



UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E COMPUTAÇÃO CIENTÍFICA
DEPARTAMENTO DE MATEMÁTICA APLICADA



Theo Trainotti Freire

Ajuste de Curvas Bézier

Campinas
25/06/2024

Theo Trainotti Freire

Ajuste de Curvas Bézier*

Monografia apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos para obtenção de créditos na disciplina Projeto Supervisionado, sob a orientação do(a) Prof. Ricardo Biloti.

*Este trabalho foi financiado pela PIBIC, projeto 2023/2024.

Resumo

Neste trabalho foi abordado o ajuste de curvas do tipo Bézier em diferentes contextos. Tais curvas, simples de serem calculadas, no entanto muito plásticas, possuem uma gama de aplicações, em especial no design gráfico computacional.

Alguns tópicos foram abordados inicialmente para posteriormente serem aplicados: a definição e construção das próprias curvas Bézier; o problema de quadrados mínimos para solução de problemas de ajuste e sua interpretação geométrica; e o método de Gauss-Newton para a obtenção de zeros de funções não lineares de múltiplas variáveis.

Os alvos do trabalho foram dois métodos para tratar o problema de ajuste de curvas Bézier em trechos de curvas paramétricas contínuas, e em um conjunto de pontos ordenado. Em ambos os casos, o objetivo foi ajustar a curva Bézier para que o resultado final fosse graficamente satisfatório, e com isso facilitar a aplicação das curvas em uma gama de problemas gráficos. Por fim, os métodos foram aplicados computacionalmente.

Abstract

In this project, topics related to fitting a Bézier curve in different contexts were addressed. Such curves, simple to compute yet very versatile, have a range of applications, especially in computational graphic design.

Several topics were initially discussed to facilitate their subsequent application: the definition and construction of Bézier curves themselves; the least squares problem for the solution of fitting problems and its geometrical interpretation; and the Gauss-Newton method for finding the zeros of a nonlinear function with multiple variables.

The focus of the study was on two methods for fitting a Bézier curve to both a segment of a continuous parametric curve and an ordered set of points. In both cases, the objective was to fit a Bézier curve such that the final result was graphically satisfactory, facilitating the application of such curves to a variety of graphics problems. Finally, the methods were computationally implemented.

Conteúdo

| | | |
|----------|---|-----------|
| 1 | Introdução | 6 |
| 2 | Fundamentos | 6 |
| 2.1 | Curva Bézier | 7 |
| 2.2 | Problema dos quadrados mínimos e o projetor ortogonal | 9 |
| 2.3 | Os métodos de Newton e de Gauss-Newton | 12 |
| 3 | Ajuste em uma curva contínua | 14 |
| 3.1 | Primeira condição | 15 |
| 3.2 | Segunda condição | 16 |
| 3.3 | Implementação, Testes e Resultados | 19 |
| 4 | Ajuste em uma amostra de pontos | 22 |
| 4.1 | Construção do Método | 23 |
| 4.2 | Aplicando Gauss-Newton | 25 |
| 4.3 | Implementação, Testes e Resultados | 28 |

1 Introdução

Neste trabalho, foram estudadas algumas aplicações para curvas do tipo Bézier, conhecidas por serem extremamente plásticas e simples de serem construídas computacionalmente, já que são formadas por polinômios. Tais aplicações consistem na aproximação de um trecho uma curva paramétrica qualquer por uma curva Bézier, e no ajuste de uma curva deste tipo em a conjunto de pontos ordenados. Cada uma destas aplicações exigem bases conceituais de diferentes áreas da matemática. Enquanto a primeira baseia-se mais em ferramentas do cálculo diferencial, a segunda se beneficia mais de métodos algébricos.

O desenvolvimento deste trabalho basou-se fundamentalmente em dois principais artigos, Penner [2019] e Borges and Pastva [2002], cada um sobre uma das aplicações estudadas. Além dos métodos alvos do estudo, outros conceitos fundamentais para o desenvolvimento deles foram pesquisados.

Na seção 2 abordei esses conceitos, são eles a origem histórica e matemática das curvas Bézier; o problema de quadrados mínimos e sua relação com projetores ortogonais; o método de Gauss-Newton para resolver problemas de otimização não lineares. Na seção 3 foi trabalhado o conteúdo do primeiro artigo citado acima, que trata do problema de ajuste de uma Bézier cubica em um trecho de uma curva paramétrica. Nesta etapa, o desenvolvimento do método utilizou-se principalmente das propriedades diferenciais das curvas, em especial suas derivadas primeiras e segundas, e com elas a curvatura das curvas. Na seção 4 foi desenvolvido o método do segundo artigo acima citado sobre ajuste de uma Bézier de grau qualquer em um conjunto de pontos ordenados. Neste tópico, o desenvolvimento do método se baseou principalmente no uso de propriedades obtidas a partir da solução de sistemas lineares e de otimização de funções não lineares de múltiplas variáveis.

2 Fundamentos

O desenvolvimento dos métodos utilizados neste relatório exige o entendimento prévio de alguns conceitos matemáticos, e, por vezes, algum ponto de vista específico sobre determinados assuntos. Assim, utilize esta seção como guia para aprender ou rever estes tópicos, que serão abordados ao longo do texto.

Primeiramente investigaremos o que são as curvas Bézier, desde o contexto em que foram criadas, suas utilidades e um método para obter as expressões delas em diferentes graus. Então abordaremos o problema de quadrados mínimos e como sua interpretação geométrica leva a importantes ferramentas para o estudo de espaços vetoriais. Por fim, estudaremos o que é o método de Gauss-Newton para a otimização de funções não lineares com múltiplas variáveis.

2.1 Curva Bézier

Funções e curvas paramétricas são objetos extremamente presentes em toda a matemática. Já no ensino médio aprendemos sobre funções polinomiais, exponenciais e trigonométricas e suas representações gráficas. Na graduação aprendemos a estudar e manipular curvas ainda mais complexas, cada uma com propriedades úteis nas mais diversas áreas do conhecimento. No entanto, nenhuma das funções e curvas mais comuns e mais estudadas nesses períodos possuem uma propriedade específica e extremamente útil, especialmente no mundo moderno e informatizado de hoje: plasticidade..

Era isso que Pierre Bézier (Fig 1) buscava em seu emprego na fabricante de carros Renault, na década de 1960. Na época, o processo de criação de um carro, desde a sua concepção até sua fabricação, sofria de um problema de comunicação. Os designers dos automóveis criavam modelos reais em cera dos carros, que posteriormente teriam que ser traduzidos para a linguagem matemática por engenheiros, para permitir análises de aerodinâmica e para a confecção de suas peças. Esse processo de conversão, no entanto, não era preciso. Os engenheiros tinham que fazer aproximações por vezes grosseiras para encontrar curvas matemáticas que representavam as formas criadas pelo time criativo.

Foi nesse contexto que Pierre buscou encontrar uma maneira de definir curvas e superfícies de forma que seu manuseio fosse simples e intuitivo o suficiente para que os carros fossem projetados desde sua concepção na linguagem matemática. Assim, surgiu a ideia de definir curvas por meio de pontos de controle, facilmente manipuláveis com o auxílio de computadores, e que alterassem a curva de forma intuitiva o suficiente para facilitar o trabalho dos designers. Estas curvas também deveriam ser simples de serem calculadas, devido ao baixo poder computacional dos computadores da época. Vamos entender melhor estas curvas.



Figura 1: Pierre Bézier, 1958 (Wikipédia).

