 \quad 31.9146 \quad 18.3498 \quad 50.3131 \quad 24.4387$
 $15.9141 \quad 14.8703 \quad 13.7290 \quad 17.4427 \quad 4.3307 \quad 23.6709 \quad 4.1761 \quad 12.6143 \quad 23.9638 \quad 9.7933$
 $17.6206 \quad 14.3452 \quad 9.3187 \quad 6.3810 \quad 6.0262 \quad 9.9103 \quad 21.1150 \quad 11.6408 \quad 20.1715 \quad 22.4350$
 $5.9308 \quad 2.4808 \quad 12.4275 \quad 15.0040 \quad 12.8893 \quad 12.5450 \quad 2.6246 \quad 29.6809 \quad 50.3131$
 $18.1900 \quad 7.8498 \quad 15.9410 \quad 12.7864 \quad 12.3205 \quad 8.1780 \quad 11.7123 \quad 15.3450 \quad 29.7967$
 $17.1711 \quad 11.0888 \quad 21.8605 \quad 7.0880 \quad 41.4662 \quad 40.3293 \quad 50.3131 \quad 26.0100 \quad 30.3368$
 $23.6656 \quad 27.0465 \quad 18.9594 \quad 12.7744 \quad 8.9632 \quad 21.1151 \quad 23.5003 \quad 13.4220 \quad 20.0967$
 $21.4167 \quad 27.5782 \quad 13.2919 \quad 5.2903 \quad 23.9492 \quad 19.7277\}$

Quantidade de pontos de controle obtida: 54. Os pontos de controle encontrados, já com a repetição são:

$P = \{(-14.2689, -12.4319) (-15.3501, -10.8492) (-15.8253, -10.0889) (-16.0185, -9.6840) (-$
 $16.5641, -8.5408) (-17.6356, -6.8462) (-18.5811, -4.7376) (-18.9487, -2.5725) (-19.1191, -$
 $0.4416) (-19.1256, 1.6807) (-18.8656, 3.7951) (-18.4201, 5.9372) (-17.6392, 8.0584) (-$

16.6092, 10.3177) (-15.0768, 12.2983) (-13.4625, 14.4250) (-11.5225, 16.0241) (-9.4484, 17.4347) (-7.4012, 18.4924) (-5.2449, 19.2190) (-3.1077, 19.5658) (-1.0144, 19.9923) (1.1153, 19.9733) (3.2969, 19.6782) (5.3515, 19.0046) (7.4872, 18.3150) (9.7396, 17.4485) (11.7652, 15.9750) (13.8426, 14.2996) (15.5279, 12.3925) (16.9183, 10.3018) (17.8967, 8.1783) (18.5359, 6.0352) (19.0978, 3.9293) (19.4016, 1.8366) (19.3143,-0.2928) (18.9848, -2.4796) (18.5437, -4.5988) (17.8350, -6.7480) (16.8090, -8.9069) (15.4291,-10.9783) (13.6814, -13.0468) (11.7332, -14.6036) (9.7062, -16.1398) (7.5974, -16.9799) (5.5361, -17.9371) (3.3749, -18.2609) (1.2790, -18.6544) (-0.8444, -18.4408) (-2.9848, -18.4732) (-5.0517, -17.7172) (-7.3008, -17.3443) (-9.3018, -16.0909) (-12.2084, -14.2253) (-14.2689, -12.4319) (-15.3501, -10.8492) (-15.8253, -10.0889) (-16.0185, -9.6840)}

Exemplo 4:

Quantidade de pontos que definem a curva: 75

Coordenadas dos pontos Q_k :

