3. Interpolation problem. Splines. Bézier curves.

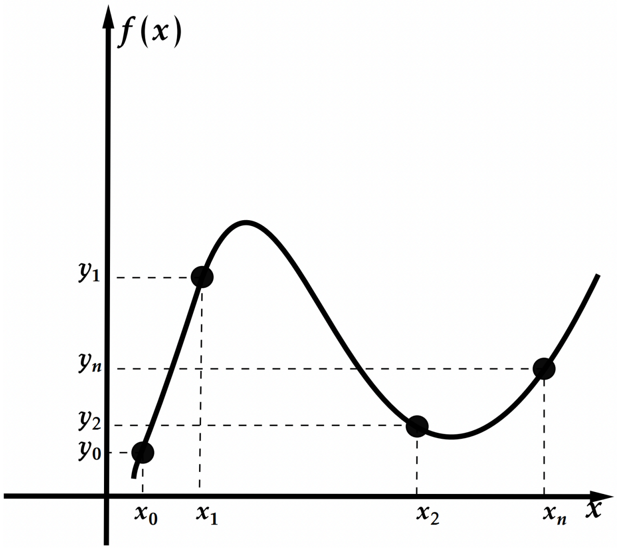
Classical polynomial interpolation problem

Classical polynomial interpolation problem is formulated as follows. Suppose some function $f(x) = a_0+a_1x+a_2x^2+\ldots+a_{n-1}x^{n-1}+a_nx^n$ is a polynomial of degree $\leq n$. Given the values of function $f(x)$

$$\begin{cases} f(x_0) = y_0 \\ \vdots \\ f(x_n) = y_n \end{cases},$$

recover $f(x)$. This means that we need to solve following system of equations

$$\begin{cases} a_0 + a_1x_0 + \ldots + a_nx_0^n = y_0 \\ a_0 + a_1x_1 + \ldots + a_nx_1^n = y_1 \\ \vdots \\ a_0 + a_1x_n + \ldots + a_nx_n^n = y_n \end{cases}.$$



Example of polynomial interpolation

Since we can rewrite the system using matrix notation, the problem is equivalent to solving following matrix equation

$$V\vec{a} = \vec{y},$$

where

$$V = \begin{bmatrix} 1 & x_0 & \ldots & x_0^n \\ 1 & x_1 & \ldots & x_1^n \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & \ldots & x_n^n \end{bmatrix}, \qquad \vec{a} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix}, \qquad \vec{y} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}.$$

Here only $\vec{a}$ is unknown. We call matrix of a form V a Vandermonde matrix.

Lemma: Vendermonde determinant

$$\begin{aligned} v &:= v(x_0, \ldots, x_n) \\ &:= \det V \\ &= (x_1 - x_0)(x_2 - x_0)\ldots(x_2 - x_1)\ldots(x_n - x_{n-1}) \\ &= \prod_{0\leq i<j\leq n} (x_j - x_i). \end{aligned}$$

Example 1: $\det V = v(x_0, x_1) = \begin{vmatrix} 1 & x_0 \\ 1 & x_1 \end{vmatrix} = x_1 - x_0.$

Example 2: $\det V = v(x_0, x_1, x_2) = \begin{vmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{vmatrix} = (x_2 - x_0)(x_1 - x_0)(x_2 - x_1).$

Theorem: Solution of polynomial interpolation

Following statements are equivalent

1. System $V\vec{a} = \vec{y}$ has unique solution $\vec{a} = V^{-1}\vec{y}$
2. $\det(V) \neq 0$
3. $x_0, \dots, x_n$ are different

Proof: The result follows from Lemma: Vendermonde determinant. □

Theorem: Lagrange form of interpolating polynomial

The solution of system $V\vec{a} = \vec{y}$ could expressed in following the form

$$f(x) = \sum_{i=0}^{n} \frac{v(x_0, \dots, x, \dots, x_n)}{v(x_0, \dots, x_i, \dots x_n)} y_i = \sum_{i=0}^{n} y_i \frac{(x - x_0) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)}{(x_i - x_0) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)}.$$

We call $f(x)$ Lagrange interpolating polynomial.

Proof: The proof is by direct calculation. □

Note

The Lagrange interpolating polynomial is the unique polynomial of lowest degree that interpolates a given set of data.

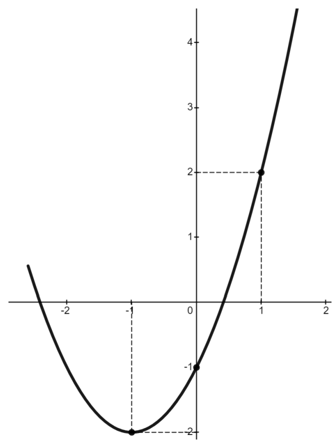
Example 3: Given set of points

$$\begin{aligned}
(x_0, y_0) &= (-1, -2), \\
(x_1, y_1) &= (0, -1), \\
(x_2, y_2) &= (1, 2),
\end{aligned}$$

find polynomial of lowest degree that interpolates all points.