Começando pelo caso mais simples, considere que temos dois pontos, P_0 e P_1 , e queremos encontrar uma curva que começa em um e termina no outro. Uma forma de fazer isso é definir

$$b_1(t) = P_0 + t(P_1 - P_0), \text{ para } t \in [0, 1] \text{ e } P_1, P_2 \in \mathbb{R}^n.$$

Inspirado nesta expressão, obtemos o que chamamos de interpolação linear, ou $lerp(A, B, t)$,

$$lerp(A, B, t) = A(1 - t) + Bt, t \in [0, 1] \text{ e } A, B \in \mathbb{R}^n.$$

Expandindo este conceito, uma forma de aumentar a complexidade e, conseqüentemente, a liberdade de formas que uma curva desse tipo pode tomar é aplicar esta função $lerp(A, B, t)$ recursivamente. Ou seja, tome os pontos P_0 , P_1 e P_2 , e com eles definimos os pontos $A(t) = lerp(P_0, P_1, t)$ e $B(t) = lerp(P_1, P_2, t)$, e por fim obteremos a curva definida por

$lerp(A(t), B(t), t)$, com a seguinte expressão:

$$b_2(t) = P_0(1-t)^2 + P_1t(1-t) + P_2t^2, t \in [0, 1] e P_0, P_1, P_2 \in \mathbb{R}^n.$$

Essa é dita uma Bézier quadrática (Fig 2). Repetindo este processo, adicionando mais um ponto de controle, obtemos a Bézier cúbica (Fig 3). Esta é a curva Bézier mais utilizada nas aplicações modernas, e tem a seguinte expressão:

$$b_3(t) = P_0(1-t)^3 + 3P_1t(1-t)^2 + 3P_2t^2(1-t) + P_3t^3, t \in [0, 1] e P_0, P_1, P_2, P_3 \in \mathbb{R}^n. \quad (1)$$

Usando esse processo recursivo, é possível criar curvas Bézier de qualquer grau, inteiro e positivo. Na seção 4 veremos que poderemos generalizar o método em questão para qualquer curva Bézier, por isso é útil termos em mãos uma expressão bem definida para encontrar uma curva de um grau qualquer n , (Lorentz [1986]):

$$b(t) = (x(t), y(t)) = \sum_{j=0}^n B_j^n(t) (x_j, y_j)$$

tal que

$$B_j^n(t) = \binom{n}{j} t^j (1-t)^{n-j} \quad (2)$$

e que x e y são os vetores em \mathbb{R}^n com as coordenadas dos pontos de controle da Bézier.

2.2 Problema dos quadrados mínimos e o projetor ortogonal

Começamos relembrando o que é o método dos quadrados mínimos, explorando sua interpretação geométrica. Seja $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ e $x \in \mathbb{R}^n$ e considere o sistema linear

$$Ax = b.$$

Essa é a expressão de um sistema equações lineares, comum em problemas de ajuste de curvas. Como, em geral, é comum que o número de restrições, ou equações, seja maior que o número de parâmetros ajustáveis, trabalha-se majoritariamente com $m > n$, que é o que faremos neste trabalho. Nesse caso, dizemos que $Ax = b$ é um sistema sobre-determinado.

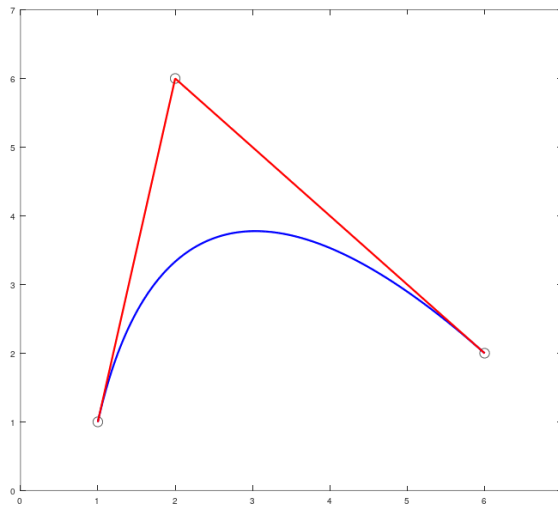


Figura 2: Bézier quadrática para os pontos $(1,1)$, $(2,6)$ e $(6,2)$.

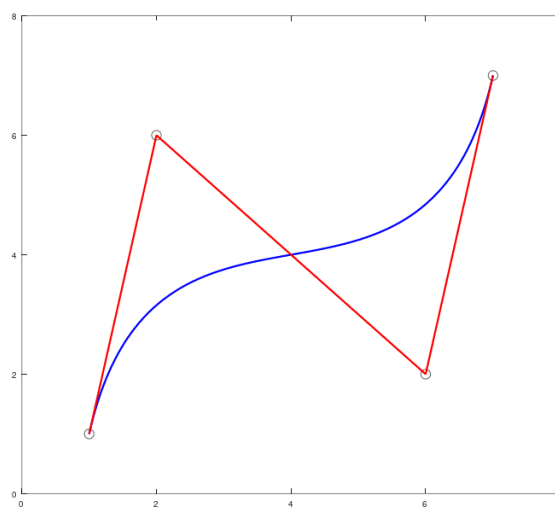


Figura 3: Bézier quadrática para os pontos $(1,1)$, $(2,6)$, $(6,2)$ e $(7,7)$.

Geometricamente, podemos associar A a um subespaço de \mathbb{R}^m , $\mathcal{R}(A)$, sendo as n colunas de A os vetores que geram o subespaço. Assim, resolver o sistema equivale a responder a pergunta “Qual é a combinação linear das colunas de A gera o vetor b ?”

Agora note que b , como um vetor qualquer do espaço \mathbb{R}^m , não necessariamente pertencerá a $\mathcal{R}(A)$. Ou seja, x pode não ser possível escrever b como combinação linear das colunas de A . O que podemos fazer é reformular a pergunta, como “Qual é a combinação linear das colunas de A é a mais próxima de b ?”. Matematicamente, esta pergunta consiste em determinar x que minimize

$$\|Ax - b\|_2^2.$$

No contexto desse novo problema, x armazena os coeficientes da combinação linear das colunas de A , de forma que $Ax \in \mathcal{R}(A)$, e Ax é o vetor deste espaço que possui a menor a b . Se b pertencer a este subespaço, $Ax = b$. No entanto, este é um caso especial, e o mais provável que aconteça com um b qualquer é que ele não pertença a $\mathcal{R}(A)$. É possível mostrar que este problema é resolvido pela projeção ortogonal de b em $\mathcal{R}(A)$ (*Strang* [2006]).

A partir desse mesmo livro, obtemos que se Ax é a componente de b em $\mathcal{R}(A)$, $Ax - b$ é a componente de b que é ortogonal a esse subespaço, e portanto, podemos dizer que $A^T(Ax - b) = 0$, pois isso é o mesmo que calcular o produto escalar de $Ax - b$ com todas as colunas de A . Por fim, manipulando esta última equação, obtemos a tradicional solução para o problema de quadrados mínimos, $x = (A^T A)^{-1} A^T b$.

A partir disso, podemos tirar uma segunda conclusão: para obter a projeção ortogonal, c , de um vetor qualquer y em $\mathcal{R}(A)$, basta fazer $c = Ax = A(A^T A)^{-1} A^T y$. Assim, é razoável definir, utilizando a definição de inversa generalizada de A como $A^\dagger = (A^T A)^{-1} A^T$, a matriz $A(A^T A)^{-1} A^T = AA^\dagger$ como o projetor ortogonal de A , ou

$$P_A = AA^\dagger. \tag{3}$$

Estendendo um pouco o conceito, se quisermos a projeção ortogonal de y no subespaço ortogonal ao gerado por A , P_A^\perp , fazemos

$$P_A^\perp y + P_A y = y,$$

$$P_A^\perp y = y - P_A y,$$

$$P_A^\perp y = (I - P_A)y,$$

Como isso vale para qualquer $y \in \mathbb{R}^m$,

$$P_A^\perp = I - P_A$$

$$P_A^\perp = I - AA^\dagger. \quad (4)$$

2.3 Os métodos de Newton e de Gauss-Newton

O problema de encontrar os zeros de uma função é um tópico fundamental na matemática. No caso de uma função ou sistema de funções afim, a solução é obtida a partir do método de quadrados mínimos, visto anteriormente. No entanto, no caso das funções não lineares, encontrar soluções exatas pode ser um problema complexo, e os métodos para encontrá-las, quando existem, costumam ser específicos para cada função. Sendo assim, foi útil desenvolver um método que fosse eficaz para aproximar os zeros de funções genéricas. Um dos métodos mais conhecidos e utilizados para resolver esse problema é o método de Newton, que faz uso das propriedades diferenciais das funções para buscar as soluções.

