



Universidade Federal da Paraíba
Centro de Ciências Exatas e da Natureza
Departamento de Matemática
Mestrado Profissional em Matemática
em Rede Nacional PROFMAT



Curvas de Bézier

por

EVERT ELVIS BATISTA DE ALMEIDA

sob orientação do

Prof. Dr. Lenimar Nunes de Andrade

Trabalho de conclusão de curso apresentado ao Corpo Docente do Mestrado Profissional em Matemática em Rede Nacional PROFMAT CCEN-UFPB, como requisito parcial para obtenção do título de Mestre em Matemática

Fevereiro/2015
João Pessoa - PB

A447c Almeida, Evert Elvis Batista de.
Curvas de Bézier / Evert Elvis Batista de Almeida.- João
Pessoa, 2015.
67f. : il.
Orientador: Lenimar Nunes de Andrade
Dissertação (Mestrado) - UFPB/CCEN
1. Matemática. 2. Curvas de Bézier. 3. Polinômio de
Bernstein. 4. Software Geogebra.

UFPB/BC

CDU: 51(043)

Curvas de Bézier

por

EVERT ELVIS BATISTA DE ALMEIDA

Dissertação apresentada ao Corpo Docente do Mestrado Profissional em Matemática em Rede Nacional PROFMAT CCEN-UFPB, como requisito parcial para obtenção do título de Mestre em Matemática.

Área de Concentração: Matemática.

Aprovada por:

Prof. Dr. LENIMAR NUNES DE ANDRADE -UFPB (Orientador)

Prof. Dr. FRANK SINATRA GOMES DA SILVA - UFRPE

Prof. Dr. EDSON DE FIGUEIRÊDO LIMA JUNIOR - UFPB

Fevereiro/2015

Agradecimentos

Quero agradecer aos meus colegas da turma principalmente ao galego (Erinaldo), aos meus professores que tiveram toda paciência para transmitir o conhecimento ao longo do curso na UFPB. Aos meus familiares que suportaram minha ausência durante os sábados e todas as horas de estudo. Em especial a minha adorável companheira Erika Maria Lins por todo apoio e compreensão durante esta jornada.

Dedicatória

*Ao meu avô Eufrásio, que
deus tenha em bom lugar.
In memorium*

Resumo

Neste trabalho fazemos uma introdução às Curvas de Bézier, importante item da aplicação matemática que originou-se na indústria automobilística francesa e que têm aplicações em várias áreas científicas. Diversos conceitos básicos são revisitados tais como curvas definidas parametricamente, polinômios de Bernstein e polinômios de interpolação. Ao longo do texto, é abordado o algoritmo de Casteljau para construção de curva e suas derivadas. São implementados exemplos de construção usando o GeoGebra e $\text{\LaTeX}$.

Palavras-chave: Curvas de Bézier, Polinômio de Bernstein, Software Geogebra.

Abstract

In this work we will make an introduction an important application mathematics called Bézier curves. The history of this curve originated in industry automobile French , and found many applications in various fields of science. Revisit some concepts such as parametric functions, polynomials Bernstein and interpolation for definition the curves Bézier. We will discuss the algorithm Casteljau which facilitates the construction of the curve and determine derivative. Throughout the text we will implement some examples with Geogebra software and $\text{\LaTeX}$ in addition to discuss relevant issues that arouse public interest.

Keywords: Bézier Curves, Bernstein Polynomials, Software Geogebra.

Sumário

Abstract	vii
1 Revisão matemática	2
1.1 Introdução	2
1.2 Curvas parametrizadas	3
1.3 Polinômios de Bernstein	5
1.3.1 Propriedades dos polinômios de Bernstein	9
1.4 Interpolação	11
1.4.1 Interpolador de Lagrange	11
1.4.2 Interpolador Linear	12
2 Curvas de Bézier	15
2.1 Uma breve história das curvas de Bézier e sua importância	15
2.2 Definição das Curvas de Bézier	18
2.2.1 Propriedades das Curvas de Bézier	19
2.2.2 Algoritmo de Casteljau	20
2.2.3 Cálculo da Derivada em relação a Curva de Bézier	22
2.2.4 Cálculo da integral da Curva de Bézier	23
3 Exemplos	28
3.1 Exemplos	28
3.1.1 Exemplo 1	28
3.1.2 Exemplo 2	28
3.1.3 Exemplo 3	30
3.1.4 Exemplo 4	31
3.1.5 Exemplo 5	32
3.1.6 Exemplo 6	32
3.1.7 Exemplo 7	32

4	Resultados e Considerações Finais	38
4.1	Resultados	38
4.2	Considerações finais	39
A	Construindo curvas de Bézier usando o Geogebra e com o $\text{\LaTeX}$	40
A.1	Introdução	40
A.2	Construção no Geogebra	40
A.3	Usando o PSTricks	44
A.4	Usando o ambiente <i>figure</i>	46
A.5	Usando o PGF	53
A.6	Usando o TEXdraw	54
A.7	Usando o pacote bez123	54
B	Teorema da Aproximação de Weierstrass	56
	Referências Bibliográficas	58

Lista de Figuras

1.1	Reta gerada da forma $P + tv$ construída no Geogebra.	4
1.2	circunferência gerada da forma $x^2 + y^2 = 4$ construída no Geogebra. . .	5
1.3	circunferência gerada da forma $x = 2\cos(2\pi t)$ e $y = 2\sin(2\pi t)$	6
1.4	Elipse gerada da forma $x = R\cos(2\pi t)$ e $y = r\sin(2\pi t)$	7
1.5	A trajetória elíptica ou circular descrita por um ponto fixo na barra dependendo da razão entre os pontos A e B.	8
1.6	Bernstein de grau 1.	8
1.7	Bernstein de grau 2.	9
1.8	Bernstein de grau 3.	9
1.9	Usando o Geogebra para verificar o polinômio interpolador.	13
1.10	Imagem de um pato usada na interpolação.	13
1.11	Pontos interpolados sobre a imagem do pato pelo polinômio de Lagrange.	14
1.12	Exemplo de interpolação linear usando o software wxMaxima.	14
2.1	Projeto de veículo usando curvas de Bézier.	15
2.2	Formas geradas por alguns software usando as curvas de Bézier. . . .	16
2.3	Usando curvas em cirurgias plásticas.	17
2.4	Invólucros convexo nas curvas Curvas de Bézier variando os pontos de controle.	20
2.5	Pontos da curva com três Pontos de controle gerada segundo o algoritmo de Casteljau e habilitando o lugar geométrico no Geogebra . . .	25
2.6	Curvas de Bezier com três usando o rastro no geogebra.	26
2.7	Curvas de Bézier com quatro pontos de controle usando o rastro. . . .	26
2.8	Pontos gerados pelo algoritmo De Casteljau com $t = 0.5$	27
2.9	Derivada da Curva gerada pelos pontos $(0, 0)$, $(0, 2)$, $(8, 2)$ e $(4, 0)$. . .	27
3.1	Polígono de controle formado por 17 pontos.	29
3.2	Polígono de controle formado por 17 pontos variando o t com incrementos de 0.2.	29
3.3	Polígono de controle formado por 17 pontos com $t = 0.5$	30
3.4	Curva de Bézier usando três coordenadas no espaço.	30

3.5	Gerando o Polígono de controle com os dois pentágonos.	31
3.6	Implementando o algoritmo de Casteljau e variando t	31
3.7	Usando o lugar geométrico para gerar a Curva de Bézier em Espiral. .	32
3.8	Espiral formada por alguns pontos de controle de uma sequência de triângulos.	33
3.9	Implementação do algoritmo de Castejau.	34
3.10	Curva gerada pelos pontos $(-1, 0)$, $(0, 1)$, $(1, 1)$, $(0, 2)$, $(0, 1)$, $(-1, 1)$, $(0, 2)$, $(0, 1)$ e $(1, 0)$	34
3.11	Escolha dos pontos sobre a imagem.	35
3.12	Variando o parâmetro t observamos os pontos intermediários gerados. .	35
3.13	Curva gerada usando o lugar geométrico.	36
3.14	Curva gerada para os pontos $(1, 0)$, $(-1, 1)$, $(1, 1)$ e $(-1, 0)$ em $t = 0.5$ a derivada não existe para estes valores.	36
3.15	Curva gerada para os pontos $(0, 1)$, $(0, 0)$, $(1, 1)$ e $(-1, 0)$ em $t = 0.5$ a derivada não existe para estes valores.	37
3.16	Curva gerada para os pontos $(0, -1)$, $(0, 2)$, $(1, 1)$ e $(-1, 0)$ em $t = 0.5$ a derivada não existe para estes valores.	37
A.1	Seleção da opção Ponto na barra de ferramenta.	41
A.2	Escolha dos Pontos.	41
A.3	Construção dos segmentos.	42
A.4	Segmento gerado pelos pontos A e B.	42
A.5	Construção dos segmentos dos pontos B, C e D.	43
A.6	Implementação do controle deslizante.	43
A.7	Definindo o interlalo do parâmetro t	44
A.8	Controle deslizante t	44
A.9	Construção do ponto sobre o segmento usando a caixa de entrada. . .	45
A.10	Ponto sobre o segmento $\overline{AB}$	45
A.11	Pontos F e G sobre os segmentos $\overline{BC}$ e $\overline{CD}$	46
A.12	Construção do segmento $\overline{EF}$	46
A.13	Segmentos $\overline{EF}$ e $\overline{FG}$	47
A.14	Construção dos pontos sobre os segmentos $\overline{EF}$ e $\overline{FG}$	47
A.15	Pontos H e G.	48
A.16	Gerando o ponto J usando a caixa de entrada.	48
A.17	Ponto J.	49
A.18	Seleção do lugar geométrico na barra de ferramenta.	49
A.19	Gerando a curva de Bézier em relação ao ponto J e o parâmetro t . . .	50
A.20	Variação do ponto J usando o controle deslizante.	51
A.21	Curva gerada com o comando psbezier	52
A.22	Curva gerada com o comando clvec do TEXdraw	54

Capítulo 1

Revisão matemática

1.1 Introdução

A educação brasileira tem passado nas últimas décadas por um período de grandes transformações, tanto no âmbito da condução das políticas educacionais quanto no processo ensino-aprendizagem, estabelecido a partir da evolução (às vezes involução) das teorias de ensino. Tal revolução vem produzindo uma nova cultura, colocando o conhecimento científico-tecnológico na condição de um efetivo e indispensável fator de produção no mundo capitalista.

O estabelecimento de eixos norteadores para o desenvolvimento de ações que promovam a efetividade e a eficácia do ensino-aprendizagem deve ser uma prática nas instituições de ensino, objetivando o atendimento de uma parcela cada vez maior da população, em busca do desenvolvimento local, regional e nacional.

O ensino de matemática vem passando por uma fase de revisão no que diz respeito ao conteúdo ministrado em sala de aula, revisar alguns temas, às vezes sofisticados mas de fácil entendimento, pode ser uma excelente estratégia para incentivar a curiosidade dos estudantes, e despertar o interesse pela matemática.

Neste trabalho de conclusão do curso de mestrado profissional fazemos uma breve descrição sobre a história da Curva de Bézier e suas aplicações em inúmeras áreas da ciência.

Uma revisão matemática com alguns tópicos que fundamentam a construção das Curvas de Bézier será realizada neste primeiro capítulo. Vamos revisar alguns conceitos para melhor entendimento da definição da Curva de Bézier. Começamos definindo curvas parametrizadas e polinômios de Bernstein. Por fim definimos interpolação, e introduzimos exemplos de interpolação pelo polinômio de Lagrange e interpolação linear.

No estudo da curva de Bézier algumas atividades foram implementadas ao longo da revisão matemática para ilustrar os conceitos abordados. No segundo capítulo será apresentada a curva de Bézier e sua fundamentação matemática, com alguns

exemplos implementados no Geogebra.

No terceiro capítulo abordamos alguns exemplos propostos e outros retirados do livro de referência Farin [2].

Os resultados obtidos foram discutidos nas conclusões e considerações finais, no quarto capítulo.

O presente trabalho foi realizado com apoio da CAPES, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior.

1.2 Curvas parametrizadas

Em aplicações computacionais existem diversas maneiras de representar um objeto gráfico, mas destacamos duas que são comumente usadas a descrição paramétrica ou implícita.

Na descrição paramétrica a curva é definida por uma função $f : I \subset \mathbb{R} \rightarrow \mathbb{R}^2$, $f(t) = (x(t), y(t))$, onde I é um intervalo na reta. Na física as equações paramétricas possuem uma interpretação interessante, considerando a variável $t \in I$ como sendo o tempo, a curva dada por $f(t)$ representa uma partícula no plano [10].

Uma reta, por exemplo, implicitamente representada pela equação $r : ax + by + c = 0$, esta reta r pode ser representada no plano por um ponto $P \in r$ [9], e o vetor $\mathbf{v}$ que determina a sua direção, figura 1.1 . A forma vetorial da equação paramétrica da reta é dada por

$$f(t) = P + t\mathbf{v}, t \in \mathbb{R}.$$

Se $P = (x_0, y_0)$, $\mathbf{v} = (v_1, v_2)$ e $f(t) = (x(t), y(t))$, então podemos escrever a equação paramétrica de r em coordenadas:

$$(x(t), y(t)) = (x_0, y_0) + t(v_1, v_2)$$

É fácil perceber que podemos representar uma circunferência usando duas maneiras distintas, por exemplo.

$$x^2 + y^2 = 4$$

ou

$$x = 2 \operatorname{sen}(2\pi t)$$

$$y = 2 \cos(2\pi t)$$

.

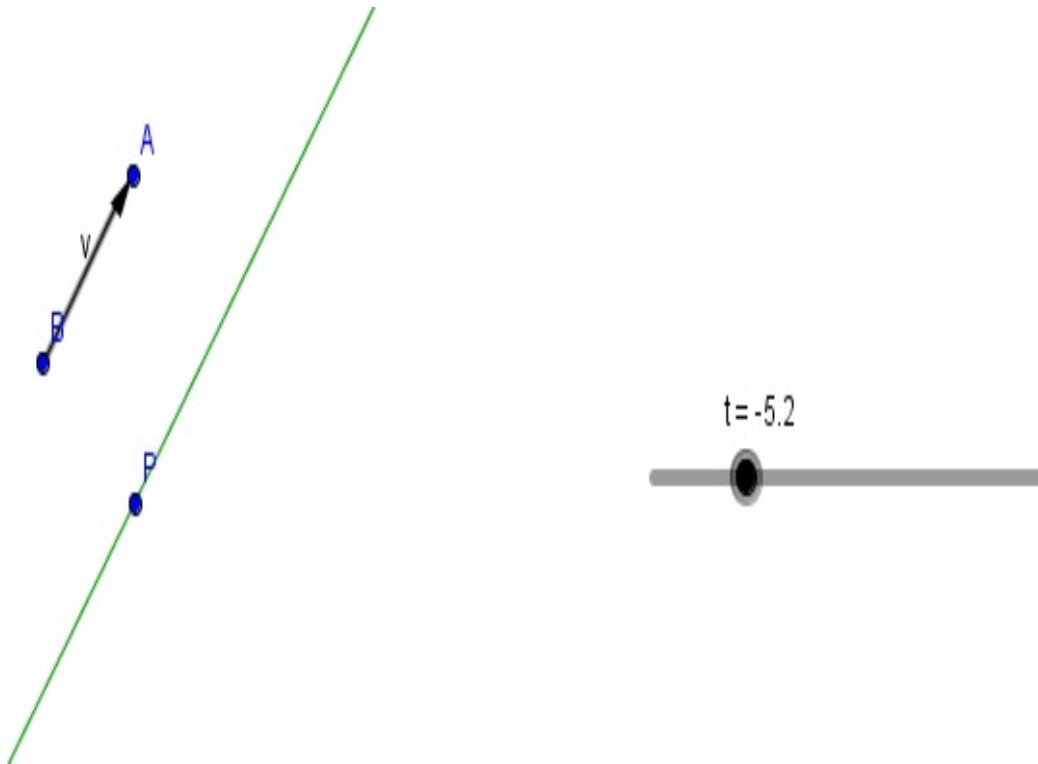


Figura 1.1: Reta gerada da forma $P + tv$ construída no Geogebra.

A primeira forma apresenta uma dificuldade, apenas os valores positivos de y são gerados, pois as coordenadas são dadas por $(x, \sqrt{4 - x^2})$, figura 1.2 . A forma paramétrica gera todos os pontos da circunferência e de maneira suave para a máquina, figura 1.3 .

Na equação da elipse podemos simplificar bastante a expressão:

$$\frac{x^2}{R^2} + \frac{y^2}{r^2} = 1$$

onde R é o raio maior e r é o raio menor, figura 1.4 . Na forma paramétrica temos:

$$x = R \cos(2\pi t)$$

$$y = r \sin(2\pi t)$$

Uma situação bastante interessante é apresentada na figura 1.5, onde uma barra tem suas extremidades nos eixos X e Y , podendo deslizar apenas ao longo dos eixos, um ponto fixo descreve uma trajetória que varia entre Elipses e uma circunferência variando a posição do ponto, figura 1.5.

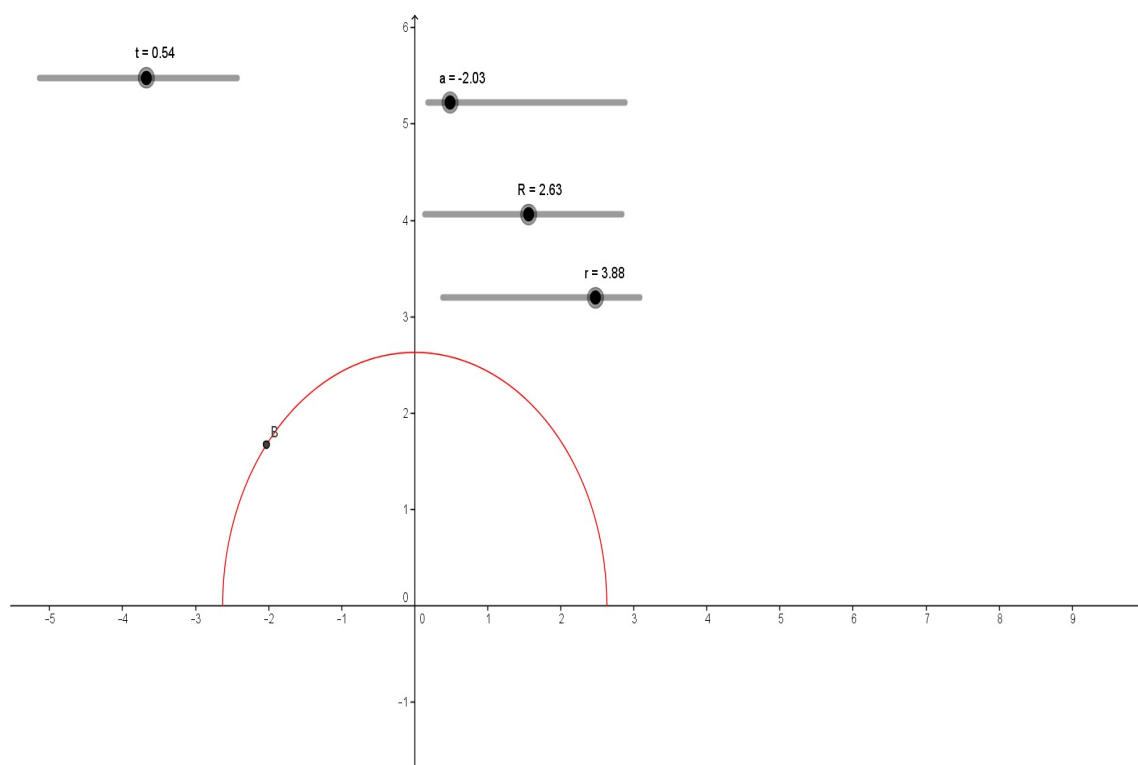


Figura 1.2: circunferência gerada da forma $x^2 + y^2 = 4$ construída no Geogebra.

1.3 Polinômios de Bernstein

Basicamente, um polinômio de Bernstein é capaz de aproximar uma função contínua f definida em um intervalo fechado e limitado $[a, b]$, bem como ajustar curvas ou superfícies.

Estes polinômios também permitem uma aproximação com conjunto de dados. Mas não necessariamente reproduz exatamente o conjunto de dados. Isto é, o gráfico da função de aproximação não passa através dos dados, mas perto deles. Como ferramenta de ajuste, estes polinômios têm várias vantagens, principalmente no desenho assistido por computador (CAD) [5].