$Q = \{(-28.9910, 46.6960) (-25.3853, 43.3358) (-22.1966, 39.6740) (-19.9130, 36.4963) (-17.0013, 31.3987) (-15.0257, 26.2683) (-13.8350, 19.2677) (-13.9351, 15.1678) (-13.4325, 10.1126) (-10.2925, 4.5292) (-6.5377, 1.6300) (-3.3702, 0.4164) (6.7700,0) (13.5290,0) (17.3665, 0.4138) (20.2910, 1.3184) (23.6570, 3.1197) (27.1558, 6.2395) (30.2743, 11.3973) (33.1811, 18.7686) (34.8441,22.9856) (35.6550, 25.0402) (36.5004, 28.6061) (35.0578, 32.2505) (31.0256, 35.3968) (29.7183, 38.6835) (30.9586, 42.8365) (34.2654, 45.8557) (36.2768, 49.1706) (37.0023, 52.6205) (36.2979, 57.0341) (34.0, 60.0) (28.6379, 63.8125) (26.4286, 64.5547) (25.3726, 65.0639) (23.1949, 66.5049) (21.9249, 67.6632) (19.6912, 70.6941) (18.8963, 72.4161) (17.0314, 77.1611) (14.9579, 78.8879) (11.9272, 79.6384) (10.7424, 79.5297) (8.8656, 78.8728) (6.7334, 77.0479) (4.2879, 71.0139) (2.6070, 68.6048) (1.5301, 67.5190) (0.0743, 66.3898) (-1.2627, 65.6079) (-2.9853, 64.8843) (-5.3002, 64.3324) (-6.7390, 64.2072) (-10.2531, 64.5779) (-11.5200, 64.9592) (-14.1251, 66.0848) (-16.0110, 66.9316) (-19.1787, 68.3540) (-20.8307, 69.0958) (-23.0995, 70.1146) (-25.5948, 71.2352) (-28.1583, 72.3863) (-30.1530, 73.2820) (-32.6627, 74.0330) (-34.5482, 74.1554) (-36.7758, 73.8356) (-39.8262, 72.4607) (-42.0043, 70.4741) (-43.4351,68.1136) (-44.0070,$

66.3504) (-43.8645, 61.4109) (-42.4277, 58.4049) (-38.3776, 54.9057) (-36.9312, 53.3568) (-32.8126, 49.9018)}

Os pesos relacionados a cada ponto são:

$w_k = 10^4 \cdot \{0.4992 \quad 1.0100 \quad 1.0096 \quad 1.0098 \quad 1.2626 \quad 1.4971 \quad 1.6309 \quad 1.2758 \quad 3.3887$
 $3.3886 \quad 3.3886 \quad 2.2591 \quad 0.2303 \quad 0.9594 \quad 2.6362 \quad 2.6361 \quad 2.6361 \quad 2.6360 \quad 1.1427$
 $0.0001 \quad 0.0039 \quad 2.3102 \quad 7.5109 \quad 5.5119 \quad 5.7347 \quad 7.9087 \quad 5.7513 \quad 3.2309 \quad 4.3136$
 $4.3138 \quad 5.7203 \quad 2.6846 \quad 3.1121 \quad 3.3890 \quad 3.3887 \quad 3.3903 \quad 3.3884 \quad 3.3889 \quad 0.7866$
 $6.0263 \quad 7.3006 \quad 7.3009 \quad 7.2995 \quad 7.3010 \quad 4.7984 \quad 2.2446 \quad 3.6501 \quad 3.6505 \quad 3.6503$
 $3.6491 \quad 3.6512 \quad 3.6502 \quad 3.6498 \quad 3.6501 \quad 2.6322 \quad 0.2745 \quad 0.0002 \quad 7.9087 \quad 0.0003$
 $0.0005 \quad 0.0007 \quad 7.9087 \quad 2.5906 \quad 4.7441 \quad 4.7458 \quad 4.7453 \quad 4.7444 \quad 4.7454 \quad 4.7451$
 $4.7451 \quad 4.7450 \quad 4.4692 \quad 1.3587 \quad 1.5382 \quad 7.9087\}$

Quantidade de pontos de controle obtida: 37. Os pontos de controle encontrados, já com a repetição são:

$P = \{(-36.5385, 53.7598) (-32.8137, 49.9071) (-28.9909, 46.6960) (-27.9266, 45.6944) (-23.5375, 41.5643) (-18.8130, 34.5845) (-12.6900, 23.9319) (-15.3050, 11.9520) (-8.5109, -0.0300) (5.4872, -0.2511) (15.7753, -0.5096) (26.3078, 3.5544) (31.4198, 13.1548) (34.8149, 22.4228) (38.0728, 31.7447) (26.3756, 37.2261) (33.8763, 45.8482) (38.6162, 52.0876) (34.2995, 62.2191) (27.3118, 63.8544) (21.9413, 67.0875) (18.5813, 72.0322) (17.4099, 78.7586) (10.6293, 80.2567) (4.9816, 76.4488) (4.5608, 69.4345) (-0.4565, 65.9881) (-4.0583, 64.2524) (-8.9253, 64.0063) (-13.6901, 65.6753) (-18.3297, 68.0562) (-23.6171, 70.2185) (-28.1200, 72.3950) (-33.1080, 74.6389) (-38.8341, 73.5674) (-43.4989, 69.7345) (-45.8846, 58.1493) (-36.5385, 53.7598) (-32.8137, 49.9071) (-28.9909, 46.6960) (-27.9266, 45.6944)\}$

Apêndice VI – Programa Final

Programa principal que plota a curva B-spline fechada:

```

%% Entradas:
%x= coordenadas em x
%y=coordenadas em y

z=ones(1, length(x));
grau=3;

ponto=principal(x,y,z,grau);
for i=1:length(x)
    if i==ponto
        xnovo = [ x(i:end) x(1:(i-1)) ];
        ynovo = [ y(i:end) y(1:(i-1)) ];
        end
    end

x=xnovo;
y=ynovo;

[pontos2 np]=invertecurva(x,y,z); %%% função que calcula a quantidade de pontos de
controle

%condição dos pontos de controle
p50=fix(length(x)/2);
if np < p50
    np=p50;
end

%% método do comprimento de corda para definir Ubarra%%
tamanho=length(x);

norma=0;
for k=2:tamanho
    norma=norma+sqrt((x(k)-x(k-1))^2+(y(k)-y(k-1))^2);
end

for k=2:tamanho-1
    Ubarra(1)=0;
    Ubarra(tamanho)=1;
    norma2=sqrt((x(k)-x(k-1))^2+(y(k)-y(k-1))^2);
    Ubarra(k)=Ubarra(k-1)+norma2/norma;
end
Ubarra;

```

```

%%% definir o vetor nó U
npc=np;

d=(tamanho)/(npc-grau+1);

U(1:grau+1)=0;
for j=1:np-grau-1
i=fix(j*d);
alfa=j*d-i;
U(grau+1+j)=(1-alfa)*Ubarra(i)+alfa*Ubarra(i+1);
end

U(npc+grau-2:2*grau+npc-2)=1;
U;

no=length(U);
ubarra=length(Ubarra);

%%% MÉTODO LEAST SQUARES CURVE APPROXIMATION
B=funcbase(grau,Ubarra,U,ubarra,no,np); %%% função que calcula a matriz de funções de
base (N)

fim3=size(B);
fim2=fim3(2);
fim=fim3(1);

peso2=raiocurva2(x,y,z); %%% função que calcula o peso de cada ponto
w=peso2;

W=diag(w);

for i=1:length(w)
    Sx(i)=w(i)*x(i);
    Sy(i)=w(i)*y(i);
end

for i=1:fim2
    soma=0;
    for j=1:fim
        soma=soma+B(j,i)*Sx(j);
    end
    Rx(i,1)=soma;
end

for i=1:fim2
    soma=0;
    for j=1:fim
        soma=soma+B(j,i)*Sy(j);

```

```

    end
    Ry(i,1)=soma;
end

%% cálculo dos pontos de controle
Px=zeros(np,1);
Py=zeros(np,1);

BT=transpose(B);
A=(BT*W);
Px=(A*B)\Rx;
Py=(A*B)\Ry;

[Cix Ciy]=derivada(U,x,y,grau); %% função que calcula a derivada
dx=[x(1) Cix];
dy=[y(1) Ciy];

ca=(Ciy-y(1))/(Cix-x(1));
cb=y(1)-ca*x(1);
a=y(1)-Ciy;
b=Cix-x(1);
c=x(1)*Ciy-Cix*y(1);

ac=-1/ca;
bc2=Py(2)-ac*Px(2);
bc3=Py(3)-ac*Px(3);

Px(2)=(cb-bc2)/(ac-ca);
Px(3)=(cb-bc3)/(ac-ca);

Py(2)=ac*Px(2)+bc2;
Py(3)=ac*Px(3)+bc3;

d=(tamanho)/(np+1);

for j=1:np+grau-3
    i=fix(j*d);
    alfa=j*d-i;
    U3(j)=(1-alfa)*Ubarra(i)+alfa*Ubarra(i+1);
end

U2=zeros(1,np+grau-3+1);
U2(1)=0;

for i=2:np+grau-3+1
    U2(i)=U3(i-1);
end