Seja $f : \mathbb{R} \rightarrow \mathbb{R}$ com ao menos n derivadas contínuas. Buscamos encontrar um x tal que $f(x) = 0$. Utilizando os polinômios de Taylor, temos que

$$f(x) = \sum_{k=0}^n \frac{f^{(k)}(a)}{k!} \cdot (x - a)^k + \mathcal{O}((x - a)^{n+1}).$$

A partir disso, podemos tirar aproximações para $f(x)$ na vizinhança de a . A primeira dessas aproximações é

$$f(x) \simeq f(a) + f'(a) \cdot (x - a) + \mathcal{O}((x - a)^2) = r(x) + \mathcal{O}((x - a)^2).$$

Note que $r(x)$ é uma função afim. Usando as propriedades dos polinômios de Taylor (Stewart [2010]), temos que na vizinhança de a , $f(x)$ pode ser aproximada por $r(x)$. Logo, podemos supor que a raiz de $r(x)$ possivelmente está mais próximo do zero de $f(x)$ do que o próprio ponto a . Supondo que esse ponto de fato está mais próxima

da solução, se repetirmos a expansão de Taylor a partir da raiz de $r(x)$, encontraremos uma nova aproximação para a função, dessa vez mais próxima do ponto que buscamos. Note como isso pode dá origem a um método iterativo para encontrarmos a solução do problema. Seguindo o passo a passo de partir de um ponto a , encontrar a reta afim que aproxima $f(x)$ em a , calcular sua raiz, e utilizar esse ponto como um novo ponto de partida. Se nossa suposição estiver certa, poderemos chegar arbitrariamente próximo ao ponto desejado, isso é, se ele existir. Matematicamente, cada iteração é dada por

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}, f'(x_k) \neq 0. \quad (5)$$

Esse é precisamente o cerne do método de Newton, que já foi extensivamente estudado e aplicado, e teve sua eficácia atestada. O método possui, claro, limitações, porém mesmo assim o método de Newton é extremamente eficaz para grande parcela das aplicações.

Uma das principais aplicações dessa técnica é em problemas de otimização. Neles, ao invés de buscar o zero de uma função, busca-se seus pontos críticos, ou seja, máximos e mínimos locais ou pontos de sela. Embora ambos esses problemas possam parecer diferentes de início, eles possuem uma grande relação. É um resultado bastante conhecido o de que em pontos críticos, a derivada da função é zero (Stewart [2010]), mas o inverso não necessariamente é verdade. Porém, em geral, buscar pontos críticos de $f(x)$ é o mesmo que buscar pontos em que $f'(x) = 0$, ou seja, basta aplicar o Método de Newton para $f'(x)$.

No entanto, para funções $f : \mathbb{R}^n \rightarrow \mathbb{R}$, o método anterior não basta. Sendo assim, buscaremos adaptá-lo para este caso.

No caso das funções multivariáveis, seus pontos críticos estão nos pontos em que $\nabla f(x) = 0$ (Marsden and Tromba [1996]). Analogamente ao método tradicional de Newton, começaremos, então, pela expansão dos polinômios de Taylor de $\nabla f(x) = F(x)$, mas agora utilizando sua forma vetorial. Expandindo apenas até o segundo termo, temos a aproximação

$$F(x) \simeq F(x_0) + J(x)(x - x_0),$$

em que $J(x)$ é o Jacobiano de $F(x_0)$, dado por

$$J(x) = \begin{bmatrix} \nabla F_1(x) \\ \vdots \\ \nabla F_n(x) \end{bmatrix}. \quad (6)$$

Assim, como buscamos o ponto em que $F(x) = 0$, temos que

$$J(x_0) \cdot x \simeq -F(x_0) + J(x_0) \cdot x_0,$$

$$x \simeq x_0 - J(x_0)^{-1}F(x_0).$$

Assim como fizemos com o método de Newton, basta reformatar essa expressão para obtermos nosso método iterativo

$$x_{k+1} = x_k - J(x_k)^{-1}F(x_k). \quad (7)$$

Esse método é conhecido como método de Gauss-Newton. Ele sofre das mesmas limitações do método de Newton, mas também possui a mesma eficácia daquele (Strang [2006]).

3 Ajuste em uma curva contínua

O objetivo desta secção é encontrar um método para aproximar um trecho de uma curva parametrizada qualquer no espaço \mathbb{R}^2 , dada certas restrições, por uma curva Bézier cubica, de modo que o ajuste fique graficamente semelhante a figura original. Para tal aproximação, necessitaremos tanto da expressão para a curva que queremos aproximar, quanto de sua primeira e segunda derivada. Definimos, então, $p : \mathbb{R} \rightarrow \mathbb{R}^2$, tal que

$$p(t) = (p_x(t), p_y(t)) = (x, y). \quad (8)$$

O trecho da curva a ser aproximado será $p(t)$ tal que $t \in [t_0, t_1]$. Além disso, definimos $p(t_0) = P_0$ e $p(t_1) = P_3$.

Uma Bézier cubica no plano é definida como $b : \mathbb{R} \rightarrow \mathbb{R}^2$, tal que

$$b(t) = A \cdot (1 - t)^3 + B \cdot 3(1 - t)^2t + C \cdot 3(1 - t)t^2 + D \cdot t^3 \quad (9)$$

para $t \in [0, 1]$ e $A, B, C, D \in \mathbb{R}^2$.

Desta forma, podemos ver que, para encontrar a aproximação, devemos achar os valores de A, B, C, D em função de $p(t), p'(t), p''(t), t_0, t_1$.

Começamos definindo que as extremidades de $p(t)$ devem coincidir com as extremidades de $b(t)$. Como $p(t)$ começa em P_0 e vai até P_3 , e sabendo que as pontas de $b(t)$ são os pontos A e D (substituir na expressão da Bézier $t = 0$ e $t = 1$ para verificar), basta definirmos $A = P_1 = p(t_0)$ e $D = P_3 = p(t_1)$.

Desta forma, nos resta apenas encontrar B e C . Como ambos são pontos do plano cartesiano, podemos defini-los como $B = P_1 = (x_1, y_1)$ e $C = P_2 = (x_2, y_2)$. Ou seja, devemos encontrar as quatro incógnitas, x_1, y_1, x_2 e y_2 , e para isso, precisamos resolver um sistema com quatro equações envolvendo essas variáveis.

3.1 Primeira condição

A primeira condição é que ambas as curvas devem ter a mesma derivada nas extremidades. Sabemos que em uma curva Bézier, os pontos B e C se localizam nas retas tangentes às extremidades da curva, portanto podemos afirmar que os pontos P_1 e P_2 deverão estar na reta tangente a $p(t)$ nos pontos P_0 e P_3 .

Podemos parametrizar uma reta qualquer a partir de um vetor diretor V e um ponto em que ela passa P como $r(u) = P + u \cdot V$.

Logo, uma forma de dizer que o ponto P_1 esta na reta tangente a curva $p(t)$ no ponto $p(t_0)$ é definir que

$$P_1 = p(t_0) + r_1 \cdot \frac{p'(t_0)}{\|p'(t_0)\|} = P_0 + r_1 \cdot \frac{p'(t_0)}{\|p'(t_0)\|}, \text{ para um } r_1 \in \mathbb{R}.$$

O mesmo argumento pode ser aplicado para o ponto P_2 , nos dando a equação

$$P_2 = p(t_1) + r_2 \cdot \frac{p'(t_1)}{\|p'(t_1)\|} = P_3 + r_2 \cdot \frac{p'(t_1)}{\|p'(t_1)\|}, \text{ para um } r_2 \in \mathbb{R}.$$

Desta forma, conseguimos deixar P_1 e P_2 em função de r_1 e r_2 , reduzindo o problema de quatro variáveis para duas. Com o intuito de facilitar contas futuras, faremos um leve ajuste as expressões encontradas acima.

Como $\frac{p'(t_1)}{\|p'(t_1)\|}$ e $\frac{p'(t_1)}{\|p'(t_1)\|}$ são ambos vetores normalizados, ou seja, possuem norma igual a um, nós podemos representá-los, respectivamente, como

$$\frac{p'(t_1)}{\|p'(t_1)\|} = (\cos(\theta_1), \sin(\theta_1)) \text{ e } \frac{p'(t_1)}{\|p'(t_1)\|} = (\cos(\theta_2), \sin(\theta_2)) \quad (10)$$

Utilizaremos tal substituição pois isso futuramente possibilitará certas manipulações matemáticas a fim de simplificar certas expressões, mas em momento algum o valor dos ângulos serão necessários.