Since Lagrange interpolating polynomial is the unique polynomial of lowest degree that interpolates a given set of data, it is a solution we need. Let's compute it

$$\begin{aligned}
f(x) &= -2 \cdot \frac{(x-0)(x-1)}{(-1-0)(-1-1)} - 1 \cdot \frac{(x+1)(x-1)}{(0+1)(0-1)} + 2 \cdot \frac{(x+1)(x-0)}{(1+1)(1-0)} \\
&= -x(x-1) + x^2 - 1 + x(x+1) \\
&= x^2 + 2x - 1.
\end{aligned}$$



Example 3: Graph of interpolation polynomial $f(x)$ of minimal degree

Hermitian interpolation problem (interpolation with multiple knots)

Definition

A number x_1 is a root of a polynomial $f(x)$ with multiplicity d if

$$f(x) = (x - x_1)^d g(x), \quad g(x_1) \neq 0.$$

Lemma

A number x_0 is a root of multiplicity d for a polynomial $f(x)$ if and only if

$$\begin{cases} f(x_0) = 0 \\ f'(x_0) = 0 \\ \vdots \\ f^{(d-1)}(x_0) = 0 \\ f^{d}(x_0) \neq 0 \end{cases}.$$

<u>Proof:</u> Let us apply Taylor expansion at point x_0 for polynomial $f(x)$

$$\begin{aligned}(x-x_0)^d g(x) &\equiv \sum_{i=0}^{n} \frac{f^{(i)}(x_0)}{i!}(x-x_0)^i \\ &= \sum_{i=0}^{d-1} \frac{f^{(i)}(x_0)}{i!}(x-x_0)^i + \sum_{j=d}^{n} \frac{f^{(j)}(x_0)}{j!}(x-x_0)^j.\end{aligned}$$

Let us denote $a_i := \frac{f^{(i)}(x_0)}{i!}$ and $z := x - x_0$. We get

$$z^d g(x) \equiv \sum_{i=0}^{d-1} c_i z^i + \sum_{j=d}^{n} c_j z^j. \tag{1}$$

We see that left hand side of (1) is divisible by z^d, meaning that the right hand side should be divisible by z^d as well. First summation (first $d-1$ terms) of the right hand side is not divisible by z^d, while second summation (remaining terms) is divisible by z^d. This means that first $d-1$ terms' coefficients are equal to zero

$$\begin{cases} c_0 = 0 \\ \vdots \\ c_{d-1} = 0 \end{cases} \Leftrightarrow \begin{cases} f(x_0) = 0 \\ \vdots \\ f^{(d-1)}(x_0) = 0 \end{cases}. \tag{2}$$

Using (1) and (2), we get

$$g(x) \equiv \sum_{i=d}^{n} c_i z^{i-d} = c_d + c_{d+1} z + \cdots + c_n z^{n-d},$$

which means that $g(x_0) \equiv c_d$. Combing this with the fact that $g(x_0) \neq 0$, we get $c_d \neq 0 \Leftrightarrow f^{(d)}(x) \neq 0$.

□

Hermitian interpolation problem is formulated as follows. Suppose some function $f(x)$ is a polynomial of degree $\leq n-1$. Given different m knots[1] $\underbrace{x_1, x_2, \ldots, x_m}_{\text{knots}} \in \mathbb{R}$ of corresponding multiplicities $\underbrace{h_1, \ldots, h_m}_{\text{multiplicities}} \in \mathbb{N}$ with $h_1 + h_2 + \ldots + h_m = n$ and

$$\begin{cases} f(x_1) = y_1,\ f'(x_1) = y_1^{(1)}, & \ldots, & f^{(h_1-1)}(x_1) = y_1^{(h_1-1)} \\ f(x_2) = y_2,\ f'(x_2) = y_2^{(1)}, & \ldots, & f^{(h_2-1)}(x_2) = y_2^{(h_2-1)} \\ & \vdots & \\ f(x_m) = y_m,\ f'(x_m) = y_m^{(1)}, & \ldots, & f^{(h_m-1)}(x_m) = y_m^{(h_m-1)} \end{cases},$$

recover $f(x)$.

Prop Hermitian interpolation problem always has a unique solution.