```

```

for i=1:length(U2)-1
d(i)=U2(i+1)-U2(i);
end

for i=(grau+1):(length(U2)+grau)
    U(i)=U2(i-grau);
end

for i=1:grau
    U(grau-i+1)=U(grau-i+2)-d(length(U2)-i);
end

for i=(length(U2)+grau+1):(length(U2)+2*grau)
    U(i)=U(i-1)+d(i-length(U2)-grau);
end

px=zeros(1,length(Px)+grau+1);
py=zeros(1,length(Py)+grau+1);

for i=1:(grau-1)
    px(i)=Px(length(Px)-grau+i+1);
    py(i)=Py(length(Py)-grau+i+1);
end

for i=grau:(length(Px)+grau-1)
    px(i)=Px(i-2);
    py(i)=Py(i-2);
end

for i=(length(Px)+grau):(length(Px)+2*grau-2)
    px(i)=Px(i-length(Px)-grau+1);
    py(i)=Py(i-length(Px)-grau+1);
end

%% Plotar a curva
no=length(U);
u=linspace(U(grau+1),U(no-grau),1000);

Cx=0;
Cy=0;
for j = 1:1000
for i=1:1:grau+1
    vao = funencontravao2(no,u(j),U,grau);
    N = funfuncbase2(no,grau,u(j),U,vao);

    Cx=Cx+N(vao-grau+i-1,grau+1)*px(vao-grau+i-1);
    Cy=Cy+N(vao-grau+i-1,grau+1)*py(vao-grau+i-1);

```

```

end

a(j) = Cx;
b(j) = Cy;
Cx = 0;
Cy = 0;
end

[pos xp yp]=curvatura(x,y,z);

plot(a,b,'k-',px,py,'k-o')
hold on
plot(x,y,'k*')

axis equal

```

Função que define o ponto inicial:

```

function ponto=principal(x,y,z,grau)
x2=zeros(1,length(x));
y2=zeros(1,length(x));

va=angulos(x,y,grau);

for it=1:length(x)

[pontos2 np]=invertcurva(x,y,z);

p50=fix(length(x)/2);
if np < p50
    np=p50;
end

tamanho=length(x);

norma=0;
for k=2:tamanho
    norma=norma+sqrt((x(k)-x(k-1))^2+(y(k)-y(k-1))^2);
end

Ubarra = zeros(1,tamanho);

Ubarra(tamanho)=1;
for k=2:tamanho-1
    norma2=sqrt((x(k)-x(k-1))^2+(y(k)-y(k-1))^2);
    Ubarra(k)=Ubarra(k-1)+norma2/norma;

```

```

end

npc=np;

d=(tamanho+1)/(npc-grau+1);

U = zeros(1,2*grau+npc-2);
U(1:grau+1)=0;
for j=1:np-grau-1
    i=fix(j*d);
    alfa=j*d-i;
    U(grau+1+j)=(1-alfa)*Ubarra(i)+alfa*Ubarra(i+1);
end

U(npc+grau-2:2*grau+npc-2)=1;

no=length(U);
ubarra=length(Ubarra);

B=funcbase(grau,Ubarra,U,ubarra,no,np);

fim3=size(B);
fim2=fim3(2);
fim=fim3(1);

peso2=raiocurva2(x,y,z);
w=peso2;
W=diag(w);

Sx = zeros(1,length(w));
Sy = zeros(1,length(w));
for ii=1:length(w)
    Sx(ii)=w(ii)*x(ii);
    Sy(ii)=w(ii)*y(ii);
end

Rx = zeros(fim2,1);
for ii=1:fim2
    soma=0;
    for j=1:fim
        soma=soma+B(j,ii)*Sx(j);
    end
    Rx(ii,1)=soma;
end

Ry = zeros(fim2,1);