Por fim, faremos uma ultima manipulação: substituindo $t = 1$ em $b'(x)$, veremos que $b'(1) = -C$. Ou seja, P_2 fica no sentido contrário ao vetor da derivada de $b(t)$ no ponto $t = 1$. Como estamos definindo que as derivadas das duas curvas devem ser iguais neste ponto, o ponto P_2 também está no sentido contrário ao vetor $\frac{p'(t_1)}{\|p'(t_1)\|}$ e, portanto, r_2 seria negativo na expressão $P_2 = P_3 + r_2 \cdot \frac{p'(t_1)}{\|p'(t_1)\|}$. Logo, para limitarmos o resultado de r_2 a apenas números positivos, inverteremos seu sinal.

Desta forma, obtemos as seguintes expressões:

$$P_1 = P_0 + r_1 \cdot (\cos(\theta_1), \sin(\theta_1)), \quad (11)$$

$$P_2 = P_0 - r_2 \cdot (\cos(\theta_2), \sin(\theta_2)). \quad (12)$$

3.2 Segunda condição

Para nossa aproximação, também definiremos que ambas as curvas devem ter a mesma curvatura em seus pontos de extremidade. Tome a seguinte expressão para a curvatura de uma curva parametrizada plana qualquer $p(u) = (x(u), y(u))$ (Penner [2019]):

$$k(u) = \frac{x'(u) \cdot y''(u) - x''(u) \cdot y'(u)}{(x'(u)^2 + y'(u)^2)^{\frac{3}{2}}}. \quad (13)$$

Observa-se que com essa expressão e com as expressões das derivadas da curva $p(t)$, é possível encontrar os valores $K_p(t_0)$ e $K_p(t_1)$ para as curvaturas nas extremidades da curva que queremos aproximar. Portanto nos resta apenas igualar estes valores com a curvatura da Bézier, nos mesmos pontos P_0 e P_1 .

Para isso, primeiro precisaremos representar a curva $b(t)$ pelas suas componentes x e y , então encontrar as duas primeiras derivadas de tais expressões, e finalmente substituir tais expressões na expressão de $k(t)$ para $t = 0$ e $t = 1$. Faremos isso.

A partir da equação 9 considere as substituições:

$$P_0 = (x_0, y_0), \quad (14)$$

$$P_3 = (x_3, y_3). \quad (15)$$

Trocando A , B , C e D por P_0 , P_1 , P_2 e P_3 em 9 e substituindo esses pontos por 11, 12, 14 e 15, temos que

$$b(t) = (x(t), y(t)), \text{ com}$$

$$x(t) = x_0 \cdot (1-t)^3 + (x_0 + r_1 \cdot \cos\theta_1) \cdot 3(1-t)^2t + (x_3 - r_2 \cdot \cos\theta_2) \cdot 3(1-t)t^2 + x_3 \cdot t^3 \text{ e}$$

$$y(t) = y_0 \cdot (1-t)^3 + (y_0 + r_1 \cdot \sin\theta_1) \cdot 3(1-t)^2t + (y_3 - r_2 \cdot \sin\theta_2) \cdot 3(1-t)t^2 + y_3 \cdot t^3.$$

Derivando essas expressões, obtemos

$$x'(t) = x_0 \cdot (-3t^2 + 6t - 3) + (x_0 + r_1 \cdot \cos\theta_1) \cdot (9t^2 - 12t + 3) + (x_3 - r_2 \cdot \cos\theta_2) \cdot (-9t^2 + 6t) + x_3 \cdot (3t^2)$$

e

$$y'(t) = y_0 \cdot (-3t^2 + 6t - 3) + (y_0 + r_1 \cdot \sin\theta_1) \cdot (9t^2 - 12t + 3) + (y_3 - r_2 \cdot \sin\theta_2) \cdot (-9t^2 + 6t) + y_3 \cdot (3t^2).$$

Por fim, derivando uma segunda vez, obtemos

$$x''(t) = x_0 \cdot (-6t + 6) + (x_0 + r_1 \cdot \cos\theta_1) \cdot (18t - 12) + (x_3 - r_2 \cdot \cos\theta_2) \cdot (-18t + 6) + x_3 \cdot (6t)$$

e

$$y''(t) = y_0 \cdot (-6t + 6) + (y_0 + r_1 \cdot \sin\theta_1) \cdot (18t - 12) + (y_3 - r_2 \cdot \sin\theta_2) \cdot (-18t + 6) + y_3 \cdot (6t).$$

Com essas expressões, conseguimos calcular a curvatura da Bézier no ponto inicial ($t = 0$) e final ($t = 1$), em função de r_1 e r_2 . Para $t = 0$:

$$x'(0) = -3x_0 + 3(x_0 + r_1 \cos \theta_1)$$

$$y'(0) = -3y_0 + 3(y_0 + r_1 \sin \theta_1)$$

$$x''(0) = 6x_0 - 12(x_0 + r_1 \cos \theta_1) + 6(x_3 - r_2 \cos \theta_2)$$

$$y''(0) = 6y_0 - 12(y_0 + r_1 \sin \theta_1) + 6(y_3 - r_2 \sin \theta_2)$$

Com isso, substituindo em 13, obtemos:

$$k_b(0) = 6 \frac{(3r_1 \cos \theta_1)(-y_0 - 2r_1 \sin \theta_1 + y_3 - r_2 \sin \theta_2) - (3r_1 \sin \theta_1)(-x_0 - 2r_1 \cos \theta_1 + x_3 - r_2 \cos \theta_2)}{(9(r_1 \cdot \cos \theta_1)^2 + 9(r_2 \cdot \sin \theta_1)^2)^{\frac{3}{2}}}$$

$$k_b(0) = 18r_1 \frac{\cos\theta_1(-y_0 + y_3) - \sin\theta_1(-x_0 + x_3) - r_2(\cos\theta_1 \sin\theta_2 - \sin\theta_1 \cos\theta_2)}{(3r_1)^3}$$

$$k_b(0) = \frac{2}{3r_1^2} (\cos\theta_1(-y_0 + y_3) - \sin\theta_1(-x_0 + x_3) - r_2(\cos\theta_1 \sin\theta_2 - \sin\theta_1 \cos\theta_2))$$

Igualando essa expressão com $K_p(t_1)$, temos

$$r_1^2 = \frac{2}{3K_p(t_0)^2} (\cos\theta_1(-y_0 + y_3) - \sin\theta_1(-x_0 + x_3) - r_2(\cos\theta_1 \sin\theta_2 - \sin\theta_1 \cos\theta_2)) \quad (16)$$

Se repetirmos os mesmos passos para $t = 1$, obteremos a expressão:

$$r_2^2 = \frac{2}{3K_p(t_1)^2} (\cos\theta_2(-y_0 + y_3) - \sin\theta_2(-x_0 + x_3) + r_1(\cos\theta_1 \sin\theta_2 - \sin\theta_1 \cos\theta_2)) \quad (17)$$

Observe que r_1 e r_2 são as únicas incógnitas das expressões. Logo, podemos reescrever as equações como

$$r_1^2 = a_1 + r_2 \cdot b_1 \quad (18)$$

$$r_2^2 = a_2 + r_1 \cdot b_2 \quad (19)$$

Sendo

$$a_1 = \frac{2}{3K_p(t_0)^2} \cdot (\cos\theta_1(-y_0 + y_3) - \sin\theta_1(-x_0 + x_3))$$

$$a_2 = \frac{2}{3K_p(t_1)^2} \cdot (\cos\theta_2(-y_0 + y_3) - \sin\theta_2(-x_0 + x_3))$$

$$b_1 = \frac{2}{3K_p(t_0)^2} \cdot (\cos\theta_1 \sin\theta_2 - \sin\theta_1 \cos\theta_2)$$

$$b_2 = \frac{2}{3K_p(t_1)^2} \cdot (\cos\theta_1 \sin\theta_2 - \sin\theta_1 \cos\theta_2)$$

Isolando r_2 em 18 e substituindo em 19, obtemos o polinômio de quarto grau, na variável r_1

$$r_1^4 + (-2a_1)r_1^2 + (-b_2b_1^2)r_1 + (a_1 - a_2b_1^2) = 0$$

Como, por construção, definimos R_1 como positivo, descartaremos as raízes complexas e negativas do polinômio. Caso haja mais de uma raiz positiva, buscaremos aquela que, quando aplicada em 18, retornará um r_2 positivo também. Com isso, basta substituir r_1 e r_2 em 11 e 12, e finalmente obteremos todos os pontos que definem a Bézier que melhor aproxima dada curva, de acordo com os critérios utilizados.