Há circunstâncias ou experiências que devem encontrar ou manter uma boa aproximação em vez de procurar interpolação dos dados. Os polinômios de Bernstein são chamadas assim porque são a base fundamental para a prova de Bernstein que conduziu ao teorema da aproximação de Weierstrass (ver apêndice B). Esta demonstração tem um profundo impacto em muitas áreas. Os polinômios de Bernstein estão ligados com a teoria de Probabilidade, com problemas de momentos e com a teoria de somas de séries divergentes.

O cálculo da expansão binomial

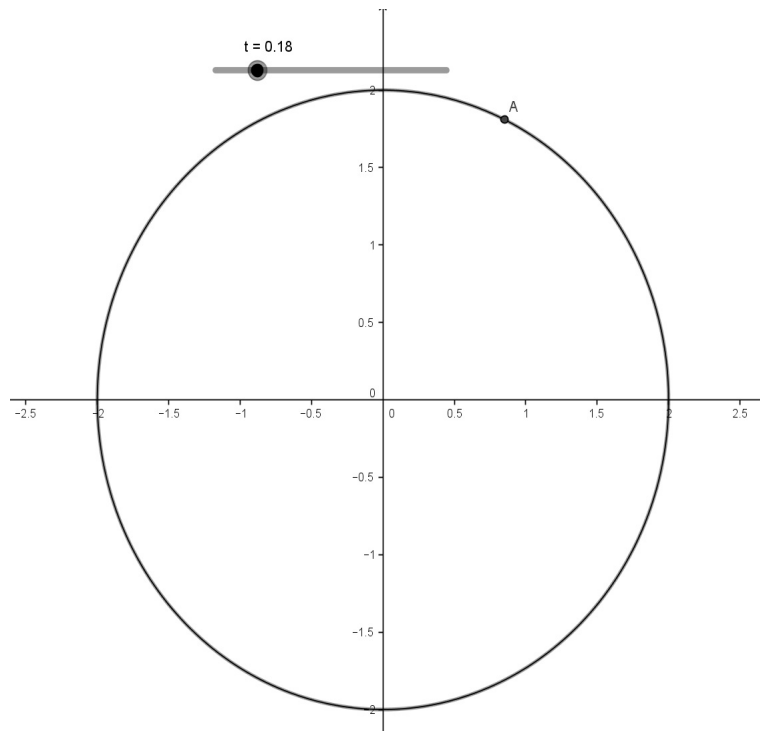


Figura 1.3: circunferência gerada da forma $x = 2 \cos(2\pi t)$ e $y = 2 \sin(2\pi t)$.

$$1 = (t + (1 - t))^n = \sum_{i=0}^n \binom{n}{i} \cdot (1 - t)^{n-i} \cdot t^i$$

com $t \in [0, 1]$, $i = 0, \dots, n$; isto nos permite obter o polinômio de Bernstein em diferentes graus n .

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

com $t \in [0, 1]$, $i = 0, \dots, n$

Bernstein de grau 1, figura 1.6:

$$B_0^1(t) = \binom{1}{0} \cdot (1 - t)^{1-0} \cdot t^0 = 1 - t$$

$$B_1^1(t) = \binom{1}{1} \cdot (1 - t)^{1-1} \cdot t^1 = t$$

Bernstein de grau 2. figura 1.7:

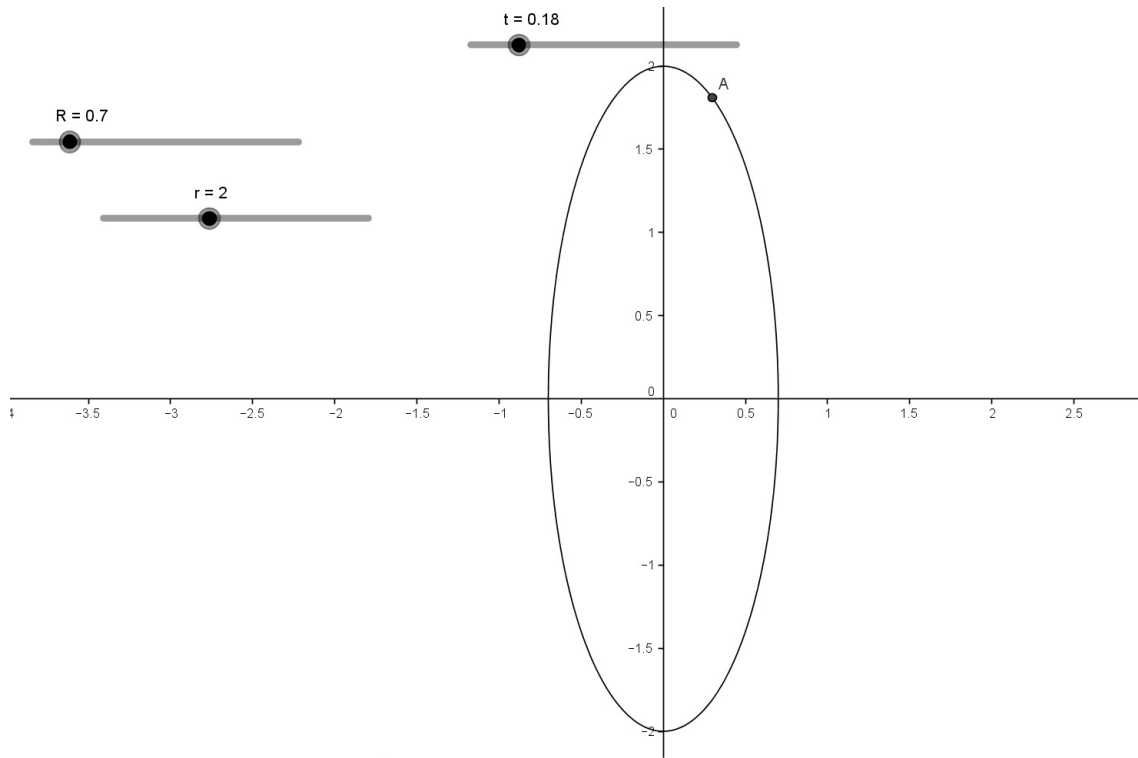


Figura 1.4: Elipse gerada da forma $x = R \cos(2\pi t)$ e $y = r \sin(2\pi t)$.

$$B_0^2(t) = \begin{pmatrix} 2 \\ 0 \end{pmatrix} \cdot (1-t)^2 \cdot t^0 = (1-t)^2$$

$$B_1^2(t) = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \cdot (1-t)^{2-1} \cdot t^1 = 2(1-t)t$$

$$B_2^2(t) = \begin{pmatrix} 2 \\ 2 \end{pmatrix} \cdot (1-t)^{2-2} \cdot t^2 = t^2$$

Bernstein de grau 3, figura 1.8:

$$B_0^3(t) = \begin{pmatrix} 3 \\ 0 \end{pmatrix} \cdot (1-t)^3 \cdot t^0 = (1-t)^3$$

$$B_1^3(t) = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \cdot (1-t)^{3-1} \cdot t^1 = 3(1-t)^2 t$$

$$B_2^3(t) = \begin{pmatrix} 3 \\ 2 \end{pmatrix} \cdot (1-t)^{3-2} \cdot t^2 = 3(1-t)t^2$$

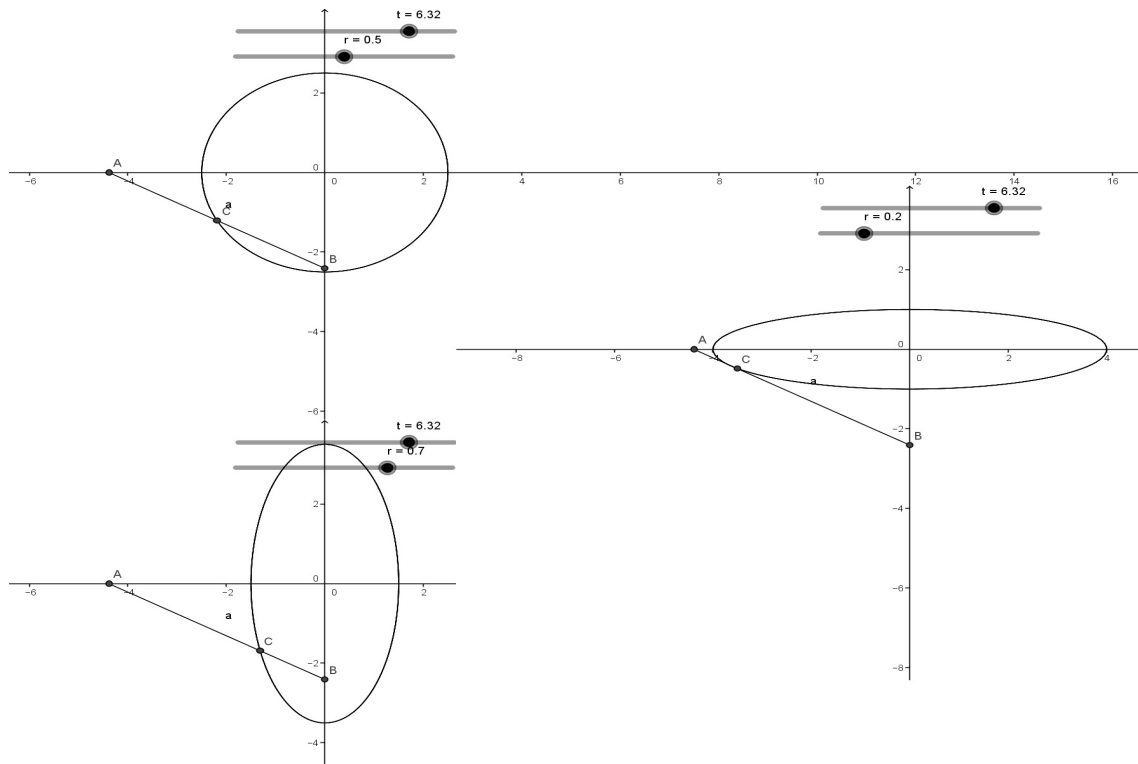


Figura 1.5: A trajetória elíptica ou circular descrita por um ponto fixo na barra dependendo da razão entre os pontos A e B.

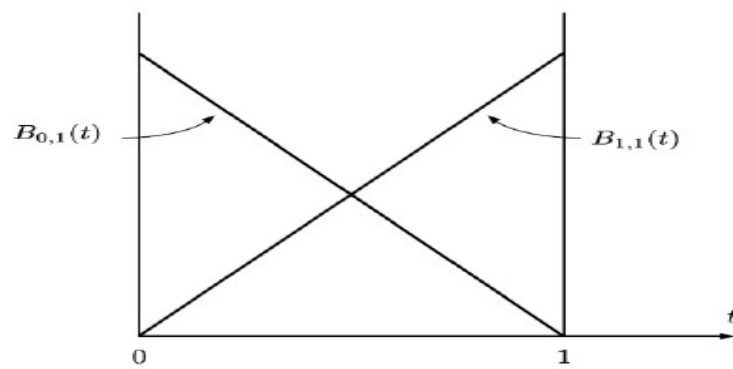


Figura 1.6: Bernstein de grau 1.

$$B_3^3(t) = \binom{3}{3} \cdot (1-t)^{3-3} \cdot t^3 = t^3$$

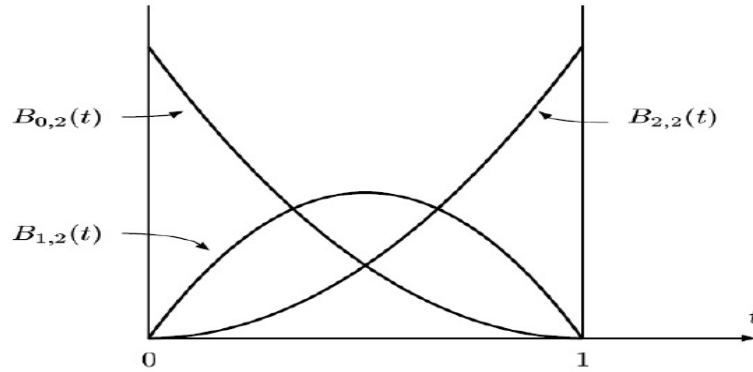


Figura 1.7: Bernstein de grau 2.

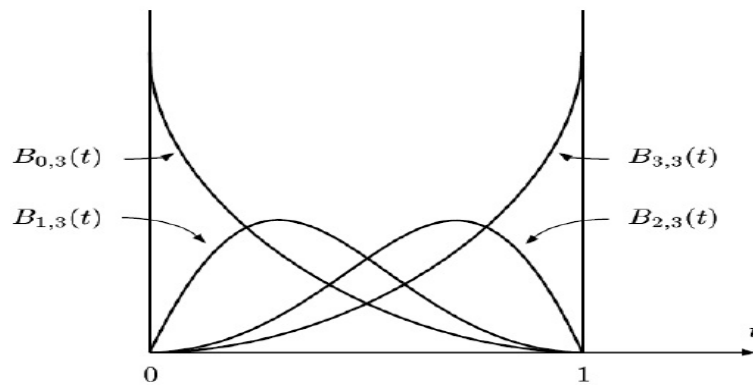


Figura 1.8: Bernstein de grau 3.

1.3.1 Propriedades dos polinômios de Bernstein

Podemos citar algumas propriedades dos polinômios de Bernstein [5]:

- São linearmente independente, de fato, dividindo $\sum_{i=0}^n c_i t^i (1-t)^{n-i} = 0$ por $(1-t)^n$ e usando $s = \frac{t}{1-t}$ obtemos $\sum_{i=0}^n c_i s^i = 0$ o que implica que $c_0 = c_1 = c_2 = \dots = c_n = 0$.
- São Simétricos,

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

Como a expressão $1 = (t + (1-t))^n$ pode ser escrita da seguinte forma:
 $1 = ((1-t) + t)^n$, explica esta propriedade na expansão binomial.

- As raízes são 0 e 1,

$$B_i^n(0) = B_{n-i}^n(1) = \begin{cases} 1 & \text{para } i = 0 \\ 0 & \text{para } i > 0 \end{cases}$$

Verifique que para $t = 0$ os termos t apresentarão valores nulos. Para $t = 1$ os termos $1 - t$ apresentarão valores nulos, isto pode ser facilmente comprovado nos gráficos apresentados anteriormente.

- A soma de todas partições do polinômios de Bernstein possui valor 1, $\sum_{i=0}^n B_i^n(t) = 1$, para todo $t \in \mathbb{R}$. Esta propriedade é facilmente verificada para expressão:

$$1 = (t + (1 - t))^n = \sum_{i=0}^n \binom{n}{i} (1 - t)^{n-i} t^i$$

Logo ao somar a expressão teremos como resultado o valor 1.

- São positivos para $(0, 1)$, $\sum_{i=0}^n B_i^n(t) > 0$, para todo $t \in (0, 1)$
Como $t \geq 0$, $(1 - t) \geq 0$ e os coeficientes binomial são positivos, podemos concluir que $\sum_{i=0}^n B_i^n(t) > 0$.
- Satisfaz a relação de recorrência

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

onde $B_{-1}^n \equiv B_{n+1}^n \equiv 0$ e $B_0^0 \equiv 1$

Esta relação de recorrência vem da identidade:

$$\binom{n+1}{i} = \binom{n}{i-1} + \binom{n}{i}$$

Os polinômios Bernstein, é gerados pela relação de recorrência, e pode ser calculado utilizando um método semelhante ao utilizado para o cálculo do binomial onde os coeficientes estão no triângulo de Pascal. Cada valor é a média ponderada dos dois valores mais próximos acima dele, onde os pesos são t e $1 - t$.

O polinômio de Bernstein segue o esquema de distribuição generalizada abaixo:

$$\begin{array}{ccccccc} & & & B_0^0 & & & \\ & & & B_0^1 & B_1^1 & & \\ & & B_0^2 & B_1^2 & B_2^2 & & \\ & \ddots & \vdots & \vdots & \vdots & \ddots & \\ B_0^n & B_1^n & B_2^n & \dots & B_n^n & & \end{array}$$

passa uma reta horizontal (polinômio de grau zero = constante), por dois pontos passa uma reta (polinômio de primeiro grau), por três pontos passa uma parábola (polinômio de segundo grau), etc. Há muitas maneiras pela qual se pode encontrar o polinômio de grau n que passa pelos $n + 1$ pontos dados, embora ele seja único, ou seja, o polinômio pode ser escrito de diversas formas equivalentes. Talvez a forma mais simples seja a que chamamos *Polinômio interpolador de Lagrange* [7].

$$P_n x = \sum_{i=1}^{n+1} p_i(x) y_i,$$

onde cada p_i é um polinômio de grau n definido por

$$p_i(x) = \frac{(x - x_1)(x - x_2) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_{n+1})}{(x_i - x_1)(x_i - x_2) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_{n+1})}.$$

Como exemplo de aplicação dada a tabela encontrar o polinômio interpolador de Lagrange:

x	$f(x)$
-1	15
0	8
3	-1

Solução:

$$P_3(x) = 15 \times \frac{(x-0)(x-3)}{(-1-0)(-1-3)} + 8 \times \frac{(x+1)(x-3)}{(0+1)(0-3)} - 1 \times \frac{(x+1)(x-0)}{(3+1)(3-0)} = 15 \times \frac{x^2-3x}{4} + 8 \times \frac{(x^2-2x-3)}{-3} - 1 \times \frac{x^2+x}{12}.$$

Agrupando os termos semelhantes, segue que:

$$P_3(x) = x^2 - 6x + 8, \text{ figura 1.9 .}$$

Vamos usar na figura 1.10 para interpolar utilizando o interpolador de Lagrange, figura 1.11 .

1.4.2 Interpolador Linear

Suponhamos que a função $f(x)$ seja conhecida nos pontos $x_i, i = 1, 2, 3, \dots, n + 1$ onde $f(x_i) = y_i$. Temos então n intervalos entre pontos consecutivos (x_i, x_{i+1}) . A maneira mais simples de unir estes pontos por uma linha contínua é traçar uma reta entre cada dois pontos (x_i, y_i) e (x_{i+1}, y_{i+1}) . Vamos chamar de $g_i(x)$ a reta que une os pontos (x_i, y_i) e (x_{i+1}, y_{i+1}) .

A equação da $g_i(x)$ é, então,

$$g_i(x) = y_i + (x - x_i) \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \equiv y_i + (x - x_i) \frac{\Delta y_i}{\Delta x_i}.$$

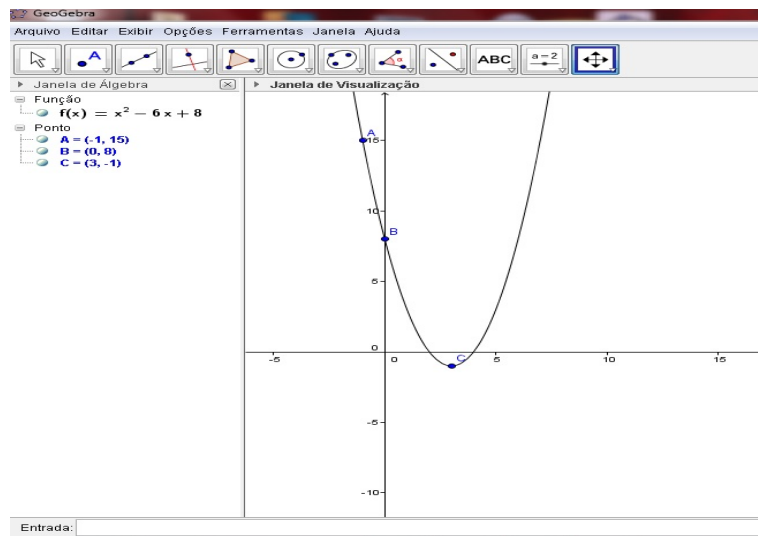


Figura 1.9: Usando o Geogebra para verificar o polinômio interpolador.