```

```

for ii=1:fim2
    soma=0;
    for j=1:fim
        soma=soma+B(j,ii)*Sy(j);
    end
    Ry(ii,1)=soma;
end

```

```

Px=zeros(np,1);
Py=zeros(np,1);
BT=transpose(B);
A=(BT*W);
Px=(A*B)\Rx;
Py=(A*B)\Ry;

```

%% encontrar a distância dos pontos de controle 2 e 3, assim encontrar o ponto de interseção dessa distancia com a reta da tangente.

```

[Cix Ciy]=derivada(U,x,y,grau);
dx=[x(1) Cix];
dy=[y(1) Ciy];

```

```

%%equação da reta
ca=(Ciy-y(1))/(Cix-x(1));    %coeficiente angular
cb=y(1)-ca*x(1);            %coeficiente linear

```

```

%%equação geral da reta
a=y(1)-Ciy;
b=Cix-x(1);
c=x(1)*Ciy-Cix*y(1);

```

```

d2=(abs(a*Px(2)+b*Py(2)+ c))/ (sqrt(a^2+b^2));    %%distância de ponto a reta
d3=(abs(a*Px(3)+b*Py(3)+ c))/ (sqrt(a^2+b^2));
dist(it)=d2+d3;

```

```

x2 = [x(2:end) x(1)];
y2 = [y(2:end) y(1)];

```

```

x=x2;
y=y2;

```

```

end

```

```

dist;
va=angulos(x,y,grau);
idx = va(va~=0);
dist2 = dist(idx);
[mindist, ponto] = min(dist2);
ponto = idx(ponto);

```

```

if va==0;
    ponto=1;
end;

```

Função que calcula os ângulos entre os pontos dados (Q) e verifica quais pontos estão mais alinhados:

```

function va=angulos(x,y,grau)
X=zeros(1,length(x)+2);
Y=zeros(1,length(y)+2);

for i=1:length(x)
    X(i)=x(i);
    Y(i)=y(i);
end

for i=1:2
    X(length(x)+i)=x(i);
    Y(length(y)+i)=y(i);
end

for i=1:length(x)
    a=sqrt((X(i+1)-X(i))^2+(Y(i+1)-Y(i))^2);
    c=sqrt((X(i+2)-X(i))^2+(Y(i+2)-Y(i))^2);
    b=sqrt((X(i+2)-X(i+1))^2+(Y(i+2)-Y(i+1))^2);
    lados(i,:)=[a c b];
    ang(i)=acos(((a^2+b^2)-c^2)/(2*a*b));
end

angulo=zeros(1,length(ang));
for i=2:length(ang)
    angulo(i)=ang(i-1);
end
angulo(1)=ang(end);
teta=angulo*(180/pi);

am=mean(teta); %% média dos ângulos
adp=std(teta); %% desvio padrão dos ângulos

va=zeros(1,length(teta));
for i=1:length(teta)
    if teta(i)>am+adp/2;
        va(i)=i;
    end
end

for i=1:length(va)
    if va(i)~=0

```

```

    x2(i)=x(i);
    y2(i)=y(i);
end
end

```

Função que calcula o vetor unitário e define o ponto que muda a curvatura:

```

function [pontos2 np]=invertecurva(x,y,z)

c=length(x);
Q=[x;y;z];
q=Q';

q2=zeros(c+2,3);
q2(1:c,1:3) = q(1:c,1:3);
q2(c+(1:2),1:3) = q(1:2,1:3);

V1 = q2(2:(end-1),:) - q2(1:(end-2),:);
V2 = q2(3:end,:) - q2(1:(end-2),:);
G = cross(V1,V2,2);

Gf=G(:,3);
Gff=Gf';

%% verifica no vetor G qual a posição em que os vetores mudam de sinal
prod=[0 Gff].*[Gff 0];
mud=prod<=0;

% pega os pontos nos quais o vetor muda de sinal
a=1;
for i=1:length(Gf)
    if mud(i)==1
        pontos(a,:)=q(i,:);
        a=a+1;
    end
end

for j=1:2
    for i=1:length(pontos)
        pontos2(i,j)=pontos(i,j);
    end
end

x2=pontos2(:,1);
y2=pontos2(:,2);

np=length(x2);