3.3 Implementação, Testes e Resultados

Nesse estudo, buscamos uma forma de ajustar as curvas para que o resultado fique graficamente satisfatório, assim a análise dos resultados será feita qualitativamente.

Nas figuras 3.3 e 3.3, é possível observar como o intervalo em que o ajuste é feito afeta a aproximação da curva. Para intervalos menores, a diferença entre as curvas chega a ser praticamente imperceptível a olho nu, e conforme o intervalo cresce, a tendência é que a diferença aumente. Isso é consequência direta de como a aproximação é construída, baseada nas propriedades analíticas da curva em suas extremidades. Essas características tendem a ser bem representativas dos entornos mais próximos desses pontos, portanto

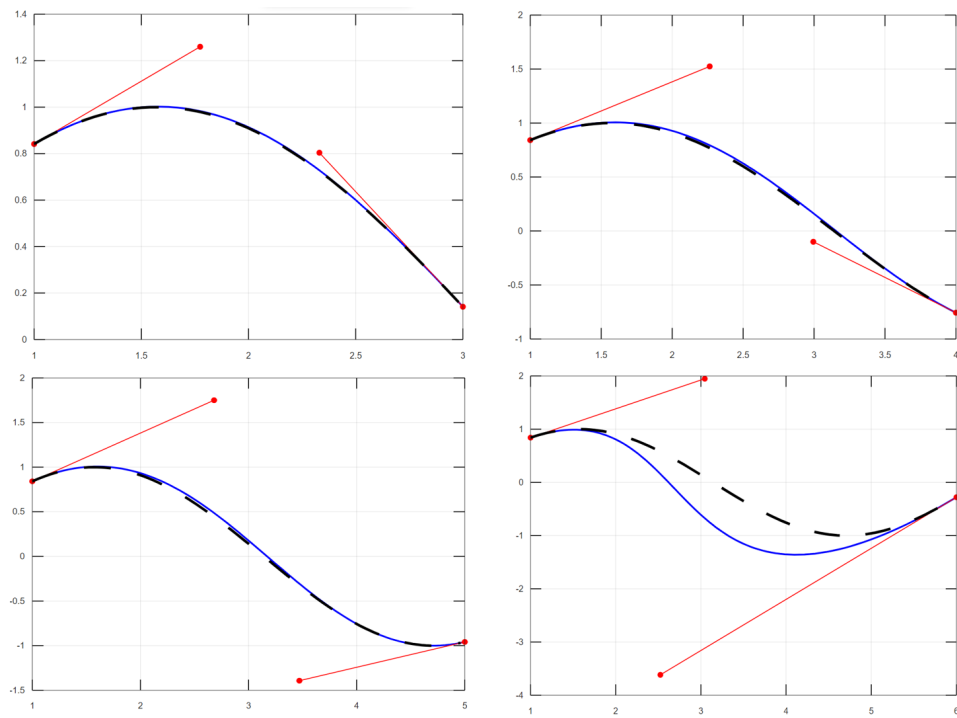


Figura 4: Aproximação da função $y = \sin(x)$ em diferentes intervalos, de $x = 1$ á, respectivamente, $x = 3$, $x = 4$, $x = 5$ e $x = 6$

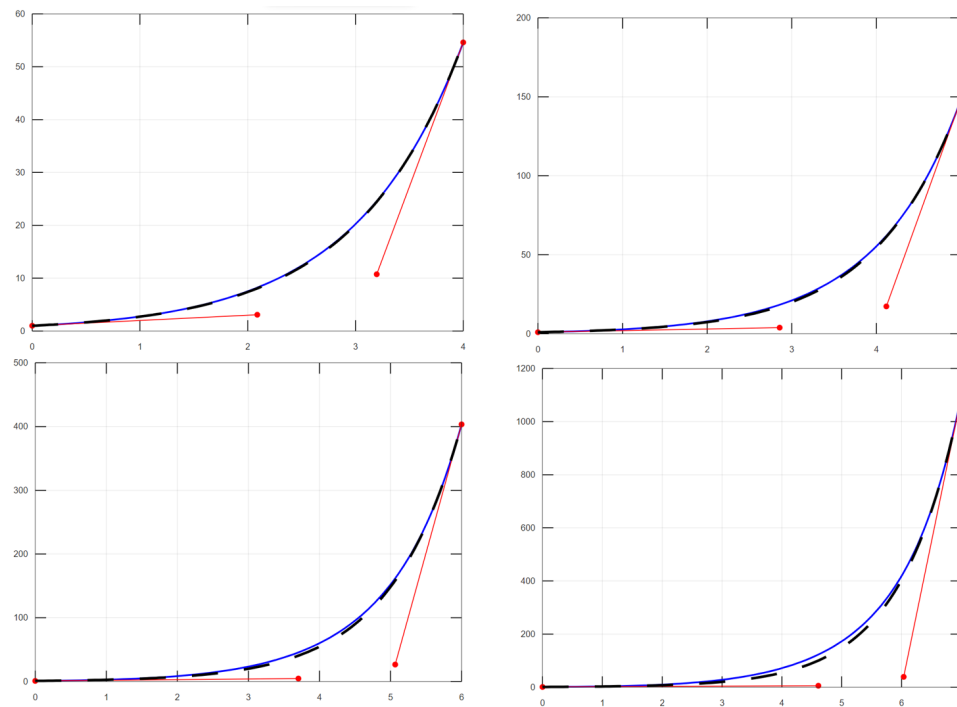


Figura 5: Aproximação da função $y = e^x$ em diferentes intervalos, de $x = 0$ á, respectivamente, $x = 4$, $x = 5$, $x = 6$ e $x = 7$

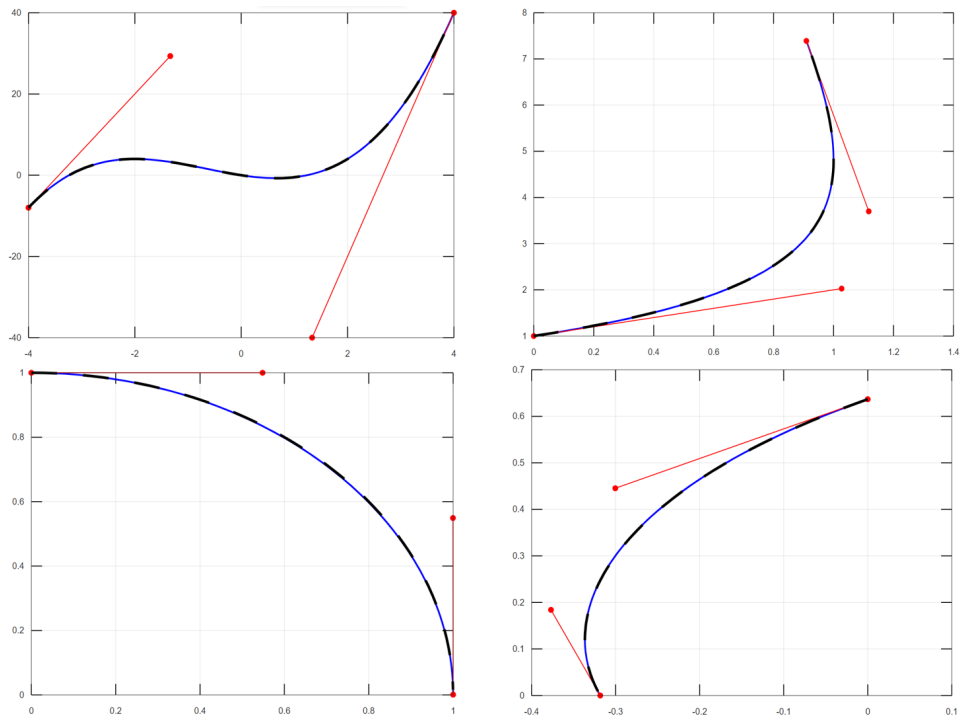


Figura 6: Aproximação das curvas paramétricas respectivas por uma Bézier: $(t, 0.5t^3 + t^2 - 2t), t \in (-4, 4)$; $(\text{sen}(t), e^t), t \in (0, 2)$; $(\text{cos}(t), \text{sen}(t)), t \in (0, \frac{\pi}{2})$; $(\frac{\text{cos}(t)}{t}, \frac{\text{sen}(t)}{t}), t \in (\frac{\pi}{2}, \pi)$.