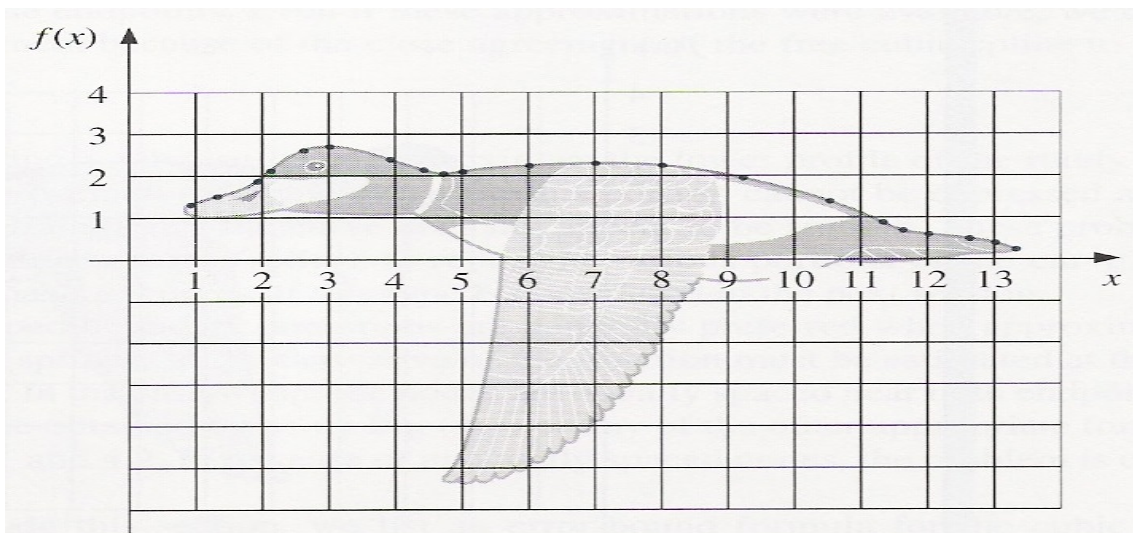


Figura 1.10: Imagem de um pato usada na interpolação.

Usaremos os dois métodos estudados para fazer uma interpolação em uma figura disposta em um plano cartesiano, para realizar a interpolação foi usado o software wxMaxima. O Polinômio mais sinuoso linha vermelha é o interpolador de Lagrange enquanto que a linha Azul é o interpolador linear, figura 2.12 [1] .

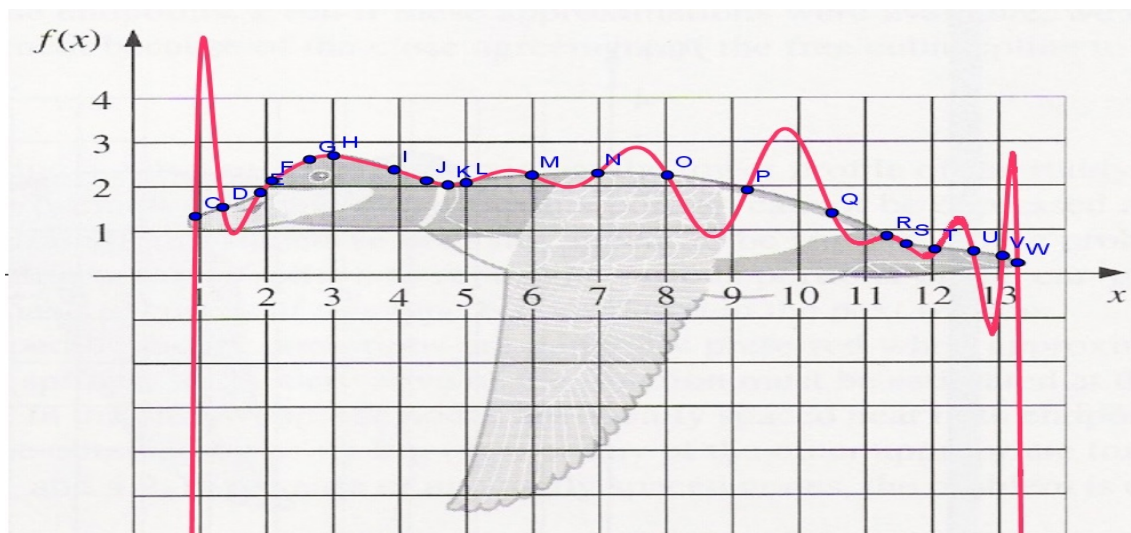


Figura 1.11: Pontos interpolados sobre a imagem do pato pelo polinômio de Lagrange.

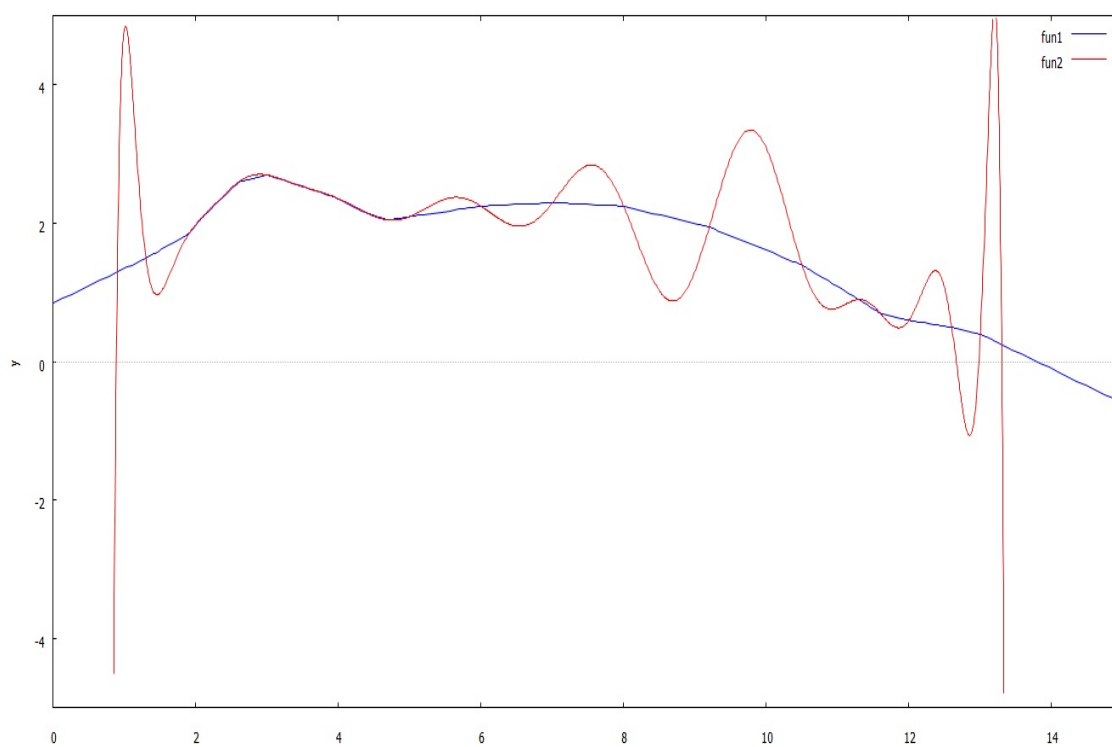


Figura 1.12: Exemplo de interpolação linear usando o software wxMaxima.

Capítulo 2

Curvas de Bézier

2.1 Uma breve história das curvas de Bézier e sua importância

Arquitetos navais renascentistas na Itália foram os primeiros a usar técnicas envolvendo secções cônicas [2]. Antes disso, os navios eram feitos a mão, sem qualquer matemática envolvida. Essas técnicas de design foram refinadas através dos séculos, culminando com o uso de vigas de madeiras dobradas nas formas ideais através de pesos, vem daí a origem do nome "splines". No início do século XX, com o invento dos aviões o uso das secções cônicas foram mais difundidas devido à necessidade de suavizar a trajetória do ar ao longo da superfície da aeronave. O automóvel surgiu como um dos ícones do século XX, precisando de novos projetos que viabilizasse a produção em massa, com o uso de CAD/CAM (desenho auxiliado por computador / Fabricação assistida por computador).

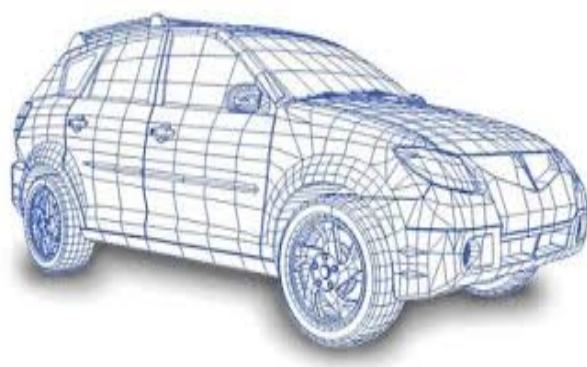


Figura 2.1: Projeto de veículo usando curvas de Bézier.

No final da década de 1950, tornou-se possível a usinagem em 3D de blocos de

madeira ou aço, na construção dos moldes. Estes moldes poderiam, então, ser usados como matrizes para estampagem de peças como o capô e a lataria de carros. O grande problema destes métodos de produção foi a falta de software adequado que permitisse a comunicação com as máquinas. A fim realizar desenhos em computadores, geralmente contamos com linhas e curvas. As linhas são fáceis de desenhar dando ao computador o primeiro e último ponto temos informação suficiente para gerar os pontos colineares aos pontos dados. As curvas, no entanto possuem um grau de complexidade maior, embora possamos desenhar curvas com facilidade à mão livre, nos computadores estas curvas precisam ser definidas como uma função matemática, mas nem todas as funções que conhecemos podem se adequar a maioria dos problemas de computação gráfica. Outro problema encontrado era que as curvas precisavam ser representadas numericamente. O uso de formas paramétricas surgiram como solução, pois até então não era possível gerar curvas muito complexa que pudessem ser interpretadas por máquinas de controle numérico os CNC's.

As curvas de Bézier foram bastante promissoras e são usadas na maioria das situações em que se requer o desenho de uma curva. A homenagem ao engenheiro Francês Pierre Bézier(1910-1999), se deve ao fato dele ser o principal responsável por levá-las ao conhecimento de todos através de seu trabalho de design (trabalhando para a Renault durante muitos anos e publicando suas investigações em 1962), embora ele não tenha sido o primeiro. Podemos ser tentados a dizer que o matemático Paul De Casteljaeu foi o primeiro, investigando a natureza dessas curvas em 1959, enquanto trabalhava na Citroën, no entanto, De Casteljaeu não publicou seu trabalho.

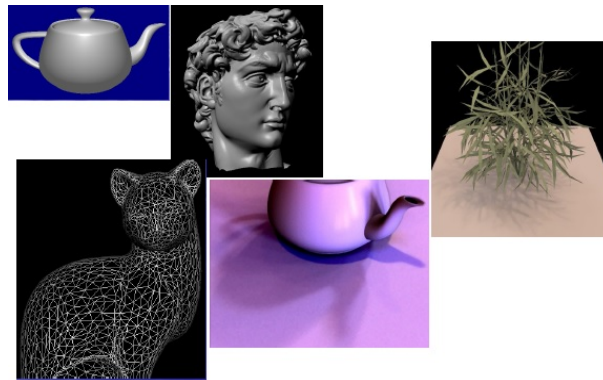


Figura 2.2: Formas geradas por alguns software usando as curvas de Bézier.

As curvas Bézier foram desenvolvidas de forma independente por De Casteljaeu da Citroën e por Bézier na Renault servindo para desenvolvimento de um método de produção de automóveis, fazendo com que estas empresas possuíssem um design mais suave e agradável em seus produtos. Independente de quem criou o que importa é que essas curvas são bastante úteis, pois ligando várias curvas de Bézier sua combinação

gera uma única curva suave. Ao desenhar em softwares como o Photoshop, Flash ou Inkscape, as curvas desenhadas são simplesmente Curvas de Bézier, figura 2.2 .

As Curvas de Bézier se diferenciam de outras técnicas na época por terem uma grande flexibilidade e alta capacidade de adaptação. Elas podem ser construídas em qualquer comprimento e diversidade de forma, unindo o ponto de extremidade de uma curva ao ponto de início de uma outra. Estas curvas podem expandir e se transformar em superfícies de Bézier.

Entre as aplicações recentes, temos o uso de software em cirurgias plásticas onde o paciente pode visualizar os contornos antes do procedimento, figura 2.3.

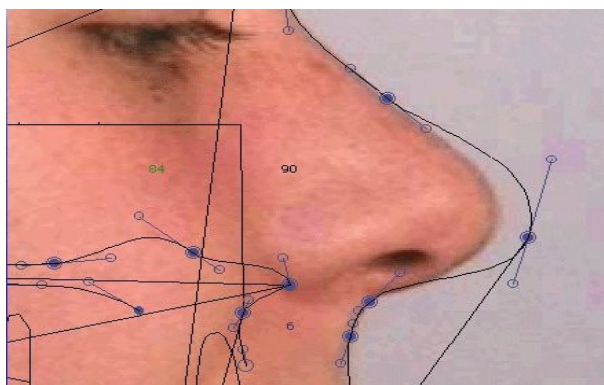


Figura 2.3: Usando curvas em cirurgias plásticas.

O Processo de construção da curva de Bézier é bastante intuitivo, levando o usuário a realizar uma construção sem grandes conhecimentos, isto era favorável, pois os designs não dominavam os conceitos de matemática. Ao longo das décadas a computação gráfica se tornou uma grande ferramenta para construções geométricas. No âmbito de ensino o estudo das curvas de Bézier permite revisitar alguns conceitos como proporcionalidade, geometria analítica, parametrização de curvas, médias, funções e polinômios. Aplicações como criação de ambientes virtuais, construção de desenhos animados, ajustamento de curvas, reconstrução de figuras em 3D e na medicina em processamento de imagens podem ser citadas, o que é bastante motivador observar os conceitos matemáticos sendo utilizados no mundo real.

Ao longo do trabalho estudamos os conceitos básicos que levaram ao seu desenvolvimento como os polinômios de Bernstein e curvas paramétricas. Faremos uma abordagem sobre as propriedades das curvas de Bézier, e casos especiais. Com os conceitos básicos estudados no capítulo anterior definimos as Curvas de Bézier e desenvolvemos as ideias intuitivas para gerar estas curvas. O Conhecimento do Algoritmo De Casteljau facilita a implementação posterior em software matemáticos como wxMaxima e o Geogebra.

Por fim, mostramos exemplos propostos e outros sugeridos no livro de referência [2]. A computação gráfica é um combustível motivador para o estudo das Curvas de

Bézier, está atualizado com as bases deste tema traz uma segurança para abordá-lo durante as aulas. O tema revisa alguns conteúdos matemáticos básicos, e serve bem como aplicação do conhecimento. Fazer uma comparação da curva de Bézier com outros métodos de interpolação como o Polinômio de Lagrange e a interpolação linear é bastante relevante para o estudo, pois traz uma resposta clara pela predileção por essas curvas.

2.2 Definição das Curvas de Bézier

Essencialmente o que a construção de De Casteljau faz é uma média ponderada entre diversos pontos dados inicialmente. Os pesos desta média em cada ponto são funções do parâmetro t ; como em cada etapa faz-se uma interpolação linear sobre os pontos da etapa anterior essas funções são polinômios de grau n , verifique que temos $n + 1$ pontos e o polinômio é um grau menor que o número de pontos [9].

As curvas de Bézier podem ser definidas como:

$$C(t) = \sum_{i=0}^N P_i B_i^n(t)$$

em que B_i^n são os polinômios de Bernstein de grau n e P_i são os pontos de controle escolhidos previamente. Temos então:

$$x(t) = \sum_{i=0}^N x_i B_i^n(t)$$

$$y(t) = \sum_{i=0}^N y_i B_i^n(t)$$

Problema: Ache a curva de Bezier que possui os pontos de controle: $(x, y) = (2, 2), (1, 1.5), (3.5, 0), (4, 1)$

Solução:

$$\begin{aligned} x(t) &= 2B_0^3(t) + 1B_1^3(t) + 3.5B_2^3(t) + 4B_3^3(t) = \\ &= 2t^3 + t^2(1-t) + 3.5t(1-t)^2 + 4(1-t)^3 = \\ &= \frac{t^3 + 12t^2 - 17t + 8}{2}. \\ y(t) &= 2B_0^3(t) + 1.5B_1^3(t) + 0B_2^3(t) + 1B_3^3(t) = \\ &= 2(t^3) + 1.5(t^2)(1-t) + 0t((1-t)^2) + 1((1-t)^3) = \\ &= \frac{t^3 - 9t^2 + 6t - 2}{2}. \end{aligned}$$

2.2.1 Propriedades das Curvas de Bézier

Algumas propriedades das curvas de Bézier [2] são:

- **Invariância a transformações afins:** Interpolação linear é invariante a transformações afins. A curva de Bézier é gerada por interpolações lineares alinhadas. Logo, também é invariante a transformações afins. Ou seja, calcular um determinado ponto na curva de Bézier original e, em seguida, aplicar uma transformação afim ou aplicar esta mesma transformação afim a todos os pontos de controle e, em seguida, gerar o ponto na curva de Bézier, não fará diferença alguma.
- **Invariância a transformações afins de parâmetros:** Muitas vezes, se pensa que uma curva de Bézier é definida ao longo do intervalo $[0, 1]$. Isto é feito porque ele é conveniente, não porque é necessário. O algoritmo De Casteljau é "cego" para o intervalo real, pois é a curva é definida usando proporções. Pode-se, portanto, pensar na curva como sendo definida sobre qualquer intervalo arbitrário $a < u < b$ na reta real, onde $t = \frac{u-a}{b-a}$. Esta propriedade é herdada a partir do processo de interpolação linear. Novamente, a interpolação linear é invariante a transformações afins de parâmetro. E assim é, também, a curva de Bézier. Ou seja, podemos definir uma Curva de Bézier no intervalo $[a, b]$ da seguinte maneira:

$$b_i^r(u) = \frac{b-u}{b-a} b(u)_i^{r-1} + \frac{u-a}{b-a} b(u)_{i+1}^{r-1}.$$

Esta propriedade é particularmente interessante para compor objetos com mais de uma curva de Bézier que tenha alguma propriedade particular desejada.

- **Propriedade do invólucro convexo:** Para $t \in [0, 1]$, $b^n(t)$ localize-se dentro ou na borda do invólucro convexo da poligonal de controle.

Isto ocorre uma vez que cada b_r intermediário é obtido como uma combinação baricêntrica convexa anterior b_j^{r-1} em nenhum passo do algoritmo De Casteljau que produz pontos fora do invólucro convexo de b_i . Uma simples consequência da propriedade fecho convexo é que o controle planar do polígono sempre gera uma curva plana. A importância da propriedade fecho convexo está no que é conhecido como verificação de interferência. Suponha que queremos saber se duas curvas Bézier se cruzam, por exemplo, cada um pode representar o caminho de um braço de robô, e nosso objetivo é ter certeza de que os dois caminhos não se interceptam, evitando assim colisões dos robôs. Em vez de realmente calcular uma possível intersecção, podemos realizar um teste muito mais simples: circunscrever a menor caixa possível em torno de cada curva de modo que suas bordas sejam paralelas ao sistema de coordenadas. Tais caixas

são chamadas Minmax, e são criadas pelas coordenadas mínimas e máximas do polígono de controle. Claramente cada caixa contém o seu polígono de controle, seu invólucro convexo e a curva de Bézier correspondente, figura 2.4. Podemos verificar se as duas caixas estão sobrepostas, tendo assim a certeza de que as duas curvas não se cruzam. Se as caixas se sobrepõem, nós teríamos que realizar uma modificação nos pontos de controle das curvas.

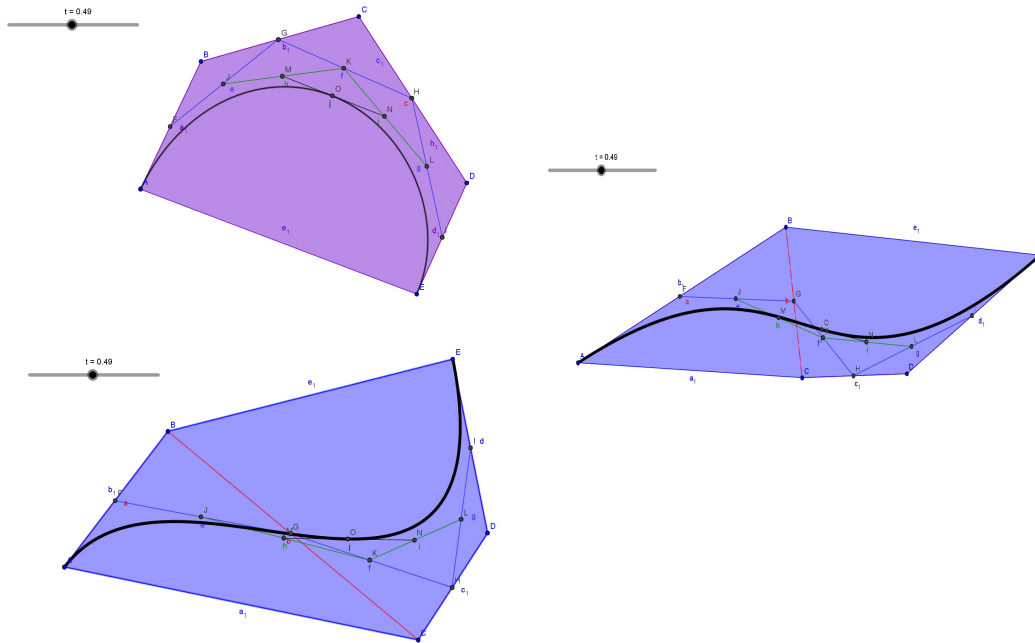


Figura 2.4: Invólucros convexo nas curvas Curvas de Bézier variando os pontos de controle.