```

Função que calcula as funções de base para o vetor no \bar{U} :

```
function B=funcbase(grau,Ubarra,U,ubarra,no,np)

for j=1:ubarra
    u(j)=Ubarra(j);

vao=encontravao(no,u(j),U,graue);

for n=1:1:no
    if n==vao
        N(n,1)=1;
    else
        N(n,1)=0;
    end
end

for p=2:1:graue+1
    for i=1:1:no-p

x=((u(j)-U(i))/(U(i+p-1)-U(i)));
    if (u(j)-U(i))==0 && (U(i+p-1)-U(i))==0
        x=0;
    end
    if (U(i+p-1)-U(i))==0
        x=0;
    else
        x=((u(j)-U(i))/(U(i+p-1)-U(i)));
    end

y=((U(i+p-1+1)-u(j))/(U(i+p-1+1)-U(i+1)));
    if (U(i+p-1+1)-u(j))==0 && (U(i+p-1+1)-U(i+1))==0
        y=0;
    end
    if (U(i+p-1+1)-U(i+1))==0
        y=0;
    else
        y=((U(i+p-1+1)-u(j))/(U(i+p-1+1)-U(i+1)));
    end

N(i,p)=x*N(i,p-1)+y*N(i+1,p-1);

    end
end
    B(j,:)=N(1:np,p);
end
```

Função que encontra o vão no vetor no \bar{U} :

```

function mid=encontravao(no,u,U,grau)
if u>=U(no)
    u=no;
    mid = u-grau-1;
else
    low =grau-1;
    high = no-2;
    mid = fix((low+high)/2);
    while ((u<U(mid))||(u>=U(mid+1)))
        if(u<U(mid))
            high = mid;

        else
            low = mid;
        end
        mid = fix((low+high)/2);
    end
end
mid;

```

Função que calcula os pesos de cada ponto:

```

function peso2=raiocurva2(x,y,z)
X=zeros(1,length(x)+2);
Y=zeros(1,length(y)+2);
for i=1:length(x)
    X(i)=x(i);
    Y(i)=y(i);
end
for i=1:2
    X(length(x)+i)=x(i);
    Y(length(y)+i)=y(i);
end

for i=1:length(x)
a1=(Y(i+1)-Y(i))/(X(i+1)-X(i));
a2=(Y(i+2)-Y(i+1))/(X(i+2)-X(i+1));
a(i,:)=[a1 a2];
end
for i=1:length(x)
    b1=Y(i)-a(i,1)*X(i);
    b2=Y(i+1)-a(i,2)*X(i+1);
b(i,:)=[b1 b2];
end

```

```

%%%DEFINIR MEDIATRIZ%%%
for i=1:length(x)
    ym1=(Y(i)+Y(i+1))/2;
    xm1=(X(i)+X(i+1))/2;
    ym2=(Y(i+1)+Y(i+2))/2;
    xm2=(X(i+1)+X(i+2))/2;
    YM(i,:)=[ym1 ym2];
    XM(i,:)=[xm1 xm2];
end
for i=1:length(x)
    am1=-1/a(i,1);
    am2=-1/a(i,2);
    am(i,:)=[am1 am2];
end

%definindo coeficiente linear da mediatriz
for i=1:length(x)
    bm1=YM(i,1)-am(i,1)*XM(i,1);
    bm2=YM(i,2)-am(i,2)*XM(i,2);
    bm(i,:)=[bm1 bm2];
end

%%%Definição do centro da circunferência%%%
for i=1:length(x)
    if a(i,1)==0
        xc= XM(i,1);
    elseif a(i,2)==0
        xc= XM(i,2);
    elseif a(i,1) && a(i,2) == 0
        xc=inf;
    else
        xc=(bm(i,2)-bm(i,1))/(am(i,1)-am(i,2));

    end

    if a(i,1)==0
        yc=am(i,2)*XM(i,1)+bm(i,2);
    elseif a(i,2)==0
        yc=am(i,1)*XM(i,2)+bm(i,1);
    elseif a(i,1) && a(i,2) == 0
        yc=inf;
    else
        yc=am(i,1)*xc+bm(i,1);
    end

    if a(i,1)==inf
        yc=YM(i,1);
        xc=(YM(i,1)-bm(i,2))/am(i,2);
    elseif a(i,2)==inf
        yc=YM(i,2);