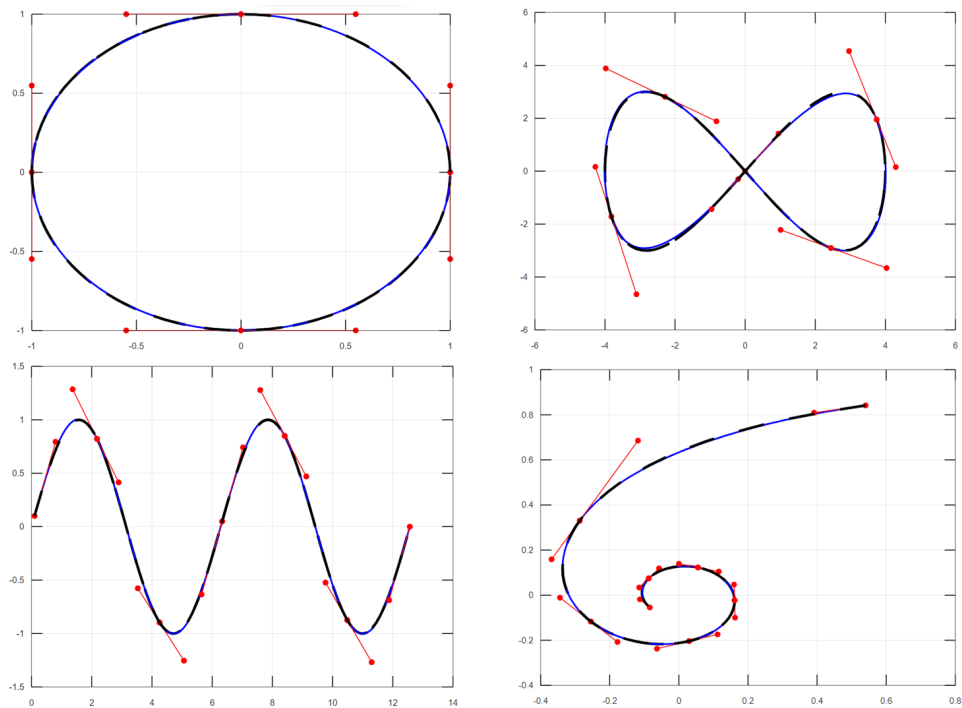


Figura 7: Aproximação das curvas paramétricas respectivas por diversas curvas Bézier: circunferência unitária, 4 Bézier; $(4\text{sen}(0.5t), 3\text{sen}(t)), t \in (0.1, 4\pi)$, 5 Bézier; $(t, \text{sen}(t)), t \in (0.1, 4\pi)$, 6 Bézier; $(\frac{\text{cos}(t)}{t}, \frac{\text{sen}(t)}{t}), t \in (1, 10)$, 7 Bézier.

é razoável que conforme o tamanho do trecho aproximado cresce, os pontos no meio da curva, que ficam cada vez mais longe das extremidades, comecem a perder relação com as características locais da curva nos pontos analisados.

Podemos observar na 3.3, inclusive, que o método utilizado funciona inclusive para curvas paramétricas com expressões mais complexas, e que o fator que de fato influencia na qualidade da aproximação é o tamanho do intervalo.

O ajuste, contudo, não precisa ficar limitado a pequenos trechos. Para fazer um ajuste a um trecho de curva maior, basta dividir este trecho em partes menores, fazer o ajuste de uma Bézier para cada um deles, e por fim uní-las, formando uma spline 3.3. Se no caso do ajuste de uma única curva, o fator determinante para a qualidade do ajuste era o tamanho do intervalo, é de se esperar que no caso da spline, esse fator é o número de intervalos em que o trecho é dividido. Além disso, também é provável que a maneira com que os intervalos são divididos também influenciem na qualidade da representação, no entanto este fator não foi investigado neste estudo.

4 Ajuste em uma amostra de pontos

Neste secção trabalharemos o problema do ajuste de uma curva Bézier a um conjunto de pontos ordenados. Considere que suas coordenadas x e y estão ordenadas, respectivamente, nos vetores u e v . Buscamos encontrar os pontos de controle de uma curva Bézier de um dado grau n que minimizam a distancia entre os pontos amostrados e a curva.

Como ajustaremos uma Bézier de um grau arbitrário n , então usaremos a definição geral de uma Bézier:

$$b(t) = (x(t), y(t)) = \sum_{j=0}^n B_j^n(t) (x_j, y_j)$$

tal que

$$B_j^n(t) = \binom{n}{j} t^j (1-t)^{n-j} \quad (20)$$

e que x e y são os vetores em \mathbb{R}^n com as coordenadas dos pontos de controle da Bézier.

Matematicamente, podemos dizer que dado o conjunto de m pontos (u_j, v_j) , para $j \in (1, m)$ e o grau n da Bézier, buscamos minimizar, em função de x , y e t ,

$$f(x, y, t) = \sum_{i=1}^m \left[\left(u_i - \sum_{j=0}^n B_j^n(t_j) \cdot x_j \right)^2 + \left(v_i - \sum_{j=0}^n B_j^n(t_j) \cdot y_j \right)^2 \right]. \quad (21)$$

Para melhorar a visualização do problema, podemos formatar essa expressão em uma forma vetorial. Para isso, utilizaremos a matriz de Bernstein, B_n , definida por

$$B_{n,(i,j)} = B_j^n(t_i).$$

Com isso, podemos reescrever a expressão que queremos minimizar como

$$f(x, y, t) = \left\| \begin{bmatrix} u \\ v \end{bmatrix} - \begin{bmatrix} B_n & 0 \\ 0 & B_n \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \right\|_2^2. \quad (22)$$

Note que o problema é semelhante ao problema de quadrados mínimos. A diferença é que enquanto que no problema de quadrados mínimos minimiza-se a soma das distâncias, ao quadrado, entre (u_j, v_j) e $f(u_j)$ de um conjunto inicial de pontos, neste problema minimizaremos a soma das distâncias, ao quadrado, entre (u_j, v_j) e $b(t_j)$. Perceba que no primeiro problema, pegamos a diferença entre os pontos (u_j, v_j) , dados, e os pontos da função em u_j . Ou seja, a variável da função nos pontos interessados fazem parte do conjunto inicial de dados. Já no problema que estamos trabalhando, a variável t_j não é dada inicialmente, e, portanto, deve ser obtida em função dos dados iniciais do problema, que no caso são u , v , e n .

No entanto, repare que com o vetor t em mãos, o problema fica de fato igual ao de QM, cuja solução é simples e conhecida. Desta forma, o principal desafio será encontrar esse vetor t ótimo.

4.1 Construção do Método

Solucionando x e y para um B_n fixo, ou seja, para um dado t , o resultado dos quadrados mínimos é

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} B_n^\dagger & 0 \\ 0 & B_n^\dagger \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix},$$

sendo B_n^\dagger a inversa generalizada 2.2 de B_n , dada por $B_n^\dagger = (B_n^T B_n)^{-1} B_n^T$. Com isso, substituímos $\begin{bmatrix} x \\ y \end{bmatrix}$ na expressão 22,

$$\begin{aligned} f(x, y, t) &= \left\| \begin{bmatrix} u \\ v \end{bmatrix} - \begin{bmatrix} B_n B_n^\dagger & 0 \\ 0 & B_n B_n^\dagger \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \right\|_2^2 = \left\| \begin{bmatrix} u \\ v \end{bmatrix} - \begin{bmatrix} B_n B_n^\dagger u \\ B_n B_n^\dagger v \end{bmatrix} \right\|_2^2 \\ &= \left\| \begin{bmatrix} (I - B_n B_n^\dagger)u \\ (I - B_n B_n^\dagger)v \end{bmatrix} \right\|_2^2 \end{aligned} \quad (23)$$

Agora analisando esta expressão, note que $I - B_n B_n^\dagger$ é o projetor ortogonal de B_n . Além disso, repare que a expressão dependerá apenas de u e v , que são dados, e de B_n , que por sua vez depende de n , também dado, e t . Logo, podemos definir esta uma função como resíduo, e é ela que gostaríamos de minimizar. Assim,

$$\mathbf{r}(\mathbf{t}) = \left\| \begin{bmatrix} P_B^\perp u \\ P_B^\perp v \end{bmatrix} \right\|_2^2. \quad (24)$$

Como vimos na seção de projetores, eles independem de qual base estamos utilizando para o subespaço. Logo, podemos manipular a matriz B_n a fim de encontrar uma base mais favorável para se trabalhar. Dentre as infinitas bases plausíveis, há um grupo que se destaca, que é das bases ortonormais. Estas possuem propriedades extras que podem ser exploradas nos cálculos. Para encontrá-la, o método mais tradicional é aplicar a decomposição QR.