- **A curva de Bézier interpola sempre os extremos:** a curva de Bézier passa sempre através dos extremos b_0 e b_n , tendo como base a propriedade das raízes dos polinômios de Bernstein temos: $b^n(0) = b_0$, $b^n(1) = b_n$, os dois pontos extremos da curva são os mais importantes na construção, e a inversão de posição não interfere no formato da curva.

2.2.2 Algoritmo de Casteljau

O algoritmo descrito é fundamental para o domínio da curva de Bézier, no entanto é surpreendentemente simples. Seu principal atrativo é a bela interação entre geometria e a álgebra [2] [5].

Descrição do algoritmo:

Entrada: um conjunto n pontos p_i^0 , e um valor de $t \in \mathbb{R}$.

Saída: Um ponto p_0^n calculado através de interpolações sucessivas avaliadas em t .

DECASTELJAU(p_i^0, t)

(1) faça $r = 1$ até n

(2) faça $i = 0$ até $n - r$

(3) $p_i^r \leftarrow (1 - t)p_i^{r-1} + tp_{i+1}^{r-1}$.

Os pontos obtidos na saída representam a curva de Bézier e os que foram usados na entrada recebe o nome de polígono de controle, os pontos intermediários são bastantes úteis na implementação de superfícies, figuras 2.5, 2.6 e 2.7. Usualmente o valor de $t \in [0, 1]$.

Podemos representar o algoritmo segundo o esquema:

$$\begin{array}{ccccccc} p_0 & & & & & & \\ p_1 & p_0^1 & & & & & \\ p_2 & p_1^1 & p_0^2 & & & & \\ \vdots & \vdots & \vdots & \ddots & & & \\ p_n & p_{n-1}^1 & p_{n-2}^2 & \cdots & p_0^n & & \end{array}$$

A regra do algoritmo de Casteljau é tomar sempre a média ponderada de dois pontos, gerando um ponto na próxima linha conforme o esquema apresentado acima. O algoritmo finaliza quando o último ponto é gerado, o mesmo procedimento é repetido para vários valores de t no intervalo $[0, 1]$, assim obtemos o perfil da curva.

No exemplo encontrado em Farin [2]: Dados os pontos $(0, 0)$, $(0, 2)$, $(8, 2)$ e $(4, 0)$ com $t = \frac{1}{2}$, temos o esquema representado acima

$$\begin{array}{ccccccc} \begin{bmatrix} 0 \\ 0 \end{bmatrix} & & & & & & \\ & \begin{bmatrix} 0 \\ 2 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \end{bmatrix} & & & & \\ & & & \begin{bmatrix} 8 \\ 2 \end{bmatrix} & \begin{bmatrix} 4 \\ 2 \end{bmatrix} & \begin{bmatrix} 2 \\ \frac{3}{2} \end{bmatrix} & \\ & & & & \begin{bmatrix} 4 \\ 0 \end{bmatrix} & \begin{bmatrix} 6 \\ 1 \end{bmatrix} & \begin{bmatrix} 5 \\ \frac{3}{2} \end{bmatrix} & \begin{bmatrix} \frac{7}{2} \\ \frac{3}{2} \end{bmatrix} \end{array}$$

Verifique que todos os pontos que aparecem na figura foram calculados no esquema do exemplo, figura 2.8.

2.2.3 Cálculo da Derivada em relação a Curva de Bézier

Sabemos que a Curva de Bézier é definida por $n+1$ pontos de controle $P_0, P_1, \dots, P_{n+1}$ e tem a seguinte equação:

$$C(u) = \sum_{i=0}^n P_i B_i^n(t), \quad a < u < b, \quad t = \frac{u-a}{b-a}.$$

em que $B_i^n(t)$ é definida por:

$$B_i^n(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i}.$$

Uma vez que os pontos de controle são constantes e independentes da variável t , o cálculo da derivada da curva $C'(t)$ se reduz ao cálculo das derivadas de $B_i^n(t)$. Com algumas manipulações algébricas simples, temos o seguinte resultado para :

$$\frac{d}{du} B_i^n(t) = B_i'^n(t) = n(B_{i-1}^{n-1}(t) - B_i^{n-1}(t)).$$

Então, calculando a derivada da curva $C(u)$ temos o seguinte:

$$\frac{d}{dt} C(t) = C'(t) = \sum_{i=0}^{n-1} B_i^{n-1}(t) n(P_{i+1} - P_i).$$

Usamos $\Delta p_0 = n(P_1 - P_0)$, $\Delta p_1 = n(P_2 - P_1)$, $\Delta p_2 = n(P_3 - P_2)$, ..., $\Delta p_{n-1} = n(P_n - P_{n-1})$, a expressão acima se reduz ao seguinte:

$$C'(t) = \sum_{i=0}^{n-1} B_i^{n-1}(t) \Delta p_i.$$

Dados os pontos $(0,0)$, $(0,2)$, $(8,2)$ e $(4,0)$ com $t = \frac{1}{2}$, encontrar a primeira derivada para o valor de $t = \frac{1}{2}$, figura 2.9 :

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ 2 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 2 \end{bmatrix}$$

$$\begin{bmatrix} 8 \\ 2 \end{bmatrix} \quad \begin{bmatrix} 8 \\ 0 \end{bmatrix} \quad \begin{bmatrix} 4 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 4 \\ 0 \end{bmatrix} \quad \begin{bmatrix} -4 \\ -2 \end{bmatrix} \quad \begin{bmatrix} 2 \\ -1 \end{bmatrix} \quad \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

Uma alternativa para o cálculo da derivada seria usando a diferença $p_1^2 - p_0^2$:

$$\begin{bmatrix} 5 \\ \frac{3}{2} \end{bmatrix} - \begin{bmatrix} 2 \\ \frac{3}{2} \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

Para calcular derivadas de ordem superior usamos esta expressão para o operador de diferença:

$$\Delta^r p_j = \Delta^{r-1} p_{j+1} - \Delta^{r-1} p_j.$$

Listamos alguns exemplos:

$$\Delta^0 p_i = p_i$$

$$\Delta^1 p_i = p_{i+1} - p_i$$

$$\Delta^2 p_i = p_{i+2} - 2p_{i+1} + p_i$$

$$\Delta^3 p_i = p_{i+3} - 3p_{i+2} + 3p_{i+1} - p_i.$$

Este fator a direita são os coeficientes binomiais, formando um triângulo de Pascal, este padrão se mantém e podemos generalizar:

$$\Delta^r p_i = \sum_{j=0}^r \binom{r}{j} (-1)^{r-j} p_{i+j}$$

A r-ésima derivada da curva de Bézier é dada por:

$$\frac{d^r}{dt^r} C^n(t) = \frac{n!}{(n-r)!} \sum_{j=0}^{n-r} \Delta^r p_j B_j^{n-t}(t).$$

Observamos dois casos especiais para $t = 0$ e $t = 1$, onde obtemos:

$$\frac{d^r}{dt^r} C^n(0) = \frac{n!}{(n-r)!} \Delta^r p_0$$

e

$$\frac{d^r}{dt^r} C^n(1) = \frac{n!}{(n-r)!} \Delta^r p_{n-r}.$$

2.2.4 Cálculo da integral da Curva de Bézier

A curva de Bézier representada por:

$$C(u) = \sum_{i=0}^n p_i B_i^n(t), t = \frac{u-a}{b-a}$$

em que a integral é dada por [2] [5]:

$$c(u) = \int C(u) du = \sum_{i=0}^{n+1} c_i B_i^{n+1}(t)$$

onde

$$c_i = c_{i-1} + \frac{b-a}{n+1} p_{i-1} =$$

$$c_0 + \frac{b-a}{n+1} (p_0 + p_1 + \dots + p_{i-1}), i = n+1, \dots, 1$$

e c_0 é a constante de integração, isto pode ser verificado derivando $c(u)$. A partir da expressão anterior e da $C(u)$ que interpola os extremos p_0 e p_n se deduz a seguinte igualdade:

$$\int C(u) du = \frac{b-a}{n+1} (p_0 + p_1 + \dots + p_n)$$

e em particular:

$$\int_0^1 B_{i=0}^n(t) dt = \frac{1}{n+1}$$

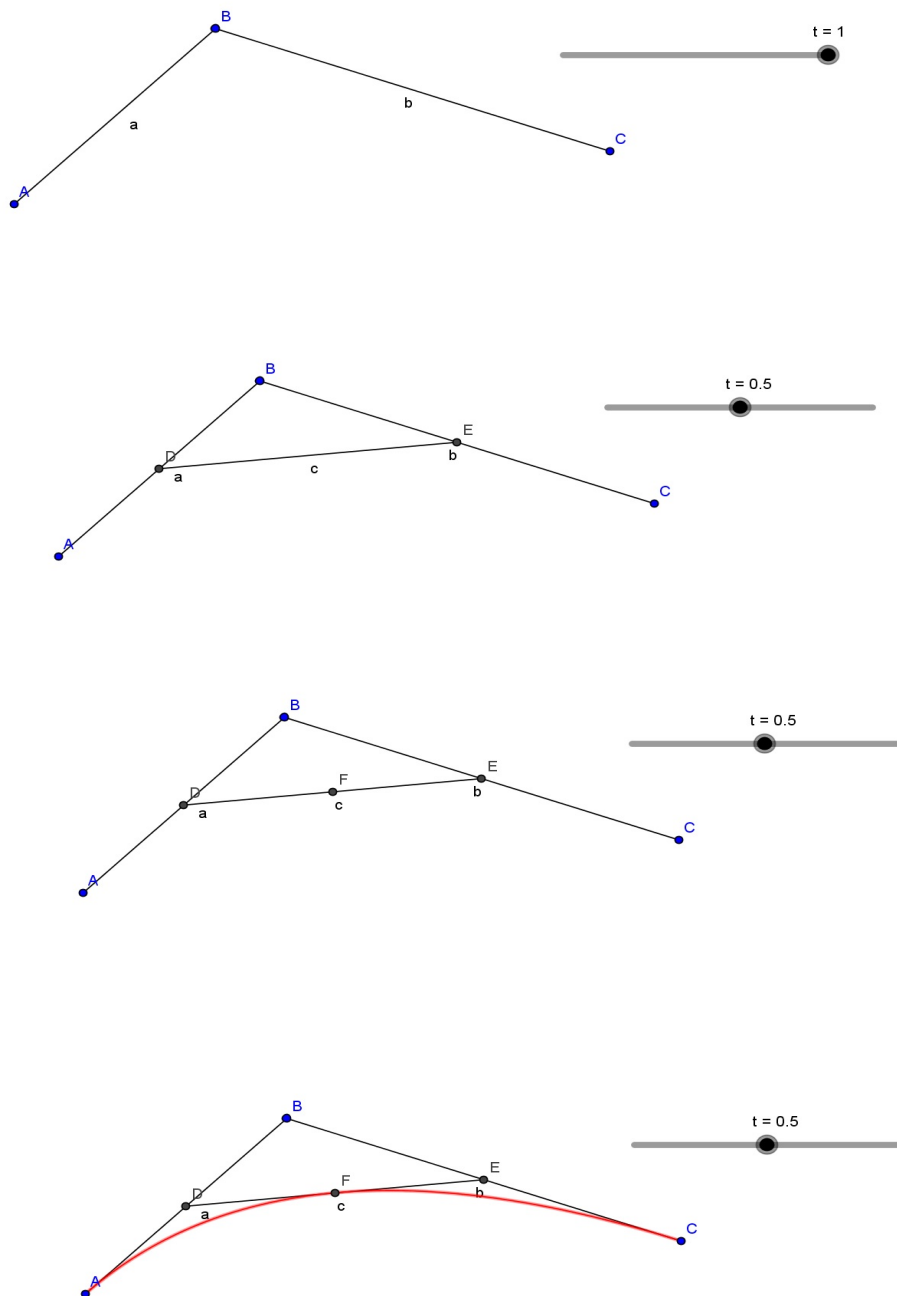


Figura 2.5: Pontos da curva com três Pontos de controle gerada segundo o algoritmo de Casteljau e habilitando o lugar geométrico no Geogebra

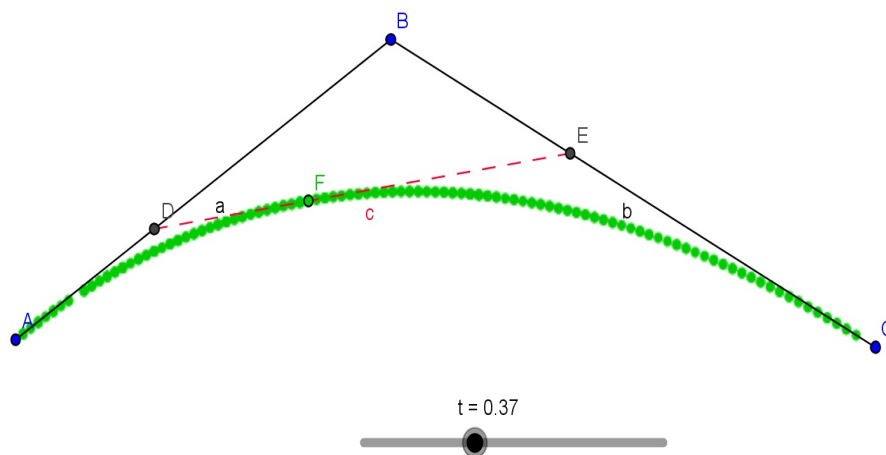


Figura 2.6: Curvas de Bezier com três usando o rastro no geogebra.

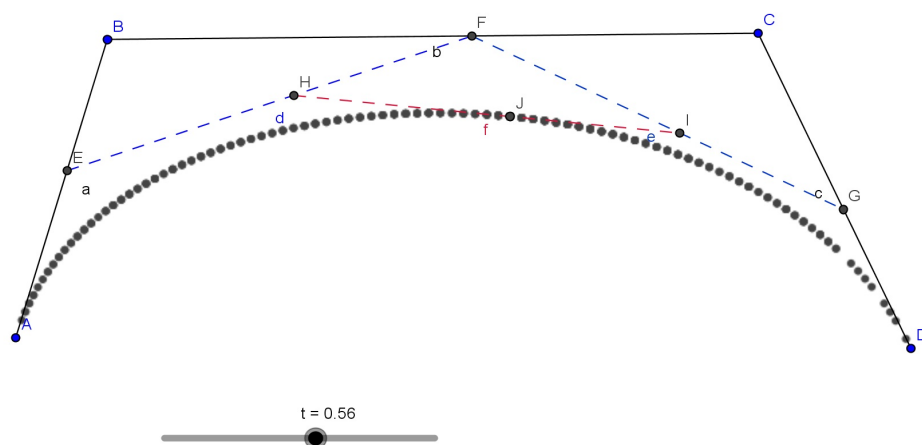


Figura 2.7: Curvas de Bézier com quatro pontos de controle usando o rastro.

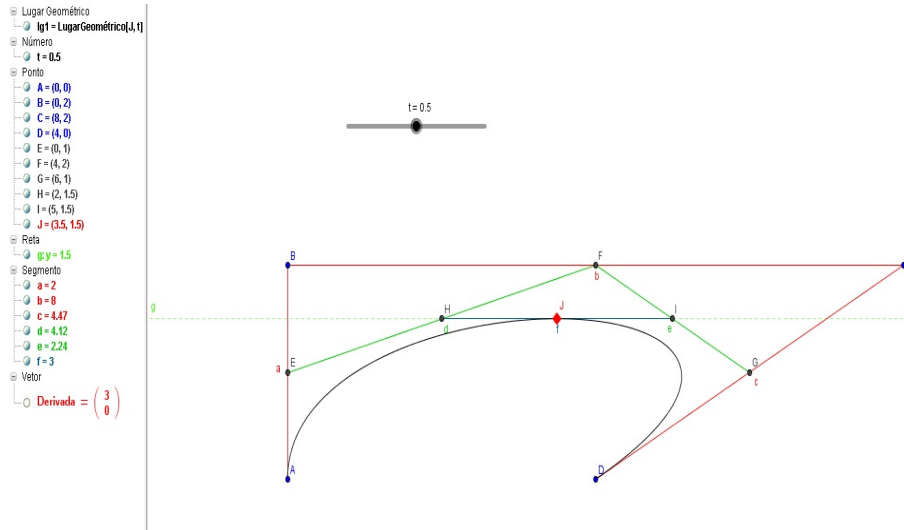


Figura 2.8: Pontos gerados pelo algoritmo De Casteljau com $t = 0.5$.

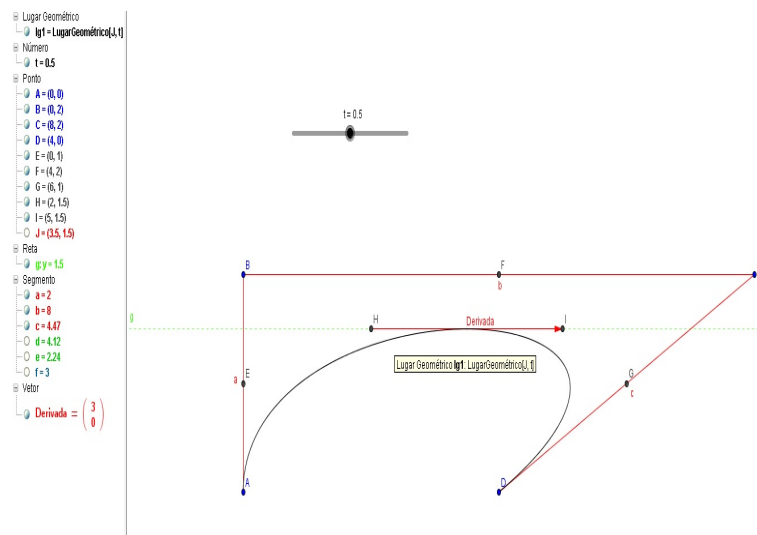


Figura 2.9: Derivada da Curva gerada pelos pontos $(0,0)$, $(0,2)$, $(8,2)$ e $(4,0)$.

Capítulo 3

Exemplos

Neste capítulo resolvemos alguns exemplos propostos sobre Curvas de Bézier, dois destes estão no livro de referência [2] .

3.1 Exemplos

Vamos apresentar alguns exemplos de construções e aplicações das curvas de Bézier.

3.1.1 Exemplo 1

Como primeiro exemplo vamos detalhar os passos para a construção da Curva de Bézier usando o Geogebra.

Como primeiro exemplo usamos um polígono de controle com 17 pontos escolhidos aleatoriamente no Geogebra usando o mouse para determinar os pontos, como vemos na figura 3.1. Concluída a escolha dos pontos, implementamos o algoritmo De Casteljau nas planilhas disponível do software usando o parâmetro t para determinar os pontos ao longo da curva de Bézier.

Para concluir e desenhar a curva usamos o comando lugar geométrico do último ponto gerado na planilha em função parâmetro t observado na figura 3.3. Podemos verificar a movimentação dos pontos gerados no algoritmo de Casteljau ao variar o parâmetro t com incrementos de 0.2, na figura 3.2.