```

```

        xc=(YM(i,2)-bm(i,1))/am(i,1);
    end
    C(i,:)=[xc yc];
end

%cálculo do raio
for i=1:(length(X)-2)
    r=sqrt((C(i,1)-X(i+1))^2+(C(i,2)-Y(i+1))^2);
    raio(i)=[r];
end
r2=zeros(1,length(raio));
for i=2:length(raio)
    r2(i)=raio(i-1);
end
r2(1)=raio(end);
raiono=r2;
for i=1:length(x)
    if r2(i)==Inf
        r2(i)=1000;
    end
end
end

C2=zeros(length(raio),2);
for i=2:length(raio)
    C2(i,:)=C(i-1,:);
end
C2(1,1)=C(length(raio),1);
C2(1,2)=C(length(raio),2);
C3=C2;
raio2=r2;

for i=length(x):-1:1
    if raio2(i)==1000
        raio2(i)=[];
        C2(i,:)=[];
    end
end

s=sum(raio2); %cálculo da soma dos raios
rm=mean(raio2); %cálculo da media dos raios
rdp=std(raio2); % cálculo do desvio padrão
rmax=rm+rdp; % raio máximo= media + desvio padrão
rmin=rm-rdp; % raio mínimo= media - desvio padrão
rc=raio2;
for i=length(raio2):-1:1
    if rc(i)>=rmax
        rc(i)=[];
    end
end
end

```

```

%calculo do peso
for i=1:length(rc)
    p(i)=rmax/rc(i);
end
mp=max(p);

peso2=zeros(1,length(r2));
for i=1:length(r2)
if r2(i)>=rmax || r2(i)==1000
    peso2(i)=mp;
end
end

k=0;
j=1;
for i=1:length(peso2)
if peso2(i)==0
    k=p(j);
    j = j+1;
peso2(i)=k;
end
end
peso2;

```

Função que calcula a derivada do primeiro ponto:

```

function [Cfx Cfy]=derivada(U,x,y,grau)
Cix=(grau/U(grau+2))*(x(2)-x(1));
Ciy=(grau/U(grau+2))*(y(2)-y(1));
Cfx=(grau/(1-U(end-grau-1)))*(x(end)-x(end-1));
Cfy=(grau/(1-U(end-grau-1)))*(y(end)-y(end-1));
dx=[x(1) Cix];
dy=[y(1) Ciy];

dfx=[x(end-1) Cfx];
dfy=[y(end-1) Cfy];

```

Função que encontra o vão no vetor no U:

```

function mid=funencontravao2(no,u,U,grau)

if u>=U(no)
    u=no;
    mid = u-grau-1;

else
    low = grau-1;
    high = no-2;
    mid = fix((low+high)/2);

```

```

while ((u<U(mid))||(u>=U(mid+1)))
if(u<U(mid))
high = mid;
else
low = mid;
end
mid = fix((low+high)/2);
end
end
mid;

```

Função que calcula as funções de base para o vetor no U:

```

function N=funfuncbase2(no,grau,u,U,vao)

for n=1:1:no
if n==vao
N(n,1)=1;
else
N(n,1)=0;
end
end

for p=2:1:grau+1
for i=1:1:no-p

x=((u-U(i))/(U(i+p-1)-U(i)));
if (u-U(i))==0 && (U(i+p-1)-U(i))==0
x=0;
end
if (U(i+p-1)-U(i))==0
x=0;
else
x=((u-U(i))/(U(i+p-1)-U(i)));
end

y=((U(i+p+1-1)-u)/(U(i+p+1-1)-U(i+1)));
if (U(i+p-1+1)-u)==0 && (U(i+p-1+1)-U(i+1))==0
y=0;
end
if (U(i+p-1+1)-U(i+1))==0
y=0;
else
y=((U(i+p-1+1)-u)/(U(i+p-1+1)-U(i+1)));
end
N(i,p)=x*N(i,p-1)+y*N(i+1,p-1);
end
end end

```