Fazendo esta decomposição em B_n , obteremos que

$$B_n = QR, \quad (25)$$

tal que as colunas de Q formam uma base ortonormal para o espaço \mathbb{R}^m , e R armazena as combinações lineares das colunas de Q que formam a base de B_n . Mas mais do que isso, as n primeiras colunas de Q formam uma base ortogonal para o subespaço formado pelas colunas de B_n , e as colunas restantes formam uma base para o subespaço ortogonal

a ele. Assim, podemos definir como Q_1 a matriz com as n primeiras colunas de Q e Q_2 a matriz com as colunas restantes. Desta maneira, temos que

$$P_B^\perp = Q_2 Q_2^T. \quad (26)$$

Com essas informações, temos um problema bem definido de minimização de uma função não linear multivariável, e para resolvê-lo, utilizaremos o método de Gauss-Newton.

4.2 Aplicando Gauss-Newton

Para aplicar o método de Gauss-Newton, definiremos a função que queremos minimizar como

$$f(t) = \left\| \begin{bmatrix} P_B^\perp u \\ P_B^\perp v \end{bmatrix} \right\|_2^2 = \|r(t)\|_2^2 \quad (27)$$

Como visto em 2.3, precisaremos calcular o Jacobiano de $f(t)$, cuja a n -ésima coluna é dada por

$$J_j = \begin{bmatrix} \frac{\partial P_B^\perp}{\partial t_j} u \\ \frac{\partial P_B^\perp}{\partial t_j} v \end{bmatrix}. \quad (28)$$

Assim, precisamos encontrar forma de avaliar $\frac{\partial P_B^\perp}{\partial t_j} u$ e $\frac{\partial P_B^\perp}{\partial t_j} v$. Para isso, podemos utilizar o seguinte teorema (Borges and Pastva [2002]):

Teorema 1. Seja A uma matriz $m \times n$ e A^\dagger sua inversa generalizada. Então:

$$\frac{\partial P_A^\perp}{\partial t_k} = -P_A^\perp \frac{\partial A}{\partial t_k} A^\dagger - (P_A^\perp \frac{\partial A}{\partial t_k} A^\dagger)^T. \quad (29)$$

A demonstração completa passo a passo pode ser vista na referência. Aplicando o teorema no caso do nosso problema, encontramos que

$$\frac{\partial P_B^\perp}{\partial t_k} = -P_B^\perp \frac{\partial B_n}{\partial t_k} B_n^\dagger - (P_B^\perp \frac{\partial B_n}{\partial t_k} B_n^\dagger)^T. \quad (30)$$

Substituindo a equação 25 na 30, surge que

$$\frac{\partial P_B^T}{\partial t_j} = -Q_2 Q_2^T \frac{\partial B_n}{\partial t_j} B_n^\dagger - (-Q_2 Q_2^T \frac{\partial B_n}{\partial t_j} B_n^\dagger)^T.$$

Dessa forma, podemos reescrever J como

$$J = - \begin{bmatrix} \bar{J}_u \\ \bar{J}_v \end{bmatrix} - \begin{bmatrix} \bar{J}_u^T \\ \bar{J}_v^T \end{bmatrix}, \text{ tal que a } j\text{-ésima coluna } \bar{J}_{w,j} = Q_2 Q_2^T \frac{\partial B_n}{\partial t_j} B_n^\dagger w. \quad (31)$$

Como na matriz de Bernstein o único lugar em que t_j aparece é na fileira j , a matriz $\frac{\partial B_n}{\partial t_j}$ será toda de zeros, com exceção da fileira j . Logo, segue que a matriz

$$\frac{\partial B_n}{\partial t_j} B_n^\dagger = \begin{bmatrix} 0 \\ \dots \\ c_j \\ \dots \\ 0 \end{bmatrix}, \text{ tal que} \quad (32)$$

$$c_j = \frac{\partial}{\partial t_j} \begin{bmatrix} B_0^n(t_j) & \dots & B_n^n(t_j) \end{bmatrix} B_n^\dagger \quad (33)$$

Logo

$$\bar{J}_{w,j} = Q_2 Q_2^T \frac{\partial B_n}{\partial t_j} B_n^\dagger w = Q_2 Q_2^T \begin{bmatrix} 0 \\ \dots \\ c_j \cdot w \\ \dots \\ 0 \end{bmatrix}, \text{ e} \quad (34)$$

$$\bar{J}_w = Q_2 Q_2^T \begin{bmatrix} c_1 \cdot w & \dots & 0 \\ & c_j \cdot w & \\ 0 & \dots & c_m \cdot w \end{bmatrix} = Q_2 Q_2^T \text{diag} \left(\begin{bmatrix} c_1 \cdot w \\ \dots \\ c_j \cdot w \\ \dots \\ c_m \cdot w \end{bmatrix} \right) = Q_2 Q_2^T \text{diag} \left(\begin{bmatrix} c_1 \\ \dots \\ c_j \\ \dots \\ c_m \end{bmatrix} w \right)$$

$$= Q_2 Q_2^T \text{diag} \left(\begin{bmatrix} \frac{\partial}{\partial t_0} [B_0^n(t_0) \dots B_n^n(t_0)] B_n^\dagger \\ \dots \\ \frac{\partial}{\partial t_j} [B_0^n(t_j) \dots B_n^n(t_j)] B_n^\dagger \\ \dots \\ \frac{\partial}{\partial t_m} [B_0^n(t_m) \dots B_n^n(t_m)] B_n^\dagger \end{bmatrix} w \right)$$

$$\bar{J}_w = Q_2 Q_2^T \text{diag}(B'_n B_n^\dagger w) \quad (35)$$

$$\text{tal que } B'_n(t) = \begin{bmatrix} \frac{\partial}{\partial t_0} B_0^n(t_0) & \dots & \frac{\partial}{\partial t_0} B_n^n(t_0) \\ \dots & & \dots \\ \frac{\partial}{\partial t_j} B_0^n(t_j) & \dots & \frac{\partial}{\partial t_j} B_n^n(t_j) \\ \dots & & \dots \\ \frac{\partial}{\partial t_m} B_0^n(t_m) & \dots & \frac{\partial}{\partial t_m} B_n^n(t_m) \end{bmatrix}. \quad (36)$$

Para encontrar uma forma conveniente de calcular $B'_n(t)$, podemos explorar a estrutura de $\frac{\partial}{\partial t_j} B_k^n(t_j)$. A partir da B_{nj} , temos que, para $k \in [0, n]$ e $j \in [1, m]$

$$\frac{\partial}{\partial t_j} B_k^n(t_j) = \binom{n}{k} \cdot (k t_j^{k-1} (1-t_j)^{n-k} - (n-k) t_j^k (1-t_j)^{n-k-1}) \quad (37)$$

$$= k \binom{n}{k} t_j^{k-1} (1-t_j)^{n-k} - k \binom{n}{k} t_j^k (1-t_j)^{n-k-1} \quad (38)$$

$$= n \binom{n-1}{k-1} t_j^{k-1} (1-t_j)^{(n-1)-(k-1)} - n \binom{n-1}{k} t_j^k (1-t_j)^{(n-1)-k} \quad (39)$$

$$= n(B_{k-1}^{n-1} - B_k^{n-1}) \quad (40)$$

Assim, temos que

$$B'_n = n \left(\begin{bmatrix} \underline{0} & B_{n-1} \end{bmatrix} - \begin{bmatrix} B_{n-1} & \underline{0} \end{bmatrix} \right). \quad (41)$$

Com isso, obtemos uma expressão bem definida para a Jacobiana, partindo da 31 e aplicando a 35

$$J = \begin{bmatrix} Q_2 Q_2^T \text{diag}(B'_n B_n^\dagger u) - (B'_n B_n^\dagger)^T \text{diag}(Q_2 Q_2^T u) \\ Q_2 Q_2^T \text{diag}(B'_n B_n^\dagger v) - (B'_n B_n^\dagger)^T \text{diag}(Q_2 Q_2^T v) \end{bmatrix}, \quad (42)$$

e com ela e com $\nabla f(t) = \frac{1}{2}r(t)$, temos todas as ferramentas para aplicar o método iterativo de Gauss-Newton (7)

4.3 Implementação, Testes e Resultados

Uma bateria de testes foram executados testando diferentes aspectos do programa, o que permitiu atestar a estabilidade to método. O algoritmo depende de três entradas, o conjunto de pontos a serem ajustados, o grau da curva ajustada e a tolerância do método iterativo.