3.1.2 Exemplo 2

No exemplo 2 temos uma implementação da curva usando três coordenadas no espaço. Os pontos escolhidos são: $(2, 0, 0)$, $(-1, -3, 2)$, $(-, 3, 2)$ e $(-2, 0, 0)$. Estes pontos seguem uma dada simetria e foram utilizados para gerar um feito interessante

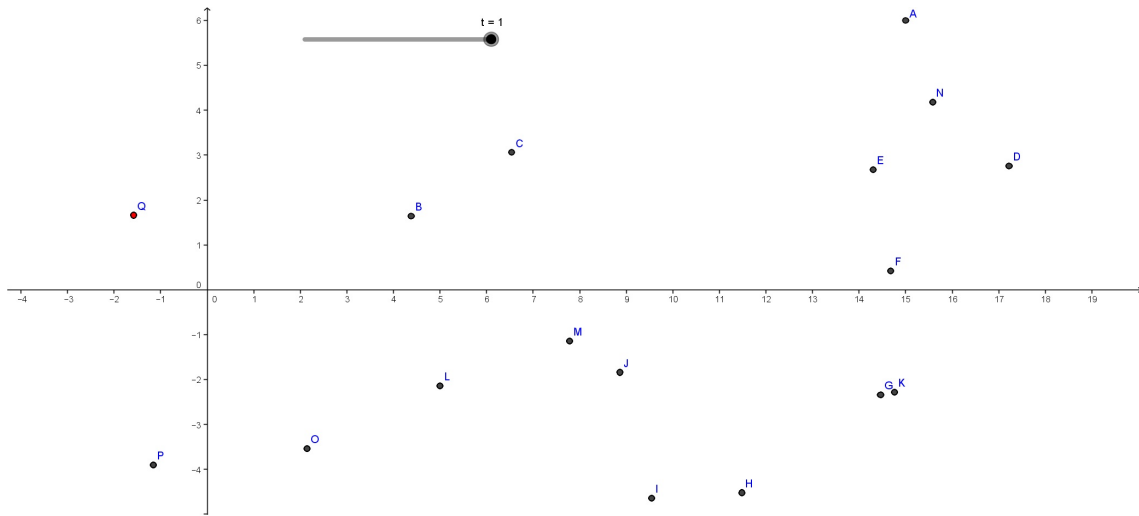


Figura 3.1: Polígono de controle formado por 17 pontos.

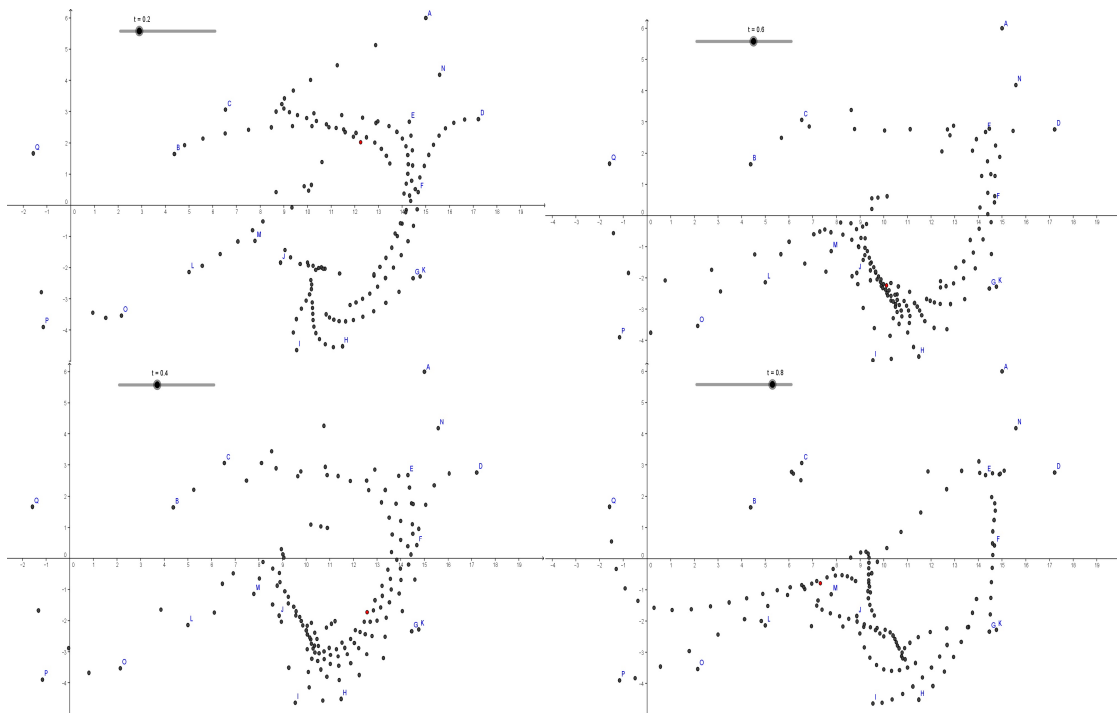


Figura 3.2: Polígono de controle formado por 17 pontos variando o t com incrementos de 0.2.

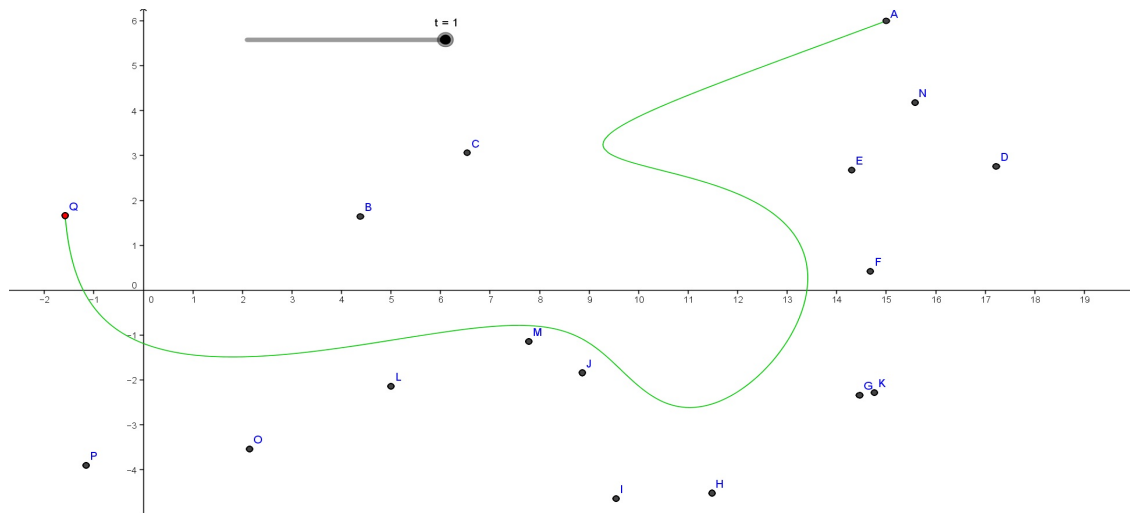


Figura 3.3: Poígono de controle formado por 17 pontos $t = 0.5$.

na curva no espaço. Usamos as planilhas do software Geogebra para gerar os pontos intermediários, e o lugar geométrico para determinar a curva mostrada na figura 3.4.

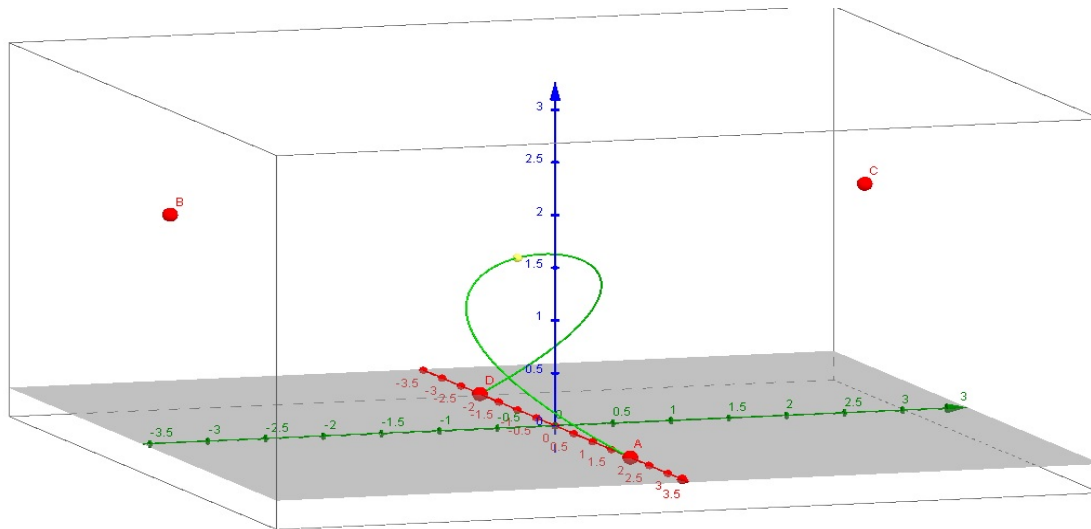


Figura 3.4: Curva de Bézier usando três coordenadas no espaço.

3.1.3 Exemplo 3

Nas figuras 3.5, 3.6 e 3.7 tentamos gerar um espiral usando como pontos de controle os vértices de figuras pentagonal inscritas.

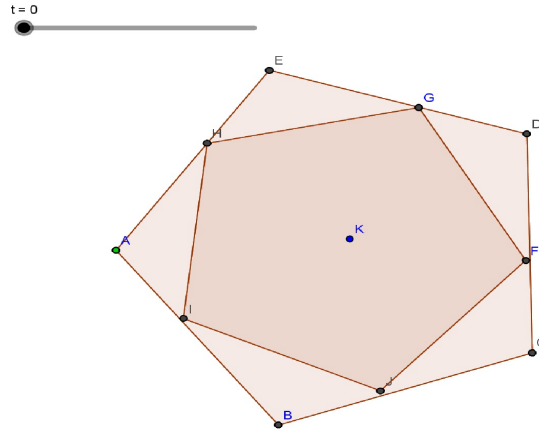


Figura 3.5: Gerando o Polígono de controle com os dois pentágonos.

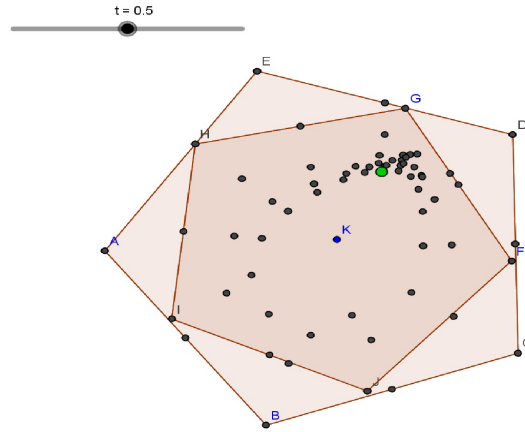


Figura 3.6: Implementando o algoritmo de Casteljau e variando t .

3.1.4 Exemplo 4

O método para gerar a curva é idêntico ao utilizado no exemplos anterior, em particular a figura 3.8 possui uma sequência de triângulos equiláteros e os vértices estão localizados no ponto médio do antecessor.

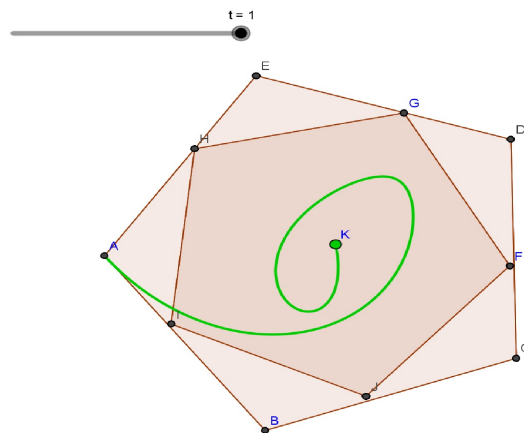


Figura 3.7: Usando o lugar geométrico para gerar a Curva de Bézier em Espiral.

3.1.5 Exemplo 5

Este exemplo foi retirado do livro do Farin [2] capítulo 4. Suponha uma curva de Bézier planar tem um polígono de controle que é simétrico com respeito ao eixo y . Esta curva também é simétrica em relação ao o eixo y ? Considerar o polígono de controle $(-1, 0)$, $(0, 1)$, $(1, 1)$, $(0, 2)$, $(0, 1)$, $(-1, 1)$, $(0, 2)$, $(0, 1)$ e $(1, 0)$. Ao gerar a curva sugerida no problema veja que não existe uma simetria com o eixo y , figuras 3.9 e 3.10 .

3.1.6 Exemplo 6

O exemplo das figura 3.11, 3.12 e 3.13, é bastante prático, inserimos uma figura no nosso caso um pato em pleno voo, escolhemos alguns pontos para gerar a curva. A implementação é feita utilizando o algoritmo de Casteljau como nos casos anteriores, e usamos o lugar geométrico para determinar a Curva observada na figura 3.13 . A silhueta formada pela curva é bastante próxima do perfil aerodinâmico do pato.

3.1.7 Exemplo 7

Mais um exemplo encontrado em Farin [2] capítulo 5 que trata de derivada. Considere a curva de Bézier cúbica dada pelos pontos de controle $(1, 0)$, $(-1, 1)$, $(1, 1)$ e $(-1, 0)$. Em $t = 1/2$, esta curva possui um limite: a sua primeira derivada desaparece e mostra uma curva fechada, ao permutar os pontos de A e B mantendo assim um polígono de controle simétrico. Observe na figura 3.14 que a derivada nos pontos

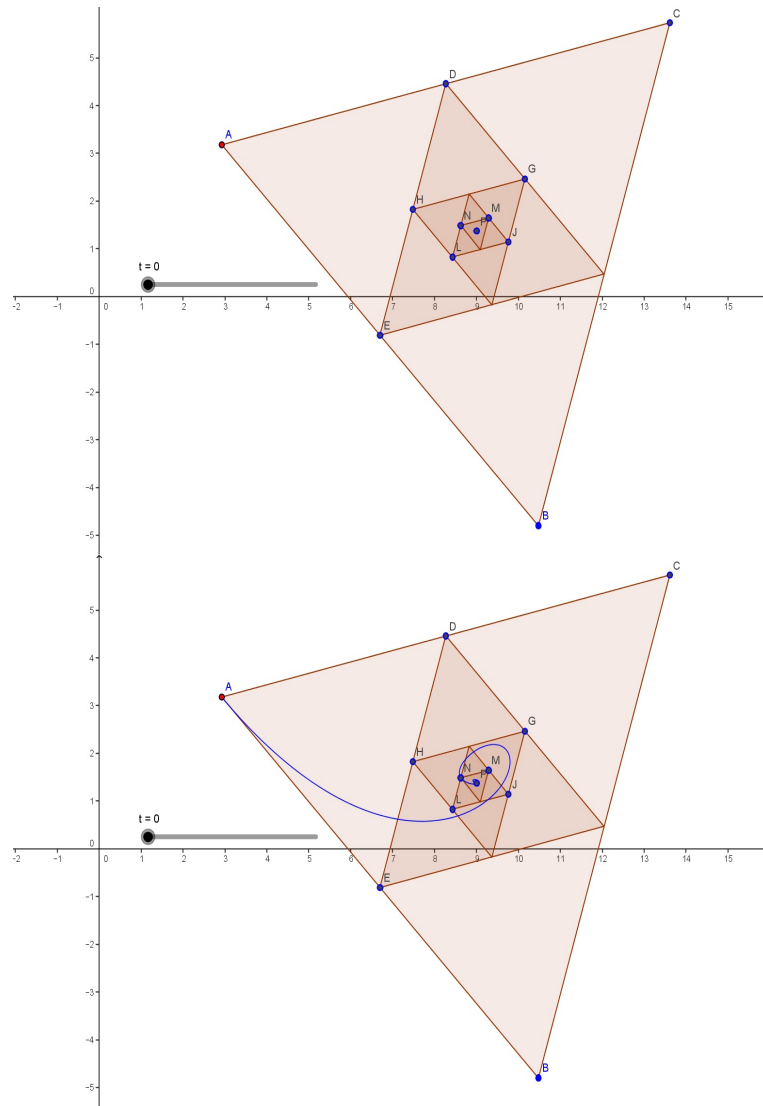


Figura 3.8: Espiral formada por alguns pontos de controle de uma sequência de triângulos.

dados não existe para $t = 0.5$. Gerar outras configurações simétrica onde também não existe derivada para $t = 0.5$, na figura 3.15 e 3.16 .

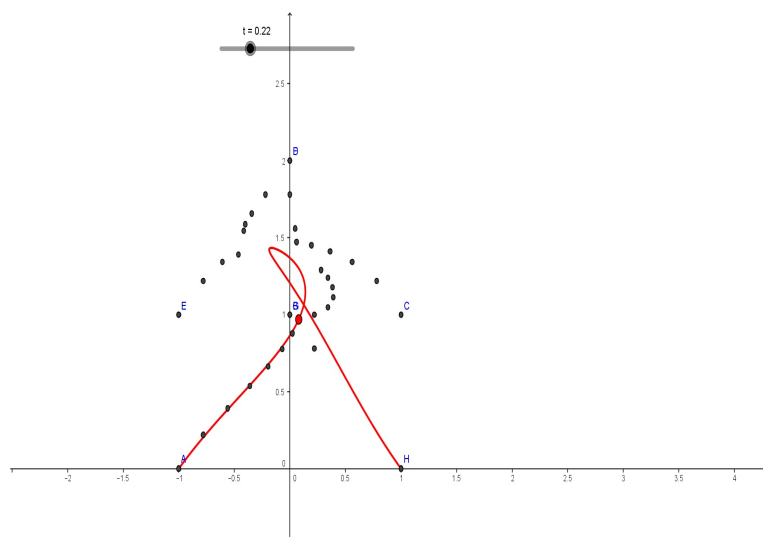


Figura 3.9: Implementação do algoritmo de Castejau.

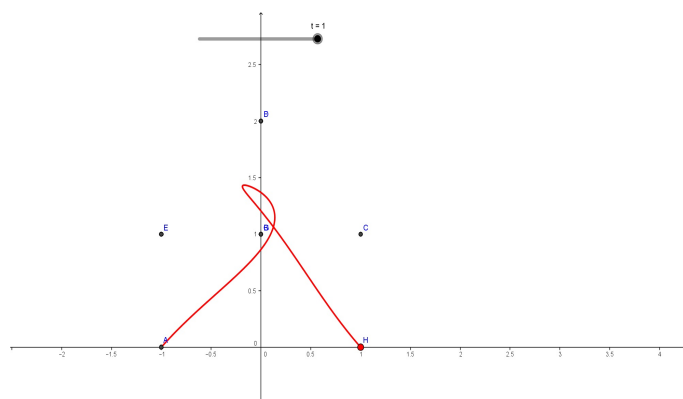
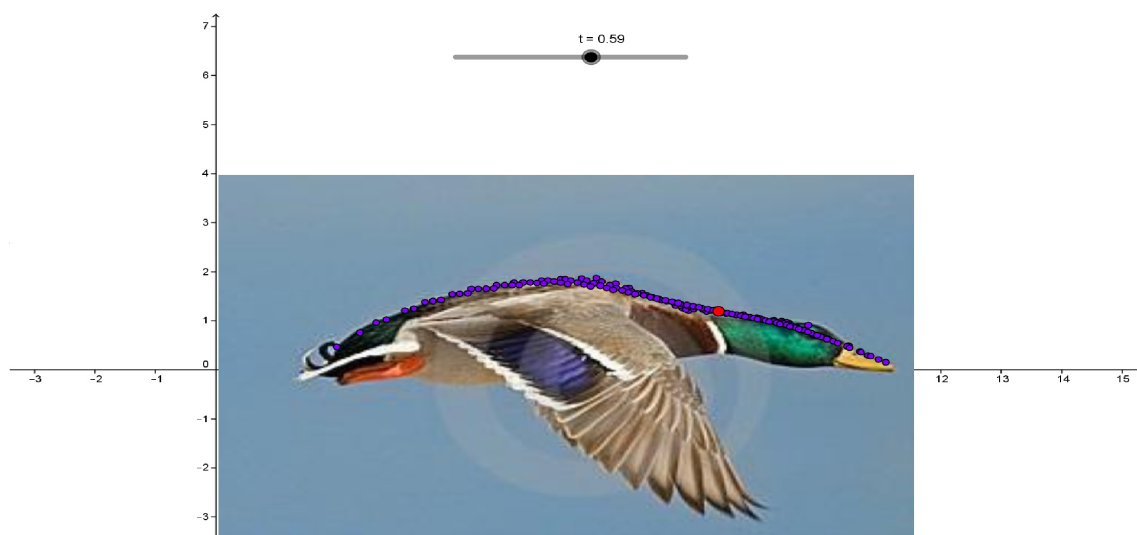


Figura 3.10: Curva gerada pelos pontos $(-1,0)$, $(0,1)$, $(1,1)$, $(0,2)$, $(0,1)$, $(-1,1)$, $(0,2)$, $(0,1)$ e $(1,0)$.



Figura 3.11: Escolha dos pontos sobre a imagem.

Figura 3.12: Variando o parâmetro t observamos os pontos intermediários gerados.