Utilizando o critério de parada abordado na 7, em especial com a norma infinito, foi testado diversos ajustes com os valores $\epsilon \in \{10^{-0}, 10^{-1}, 10^{-2}, 10^{-3}\}$. A diferença do ajuste utilizando o primeiro e o segundo valores tendeu a ser visivelmente distinta. Embora nenhum tenha gerado resultados muito fora do esperado, a diferença visual entre os ajustes tendeu a ser considerável, mas o número de iterações feitas também. Esse foi bem sensível ao conjunto inicial de dados no caso mais exigente, variando de 2 a 6 iterações necessárias, enquanto que no caso mais tolerante, o método se restringiu a uma iteração em todos os casos. A mesma tendência se manteve conforme a exigência aumentava, mas

com saltos cada vez menores tanto na diferença visual do ajuste, quanto no número de iterações.

Os testes isolando a variável do grau da Bézier ajustada foram úteis para verificar que o grau mínimo para se obter um ajuste razoável depende intrinsecamente do conjunto inicial de dados. Note que o primeiro ajuste da figura 4.3 não foi capaz de gerar uma curva realmente próxima dos dados iniciais. Isso é consequência do limite natural de formas que uma Bézier de grau 2 pode assumir. O mesmo não acontece com os ajustes seguintes. Sobre eles, observe que os resultados ficaram progressivamente mais próximos dos pontos dados, mas com diferenças cada vez mais sutis entre si, e as custas de iterações cada vez mais caras computacionalmente.

Por fim, também foram testados conjuntos de pontos iniciais variados, para testar a flexibilidade do método. Observe na imagem 4.3, a gama de formas que uma Bézier de grau 3 consegue assumir, e como o método funciona tanto para casos com um número maior quanto menor de pontos.

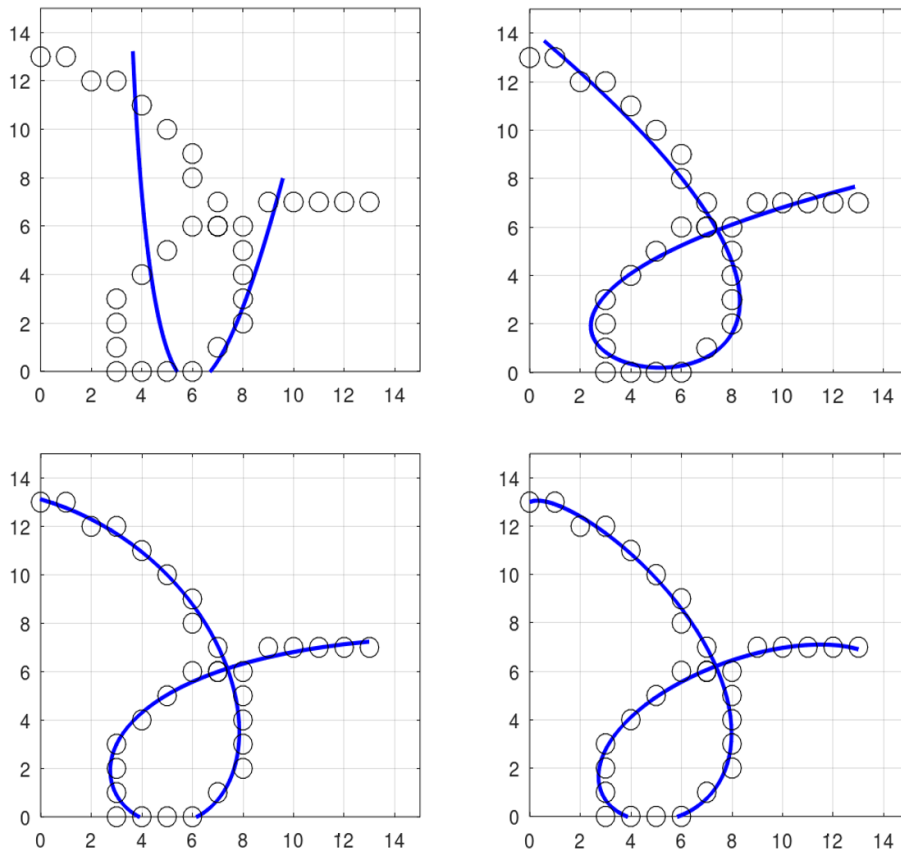


Figura 8: Ajuste de uma Bézier de grau, respectivamente, 2, 3, 4 e 5, em um mesmo conjunto de pontos ordenados utilizando um mesmo critério de parada.

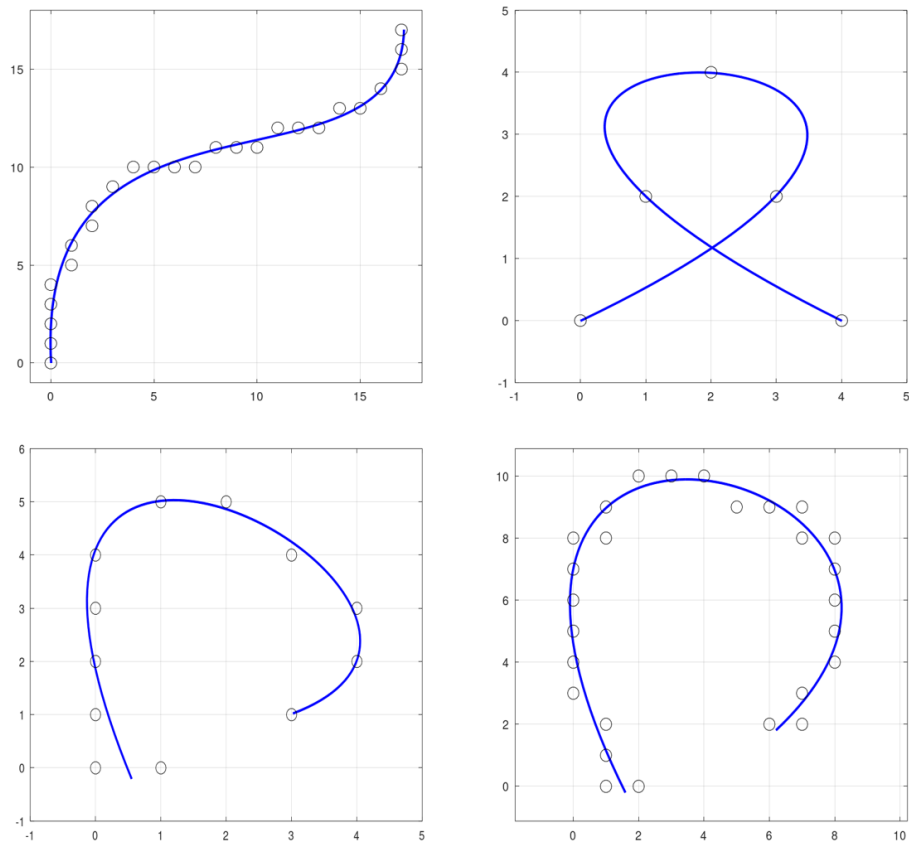


Figura 9: Ajustes de Béziers cúbicas em diferentes conjuntos de dados.

Referências

- C. F. Borges and T. A. Pastva. Total least squares fitting of bézier and b-spline curves to ordered data. *Comput. Aided Geom. Des.*, 19:275–289, 2002.
- G.G. Lorentz. *Bernstein Polynomials*. AMS Chelsea Publishing Series. Chelsea Publishing Company, 1986.
- J.E. Marsden and A. Tromba. *Vector Calculus*. Number v. 1. W.H. Freeman, 1996.
- Alvin Penner. Fitting a cubic bézier to a parametric function. *The College Mathematics Journal*, 50(3):185–196, 2019.
- J. Stewart. *CÁlculo Volume 1 - 6^ª edição*. Cengage Learning Edições Ltda., 2010.
- G. Strang. *Linear Algebra and Its Applications*. Thomson, Brooks/Cole, 2006.