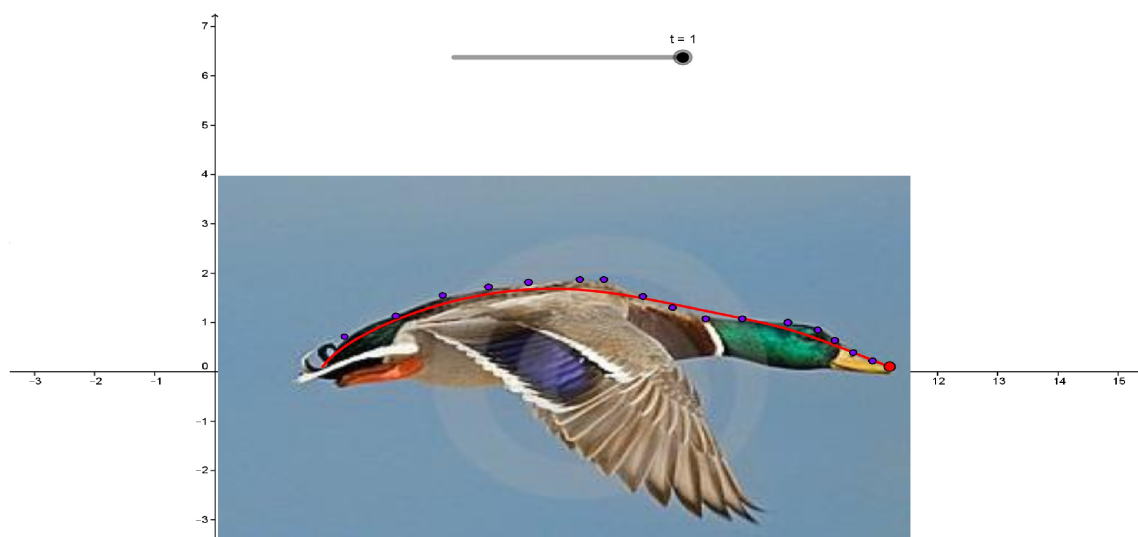


Figura 3.13: Curva gerada usando o lugar geométrico.

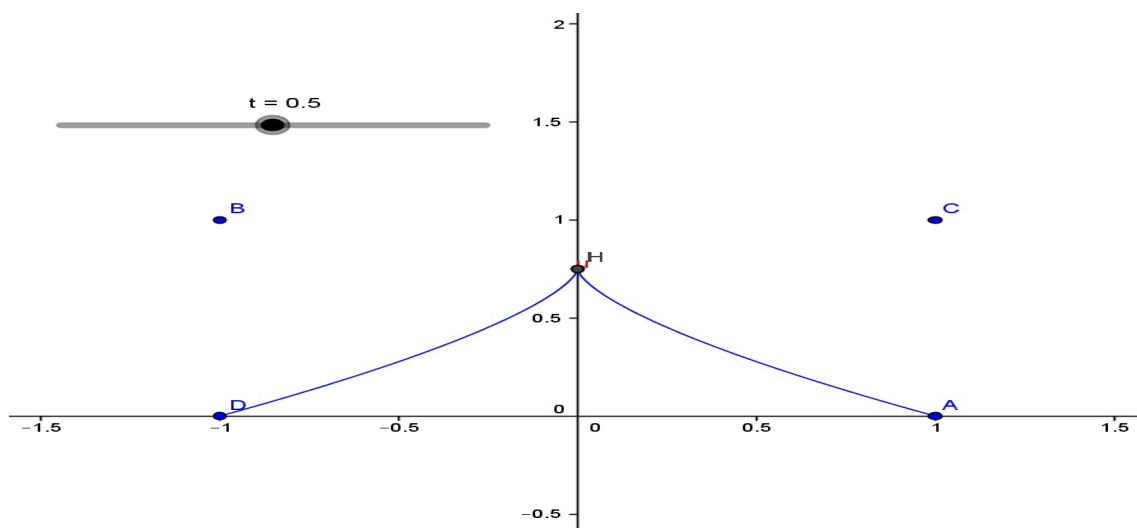


Figura 3.14: Curva gerada para os pontos $(1,0)$, $(-1,1)$, $(1,1)$ e $(-1,0)$ em $t = 0.5$ a derivada não existe para estes valores.

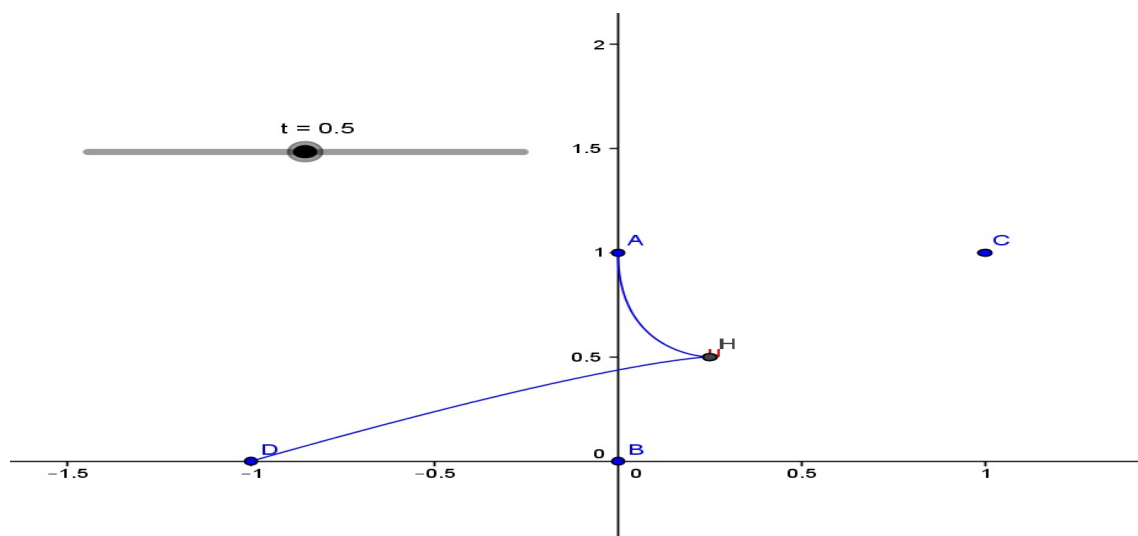


Figura 3.15: Curva gerada para os pontos $(0,1)$, $(0,0)$, $(1,1)$ e $(-1,0)$ em $t = 0.5$ a derivada não existe para estes valores.

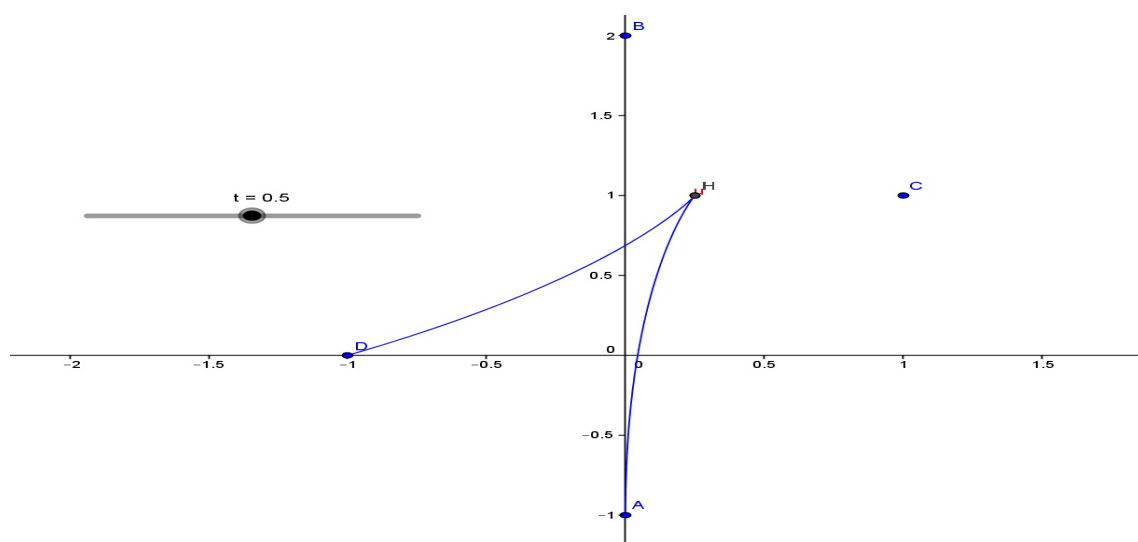


Figura 3.16: Curva gerada para os pontos $(0,-1)$, $(0,2)$, $(1,1)$ e $(-1,0)$ em $t = 0.5$ a derivada não existe para estes valores.

Capítulo 4

Resultados e Considerações Finais

Neste Capítulo, fizemos algumas observações sobre os exemplos abordados no capítulo 3, e por fim as considerações finais sobre o tema estudado.

4.1 Resultados

A implementação usando o algoritmo De Casteljaeu facilita bastante a construção da curva de Bézier, ao contrário do processo com segmentos de reta, a implementação usando o software Geogebra ocorre com muita eficiência usando a aba de planilha. O uso do comando lugar geométrico desenha a curva de Bézier muito melhor que o comando rastro, além de permitir uma reconfiguração instantânea da curva ao mudar os pontos de controle, figuras 3.1, 3.2 e 3.3.

Na curva de Bézier com os pontos de controle dados segundo três coordenadas no espaço, a implementação via algoritmo De Casteljaeu é essencial para sua construção, pois o software demonstra uma certa dificuldade na manipulação de segmentos de reta no espaço, algo que é aceitável dado o fato da versão em três dimensões ser bastante recente. A implementação realizada na figura 3.4 devido a escolha de pontos simétricos tornou a curva bastante interessante transpondo lateralmente o eixo z .

Nas espirais usamos como pontos de controle polígonos inscritos, nas figuras 3.5 e 3.8 alguns pontos dos vértices foram omitidos dos pontos de controle para que não houvesse uma auto-intersecção da curva. Estas duas situações espirais poderiam simular um escoamento rotacional de uma partícula.

O exemplo das figuras 3.9 e 3.10 apesar de serem simétricos os três pontos se repetem no polígono de controle gerando uma auto-intersecção na curva, do contrário ocorreria uma simetria na curva em relação ao eixo y .

A implementação usada nas figuras 3.11, 3.12 e 3.13 é um exemplo bastante oportuno que pode ser utilizado em situações diversas onde se deseja determinar a silhueta de uma imagem através dos pontos de contorno. A variação do parâmetro

t é capaz de simular o fluxo aerodinâmico ao longo do dorso do pato. A situação apresentada pode servir de motivação para modelar fluxos líquidos ou gases.

O problema sugerido em Farin [2] representado nas figuras 3.14, 3.15 e 3.16, demonstra como a simetria dos pontos gera uma curva cuja derivada para $t = 0.5$ não existe, ao variar a posição dos pontos A e B, gerando posições simétricas entre os quatro pontos notamos esta propriedade.

4.2 Considerações finais

O estudo das curvas de Bézier é um exemplo de aplicação matemática que leva à uma profunda discussão em temas diversos. Abordamos ao longo do texto funções paramétricas, polinômios, Binômio de Newton, triângulo de Pascal, proporções, vetores, derivadas e integrais. Alguns conceitos bastante úteis em computação gráfica foram introduzidos, o que proporciona um combustível na motivação pelo tema. As curvas de Bézier são parte introdutória para curvas mais eficientes como os B-splines e as superfícies de Bézier, que poderiam ser tratadas em situações posteriores.

A contribuição deixada por esta dissertação pode ser vista com um aditivo motivador da criatividade humana no processo de ensino-aprendizagem. As transformações que ocorrem na educação, além da necessidade de formar alunos que sejam capazes de investigar situações e tomar decisões ou tirar suas próprias conclusões sobre os resultados, estimulam o professor a desenvolver atividades como as descritas neste trabalho.

Apêndice A

Construindo curvas de Bézier usando o Geogebra e com o $\text{\LaTeX}$

A.1 Introdução

Neste apêndice mostramos algumas forma de construção da curva de Bézier utilizando o Geogebra de maneira detalhada e o $\text{\LaTeX}$. O fato de ser implementado em variados programas mostra a importância dessas curvas de Bézier e que elas podem ser utilizadas em várias aplicações. Vários pacotes e implementações do $\text{\LaTeX}$ foram encontradas em uma breve pesquisa. A seguir vamos mostrar alguns neste apêndice.

A.2 Construção no Geogebra

1ºPasso: Clique na opção Ponto mostrado pela seta vermelha na aba da figura A.1.

2ºPasso: Escolha quatro pontos no ambiente de construção do Geogebra, como mostra na figura A.2.

3ºPasso: Selecione a opção segmento nas ferramentas do Geogebra, como mostra a figura A.3.

4ºPasso: Clique nos pontos A e B para gerar o segmento de reta a , como mostra na figura A.4.

5ºPasso: Construa os outros dois segmentos b e c , como mostra a figura A.5.

6ºPasso: Selecione a opção controle deslizante para inserir o parâmetro t na construção da curva, figura A.6 .

7ºPasso: Clique no ambiente de construção e aparecerá um menu para alterar

A.2. CONSTRUÇÃO NO GEOGEBRA

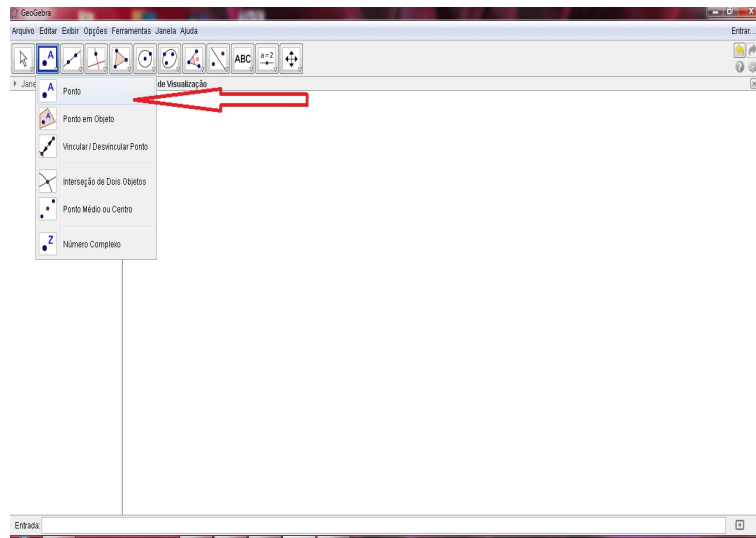


Figura A.1: Seleção da opção Ponto na barra de ferramenta.

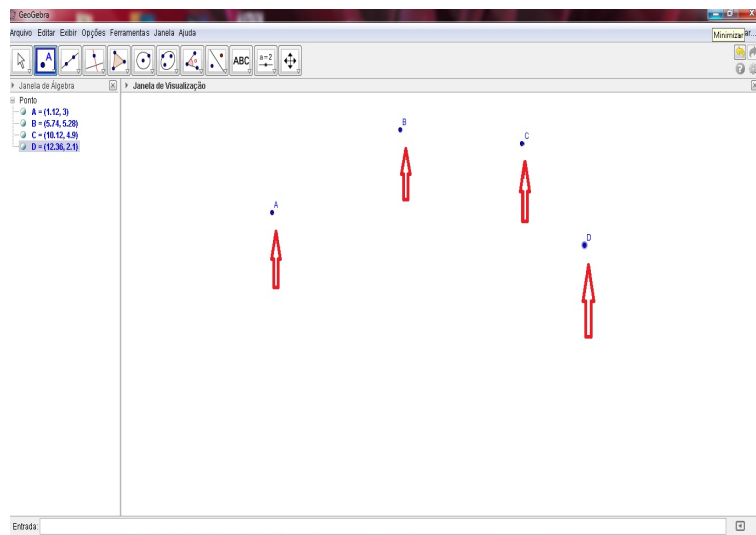


Figura A.2: Escolha dos Pontos.

o nome e os valores máximos e mínimos para t , como mostra a seta vermelha na figura A.7 e A.8.

8ºPasso: Clique na caixa de entrada para inserir a expressão, figura A.9 .

9ºPasso: Insira a expressão $t(B - A) + A$ na caixa de entrada, como mostra a figura A.9.

10ºPasso: O ponto E é gerado com o 9º passo como mostra a figura A.10.

11ºPasso: Usando a caixa de entrada gere os pontos F e G , como mostra a figura A.11.

A.2. CONSTRUÇÃO NO GEOGEBRA

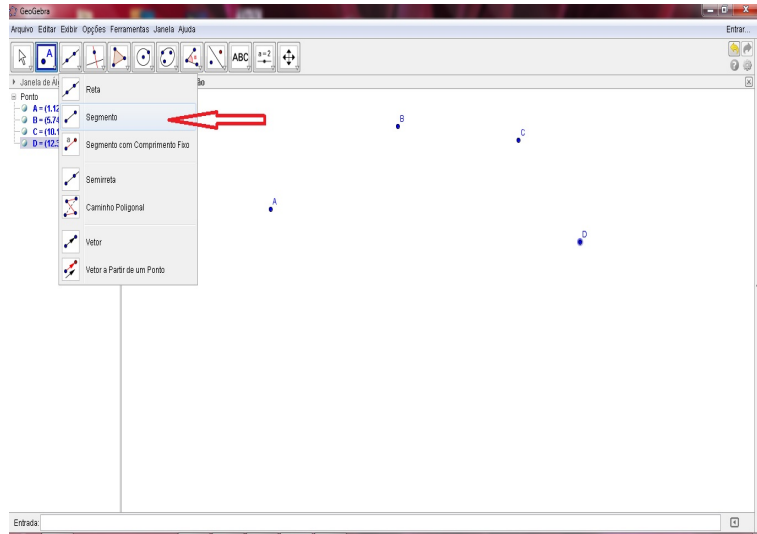


Figura A.3: Construção dos segmentos.

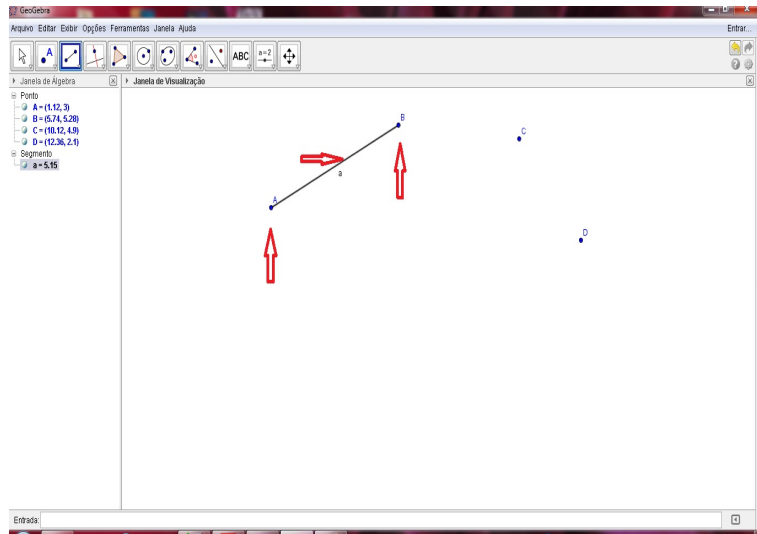


Figura A.4: Segmento gerado pelos pontos A e B.

12^oPasso: Construa os segmentos ligando os pontos E ao F e F ao G , como mostra a figuras A.12 e A.13.

13^oPasso: Use a caixa de entrada para gerar o ponto H com a expressão $t(F - E) + E$ e $t(G - F) + F$ para gerar o ponto I, como mostra a figura A.14 e A.15 .

14^o Passo: Construa um segmento entre os pontos H e I , figura A.16. O último ponto é gerado usando a caixa de entrada com a expressão $t(I - H) + H$, figura A.17 .

15^oPasso: Use o lugar geométrico na barra de ferramenta conforme mostra a

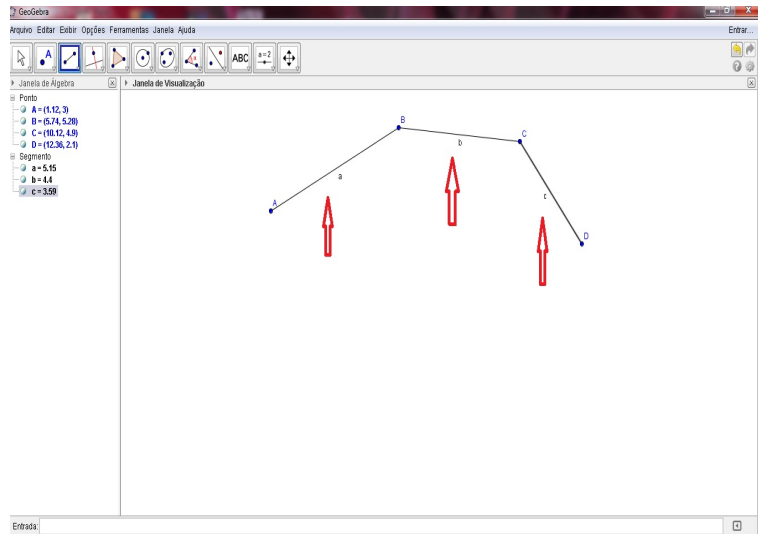


Figura A.5: Construção dos segmentos dos pontos B, C e D.

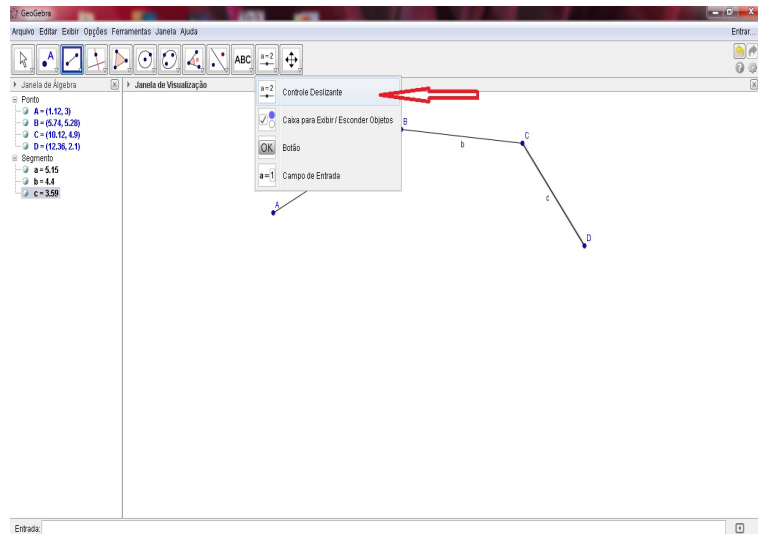


Figura A.6: Implementação do controle deslizante.

seta vermelha, figura A.18 . Clique no ponto J e no controle deslizante para gerar a Curva de Bézier, figura A.19 .

16ºPasso: Mova o controle deslizante para verificar o movimento dos segmentos e do ponto J ao longo da Curva de Bézier, figura A.20 .

A.3. USANDO O PSTricks

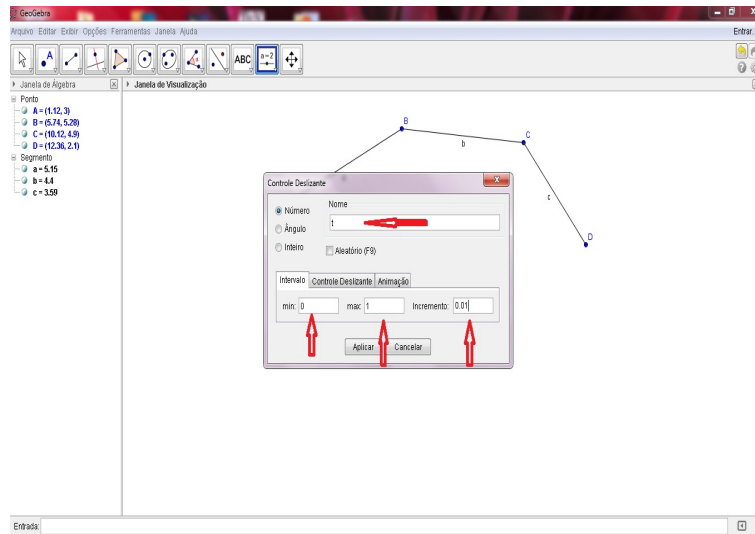


Figura A.7: Definindo o intervalo do parâmetro t .

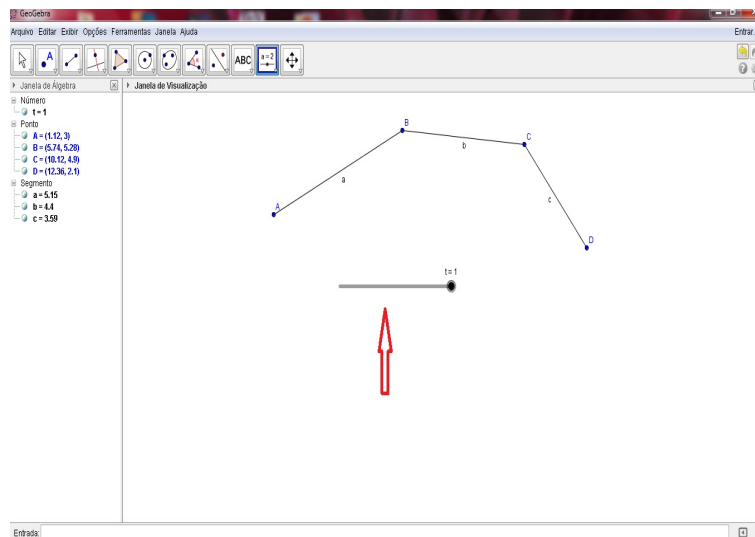


Figura A.8: Controle deslizante t .

A.3 Usando o PSTricks

Uma forma de gerar curvas de Bézier com o $\text{\LaTeX}$ é utilizando o comando: `\psbezier*[par]{arrows}(x0,y0)(x1,y1)(x2,y2)(x3,y3)`. O `\psbezier` desenha uma curva com os quatro pontos de controle. Assim como foi visto na teoria, a função interpola os primeiro e o último ponto, com os pontos intermediários exercendo influência sobre a trajetória [12]. O exemplo:

`\psbezier[linewidth=2pt,showpoints=true]{->}(0,0)(1,4)(2,1)(4,3.5),`

A.3. USANDO O PSTRICKS

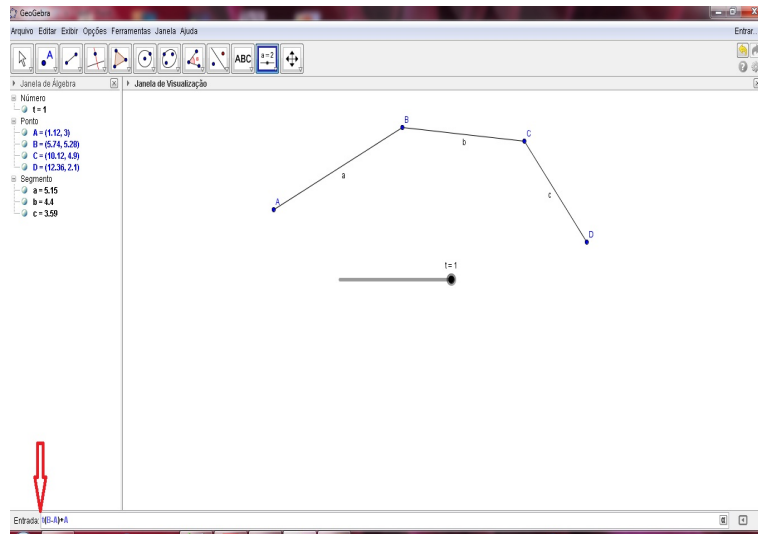


Figura A.9: Construção do ponto sobre o segmento usando a caixa de entrada.

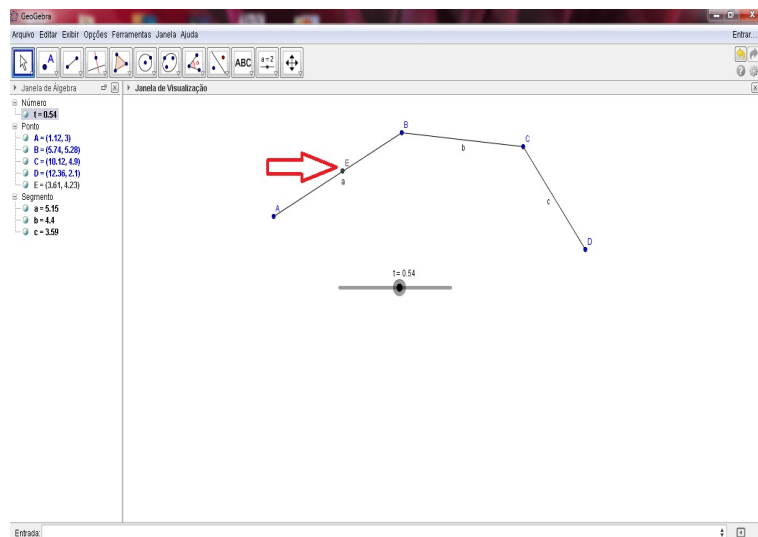


Figura A.10: Ponto sobre o segmento $\overline{AB}$.

mostra a curva da figura A.21.

O comando `showpoints=true` coloca todos os pontos de controle, e os conecta por linhas tracejadas, isto é útil quando se ajusta a curva de bezier.

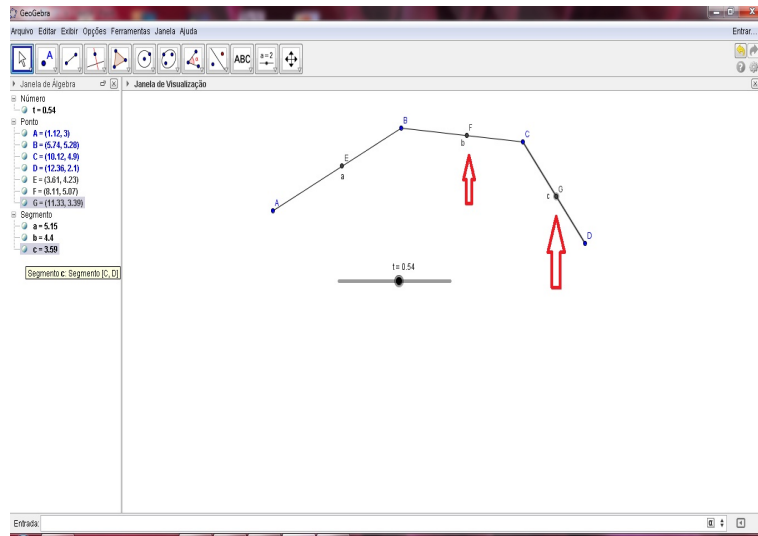


Figura A.11: Pontos F e G sobre os segmentos $\overline{BC}$ e $\overline{CD}$.

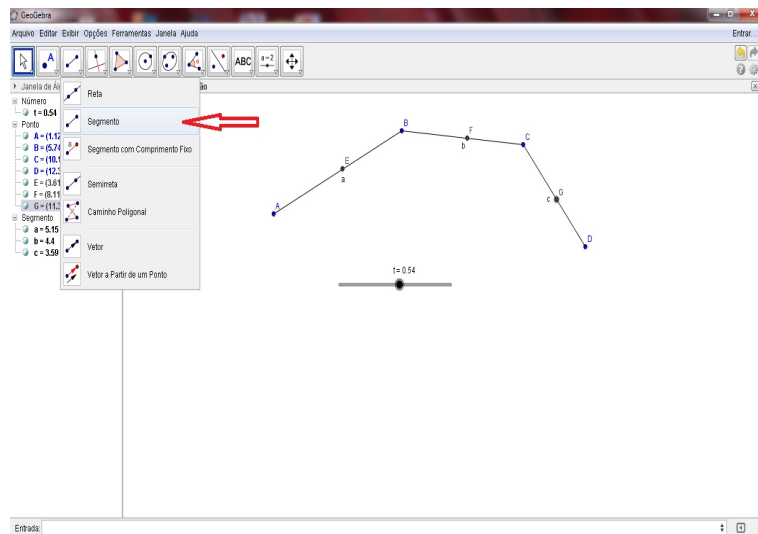


Figura A.12: Construção do segmento $\overline{EF}$.

A.4 Usando o ambiente *figure*

Existe a possibilidade de gerar curvas quadráticas de Bézier dentro do ambiente *picture*, mas é necessário fazer alguns cálculos [4]. O comando

`\multiput(x1,y1)(x2,y2){n}{objecto}`

Tem 4 argumentos: o ponto inicial, o vetor de translação de um objeto para o próximo, número de objetos, e o objeto a ser desenhado. O comando:

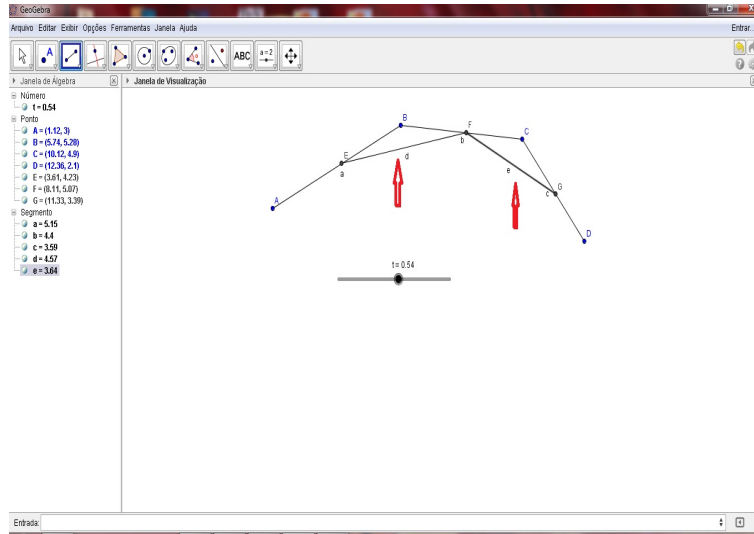


Figura A.13: Segmentos $\overline{EF}$ e $\overline{FG}$.

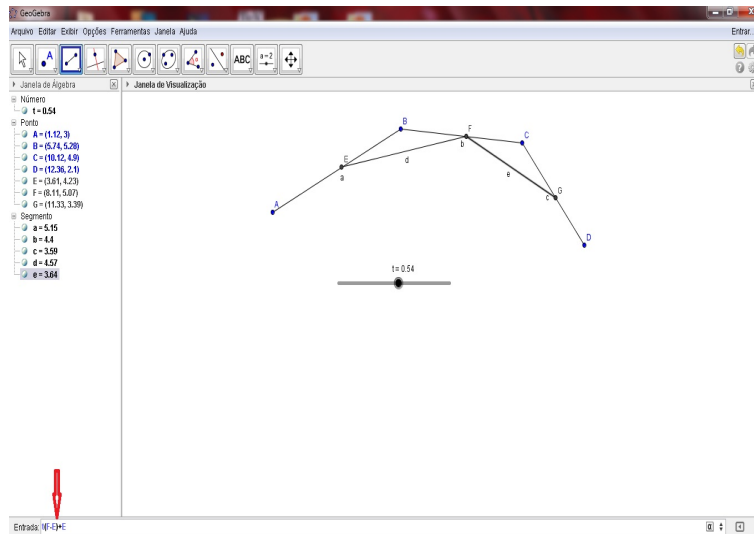


Figura A.14: Construção dos pontos sobre os segmentos $\overline{EF}$ e $\overline{FG}$.

`\linethickness`

aplica-se a segmentos horizontais e verticais mas nunca a segmentos de reta oblíquos ou circulares. No entanto, é aplicado a curvas quadráticas de Bézier. Os comandos `\put(x, y){\oval(w, h)}` ou `\put(x, y){\oval(w, h)[posição]}` produzem uma oval centrada em (x, y) com largura w e altura h . O argumento opcional posição que pode ser um entre b, t, l, r, refere-se a top (topo), bottom (fundo), left (esquerda), right (direita), e podem ser combinados como ilustra o exemplo.

A.4. USANDO O AMBIENTE FIGURE

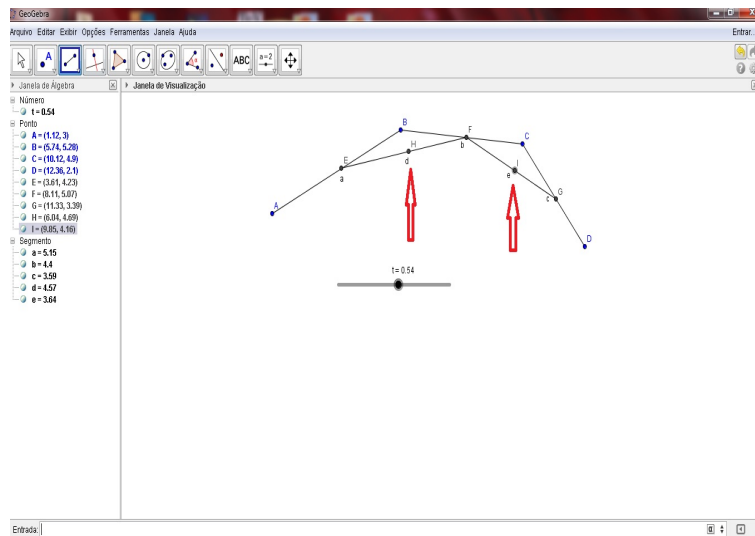


Figura A.15: Pontos H e G.

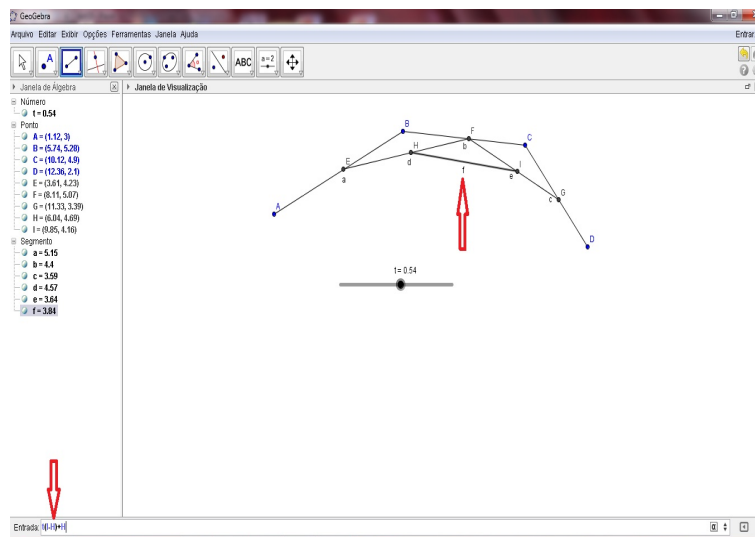


Figura A.16: Gerando o ponto J usando a caixa de entrada.

```
\setlength{\unitlength}{0.75cm}
\begin{picture}(6,4)
\linethickness{0.075mm}
\multiput(0,0)(1,0){7}%
{\line(0,1){4}}
\multiput(0,0)(0,1){5}%
{\line(1,0){6}}
\thicklines
```

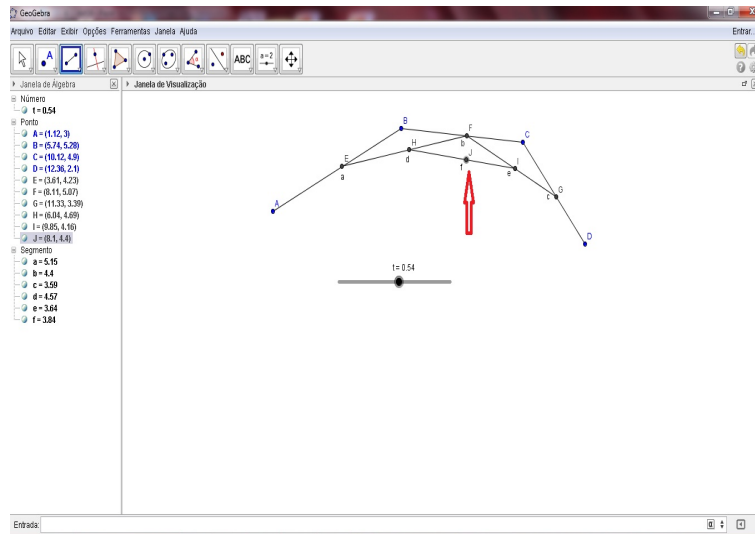


Figura A.17: Ponto J.

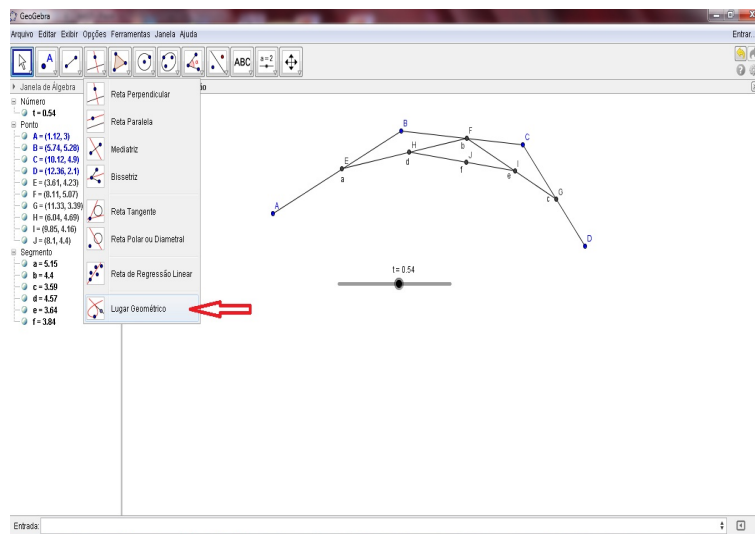


Figura A.18: Seleção do lugar geométrico na barra de ferramenta.

```
\put(2,3){\oval(3,1.8)}
\thinlines
\put(3,2){\oval(3,1.8)}
\thicklines
\put(2,1){\oval(3,1.8)[t1]}
\put(4,1){\oval(3,1.8)[b]}
\put(4,3){\oval(3,1.8)[r]}
\put(3,1.5){\oval(1.8,0.4)}
```

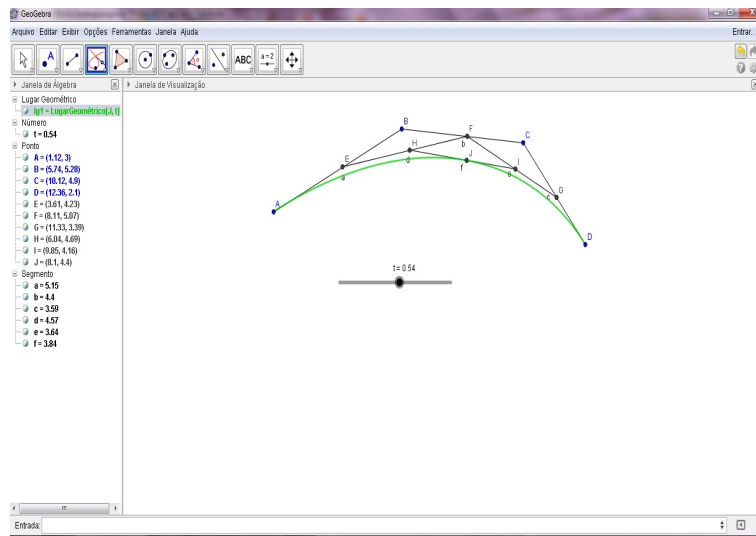
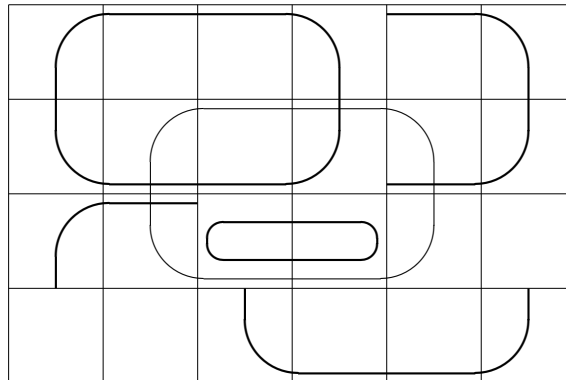


Figura A.19: Gerando a curva de Bézier em relação ao ponto J e o parâmetro t .

`\end{picture}`



A espessura das linhas pode ser controlada com dois tipos de comandos: `\linethickness{comprimento}` por um lado, `\thinlines` e `\thicklines` por outro. Enquanto que `\linethickness{comprimento}` só se aplica a linhas verticais e horizontais (e curvas quadráticas de Bézier), `\thinlines` e `\thicklines` aplicam-se também a segmentos oblíquos assim como a círculos e ovais.

No exemplo seguinte a figura mostra o efeito do comando `\linethickness` em linhas horizontais ou verticais, e o efeito dos comandos `\thinlines` e `\thicklines` em segmentos de reta oblíquos. Também mostra que ambos os tipos de comando afetam as curvas quadráticas de Bézier, cada comando substituindo todos os anteriores.

`\setlength{\unitlength}{1.0cm}`



Figura A.20: Variação do ponto J usando o controle deslizante.

```
\begin{picture}(6,4)
\centering
\linethickness{0.075mm}
\multiput(0,0)(1,0){7}
{\line(0,1){4}}
\multiput(0,0)(0,1){5}
{\line(1,0){6}}
\thicklines
\put(0.5,0.5){\line(1,5){0.5}}
\put(1,3){\line(4,1){2}}
\qbezier(0.5,0.5)(1,3)(3,3.5)
\thinlines
\put(2.5,2){\line(2,-1){3}}
\put(5.5,0.5){\line(-1,5){0.5}}
```

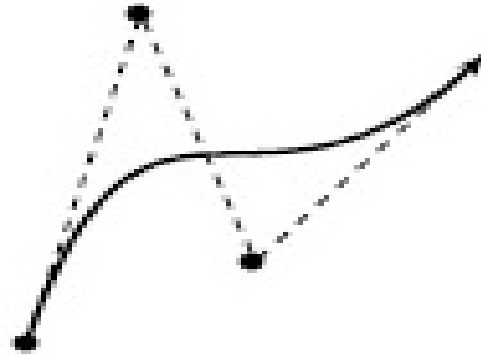
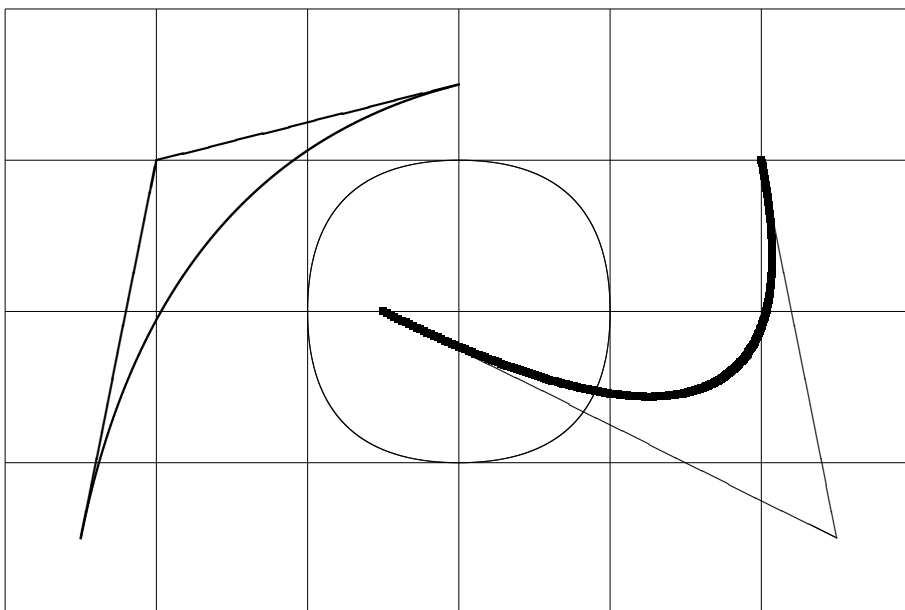


Figura A.21: Curva gerada com o comando `psbezier`

```
\linethickness{1mm}
\qbezier(2.5,2)(5.5,0.5)(5,3)
\thinlines
\qbezier(4,2)(4,3)(3,3)
\qbezier(3,3)(2,3)(2,2)
\qbezier(2,2)(2,1)(3,1)
\qbezier(3,1)(4,1)(4,2)
\end{picture}.
```

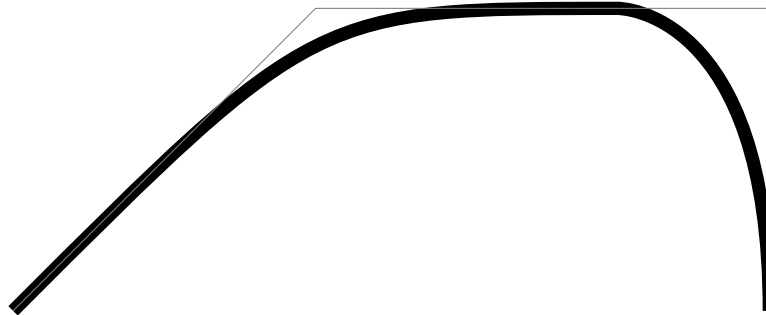


A.5 Usando o PGF

Neste procedimento temos outra forma de desenhar as curvas de Bézier.

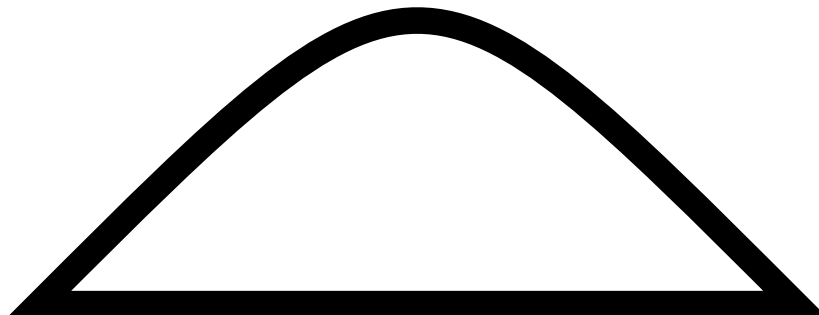
`\path . . . ..controls <c> and <d>..`

Este comando gera as curvas de Bézier quadráticas, logo os pontos de controle são dados três a três, no caso c, d e y são pontos [8]. outra opção é terminar com o atributo `cycle`. Vemos dois exemplos de utilização do comando.



```
\begin{center}
\begin{tikzpicture}{!c}
\draw[line_width=5pt](0,0)..controls(4,4)..(8,4)
..controls(8,4)and(10,4)..(10,0);
\draw[color=gray](0,0)--(4,4)--(8,4)--(8,4)..--(10,4)--(10,0);
\end{tikzpicture}
\end{center}
```

```
\begin{center}
\begin{tikzpicture}
\draw[line_width=10pt](0,0)..--(10,0)..controls(5,5)..cycle;
\end{tikzpicture}
\end{center}
```



A.6 Usando o TEXdraw

Curvas de Bezier desenhadas com o pacote TEXdraw usa quatro coordenadas de referência, duas como os pontos finais e outras duas para controlar a forma da curva.

`\clvec (x1 y1)(x2 y2)(x3 y3)`

A curva é desenhada a partir da posição atual até a coordenada $(y3\ x3)$. As coordenadas $(x1\ y1)$ e $(Y2\ x2)$ servem como pontos de controle para a curva [3]. Apenas a última cota dada é utilizado para atualizar o tamanho do desenho. Note-se que apenas 3 pares de coordenadas são especificados. O outro ponto é a posição atual antes do comando `\clvec` ser executado. Apenas a última coordenar especificada no comando `\clvec` é usado para determinar a extensão do desenho. Embora a curva de Bezier passe pelos pontos inicial e final, em geral, a curva não vai atingir os pontos de controle intermediários. A curva é sempre encapsulada pelo quadrilátero convexo definido pelas extremidade e os dois pontos de controle intermediários. Vejamos um exemplo dado pelos comandos:

```
\btexdraw
\move(5,5)
\clvec(5,6)(6,5)(6,6)
\etexdraw
```

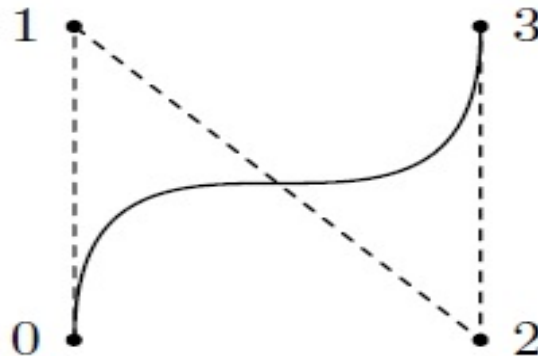


Figura A.22: Curva gerada com o comando `clvec` do TEXdraw

A.7 Usando o pacote bez123

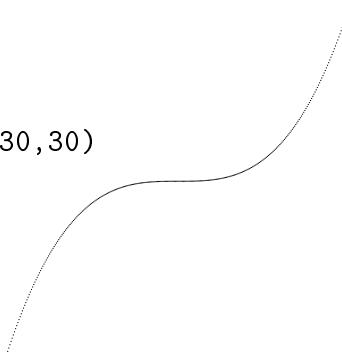
Outra maneira de desenhar as curvas de Bézier é utilizando o pacote do $\text{\LaTeX}$ `bez123` [11]. O comando `\cbezier[<N>](p0)(p1)(p2)(p3)` desenha uma curva

cúbica de Bézier usando N pontos iniciando no ponto p_0 e finalizando no ponto p_3 , com os pontos p_1 e p_2 como intermediários. Realizando as rotinas a seguir produziremos as seguintes curvas:

```
\begin{center}
\setlength{\unitlength}{0.2cm}
\begin{picture}(6,4)

\cbezier[300](0,0)(10,30)(20,0)(30,30)

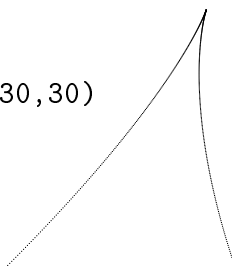
\end{picture}
\end{center}
```



```
\begin{center}
\setlength{\unitlength}{0.2cm}
\begin{picture}(6,4)

\cbezier[300](0,0)(10,30)(20,0)(30,30)

\end{picture}
\end{center}
```



Neste exemplos usamos 300 pontos para definir as curvas usando o comando `\cbezier`.

Apêndice B

Teorema da Aproximação de Weierstrass

Teorema B.1 (Weierstrass) *Seja $f : [a, b] \rightarrow \mathbb{R}$ uma função contínua. Para todo $\epsilon > 0$ existe um polinômio $p(t)$ tal que $|f(t) - p(t)| < \epsilon$, para todo $t \in [a, b]$.*

Demonstração. Seja $t = (1-x)a + xb$. Então $x = \frac{1}{b-a}(t-a)$ e $t \in [a, b]$ se, e somente se, $x \in [0, 1]$. Seja $\tilde{f}(x) = f((1-x)a + xb)$. Seja

$$\tilde{p}(x) = \sum_{k=0}^n \tilde{f}\left(\frac{k}{n}\right) \binom{n}{k} x^k (1-x)^{n-k}$$

e

$$p(t) = \tilde{p}\left(\frac{1}{b-a}(t-a)\right)$$

Este é o polinômio de Bernstein. Vamos usar o fato de que

$$\sum_{k \in A} \binom{n}{k} x^k (1-x)^{n-k} \leq \sum_{k=0}^n \binom{n}{k} x^k (1-x)^{n-k} = 1, \quad (\text{B.1})$$

para qualquer $A \subseteq \{0, 1, 2, \dots, n\}$. Como f é contínua existe $\delta > 0$ tal que

$$|x - y| < \delta \implies |\tilde{f}(x) - \tilde{f}(y)| < \frac{\epsilon}{2}. \quad (\text{B.2})$$

Sejam $b_1 = x - \delta$ e $b_2 = x + \delta$. Seja $M = \max_{x \in [0, 1]} |\tilde{f}(x)| = \max_{t \in [a, b]} |f(t)|$. Seja n tal que $4Me^{-2\delta^2 n} < \frac{\epsilon}{2}$. Vamos usar o seguinte fato que será demonstrado a seguir:

$$b_2 \leq \frac{k}{n} \leq 1 \text{ ou } 0 \leq \frac{k}{n} \leq b_1$$

$$x^{\frac{k}{n}} (1-x)^{1-\frac{k}{n}} \leq e^{-2(x-b)^2} b^{\frac{k}{n}} (1-b)^{1-\frac{k}{n}} \quad (\text{B.3})$$

Então por A.1, A.2 e A.3 temos que

$$\begin{aligned}
 |f(x) - \tilde{p}(x)| &= \left| \sum_{k=0}^n \tilde{f}(x) \binom{n}{k} x^k (1-x)^{n-k} - \sum_{k=0}^n \tilde{f}\left(\frac{k}{n}\right) \binom{n}{k} x^k (1-x)^{n-k} \right| \\
 &\leq \sum_{k=0}^n \left| \tilde{f}\left(\frac{k}{n}\right) - \tilde{f}(x) \right| \binom{n}{k} x^k (1-x)^{n-k} \leq \\
 &\leq \frac{\epsilon}{2} + 2M \sum_{\left|\frac{k}{n} - x\right| \geq \delta} \left| \tilde{f}\left(\frac{k}{n}\right) - \tilde{f}(x) \right| \binom{n}{k} x^k (1-x)^{n-k} \leq \\
 &\leq \frac{\epsilon}{2} + 2M \sum_{\frac{k}{n} \geq b_2} \binom{n}{k} x^k (1-x)^{n-k} + 2M \sum_{\frac{k}{n} \leq b_1} \binom{n}{k} x^k (1-x)^{n-k} \\
 &\leq \frac{\epsilon}{2} + 4M e^{-2\delta^2 n} \leq \epsilon.
 \end{aligned}$$

Lema B.1 Se $0 \leq x \leq b \leq \frac{k}{n} \leq 1$ ou $0 \leq \frac{k}{n} \leq b < x \leq 1$, então

$$x^{\frac{k}{n}} (1-x)^{1-\frac{k}{n}} \leq e^{-2(x-b)^2} b^{\frac{k}{n}} (1-b)^{1-\frac{k}{n}}$$

Demonstração. Precisamos mostrar que

$$\frac{x^{\frac{k}{n}} (1-x)^{1-\frac{k}{n}}}{b^{\frac{k}{n}} (1-b)^{1-\frac{k}{n}}} \leq e^{-2(x-b)^2},$$

ou aplicando-se o logaritmo nesta desigualdade, que

$$H(x) = \ln \frac{x^{\frac{k}{n}} (1-x)^{1-\frac{k}{n}}}{b^{\frac{k}{n}} (1-b)^{1-\frac{k}{n}}} + 2(x-b)^2 \leq 0$$

Temos que $H(b) = 0$.

(a) Se $0 \leq x \leq b \leq \frac{k}{n} \leq 1$, vamos mostrar que $H'(x) \geq 0$. Como, para $0 < x < 1$, $x(x-1) \leq \frac{1}{4}$, então

$$H'(x) = \frac{\frac{k}{n} - x}{x(1-x)} + 4(x-b) \geq 4\left(\frac{k}{n} - x\right) + 4(x-b) = 4\left(\frac{k}{n} - b\right) \geq 0.$$

(b) Se $0 \leq \frac{k}{n} \leq b < x \leq 1$, vamos mostrar que $H'(x) \leq 0$. Como, para $0 < x < 1$, $4 \leq \frac{1}{x(1-x)}$, então

$$H'(x) = \frac{\frac{k}{n} - x}{x(1-x)} + 4(x-b) \leq \frac{\frac{k}{n} - x}{x(1-x)} + \frac{x-b}{x(1-x)} = \frac{\frac{k}{n} - b}{x(1-x)} \leq 0$$

Referências Bibliográficas

- [1] Campos, F. F. *Algoritmos Numéricos*. 2^aed. Rio de Janeiro. LTC (2007).
- [2] Farin, G, *Curves and Surfaces for CAGD:A Practical Guide*. 5^a edition, Arizona, Morgan Kaufmann Publishers (2002).
- [3] Kabal, P. *TEXdraw: PostScript Drawings from TEX*. 2^aed. University Montreal, Quebec (1995).
- [4] Oetiker,T.,Partl, H., Hyna, I., Schlegl, E. *Uma não tão pequena introdução ao L^AT_EX2.09*. Zentraler Informatikdienst der Universität für Bodenkultur Wien (2011).
- [5] Paluszny, M. , Prautzs, H., Boehm, W. *Método de Bézier y B-splines*.1^aed. Universitätsverlag Karlsruhe (2005).
- [6] Santos, R. J., *Teorema da Aproximação de Weierstrass*. acesso: <http://www.mat.ufmg.br/~regi>, em 23/03/2015.
- [7] Scherer, C., *Métodos Computacionais da Física*. 1^aed. São Paulo. Editora Livraria da Física (2005).
- [8] Tantau, T. *The TikZ and PGF Packages Manual for version 3.0.0*. Institut für Theoretische Informatik Universität zu Lübeck (2013).
- [9] Velho, L., Gomes, J. *Fundamentos da Computação gráfica*. 1^a edição , Rio de Janeiro, SBM (2008).
- [10] Vince, J. *Mathematics For Graphics*. 2^aed. Springer-Verlag (2006).
- [11] Wilson, P., Press, H. *The bez123 and multiply packages*.(2002).
- [12] Zandt,T. V. *PSTricks: PostScript macros for Generic TeX*. Fontainebleau(Paris)(2003).