Example 4: Recover polynomial $f(x)$ based on a given set of knots. The set of knots is defined as follows

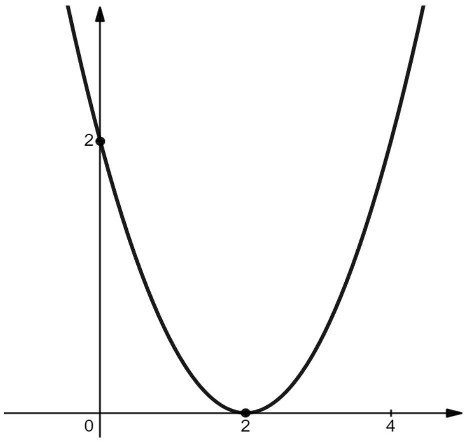
$$\begin{cases} f(0) = 2 \\ f(2) = 0,\ f'(2) = 0 \end{cases}.$$

The sum of knots multiplicities is equal to 3, which means that $f(x)$ is polynomial of degree ≤ 2

$$f(x) = ax^2 + bx + c.$$

$$\begin{cases} f(0) = 2 \\ f(2) = 0 \\ f'(2) = 0 \end{cases} \Rightarrow \begin{cases} a \cdot 0^2 + b \cdot 0 + c = 2 \\ a \cdot 2^2 + b \cdot 2 + c = 0 \\ 2a \cdot 2 + b = 0 \end{cases} \Rightarrow \begin{cases} a \cdot 0^2 + b \cdot 0 + c = 2 \\ a \cdot 2^2 + b \cdot 2 + c = 0 \\ 2a \cdot 2 + b = 0 \end{cases}$$

$$\Rightarrow \begin{cases} c = 2 \\ b = -2 \\ a = \dfrac{1}{2} \end{cases} \Longrightarrow f(x) = \frac{1}{2}x^2 - 2x + 2.$$



Example 4: Graph of interpolation polynomial $f(x)$ with multiple knots

[1] In the context of interpolation, a 'knot' and a 'point' are often used interchangeably. Both terms refer to a specific data value with coordinates (e.g., in a two-dimensional space, a point has x and y coordinates)

Polynomial Splines

Spline is a special function defined piecewise by polynomials. In interpolating problems, spline interpolation is often preferred to polynomial interpolation because it yields similar results, even when using low degree polynomials, while avoiding Runge's phenomenon[2] for higher degrees.

More formally the problem is to construct a "smooth" function $f(x)$ with knots $x_0 < x_1 \cdots < x_n$ that on each segment $I_i = [x_{i-1}, x_i]$ has form $f(x) = f_i(x)$. That is, given

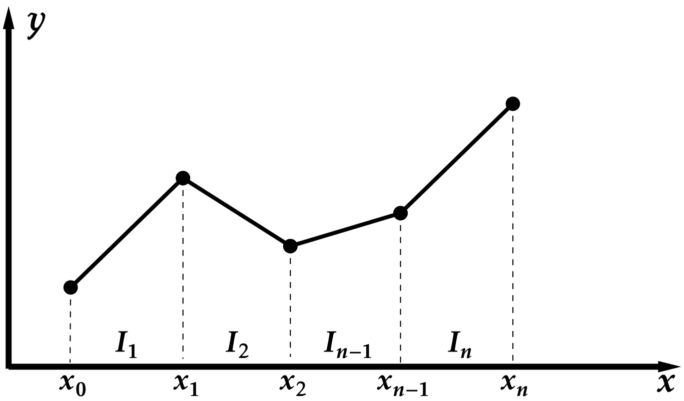
$$\begin{cases} f(x_0) = y_0 \\ f(x_1) = y_1 \\ \quad \vdots \\ f(x_n) = y_n \end{cases},$$

recover $f(x)$ in the form

$$f(x) = \begin{cases} f_1(x), \ x \in I_1 \\ f_2(x), \ x \in I_2 \\ \quad \vdots \\ f_n(x), \ x \in I_n \end{cases}.$$

Linear Spline

If by "smooth" we mean continuous then $f(x)$ could represented by linear functions $f_i(x) = a_i x + b_i$ on each segment $I_i = [x_{i-1}, x_i]$. Basically we just connect points (knots) with straight lines. Such $f(x)$ we call linear spline



Example of a linear spline

Quadratic Spline

If by "smooth" we mean with continuous derivative then $f(x)$ could represented by quadratic functions $f_i(x) = a_i x^2 + b_i x + c_i$ on each segment $I_i = [x_{i-1}, x_i]$. Moreover, every $f_i(x)$ not only satisfies $f_i(x_{i-1}) = y_{i-1}$ and $f_i(x_i) = y_i$, but also $f_i'(x_{i-1}) = f_{i-1}'(x_{i-1})$. Since for $i = 1$ last condition does not make sense, we use initial condition[3] $f_1'(x_0) = g(x)$ instead.

[2] Runge's phenomenon is a problem of oscillation at the edges of an interval that occurs when using polynomial interpolation with polynomials of high degree over a set of equispaced interpolation points. It was discovered by Carl David Tolmé Runge (1901) when exploring the behavior of errors when using polynomial interpolation to approximate certain functions

[3] Usually, for the sake of simplicity, $g(x)$ is set to be 0 or x. However, choice of $g(x)$ depends on subject area. For example, initial condition $g(x) = f_n'(x_n)$ is used for periodic processing.

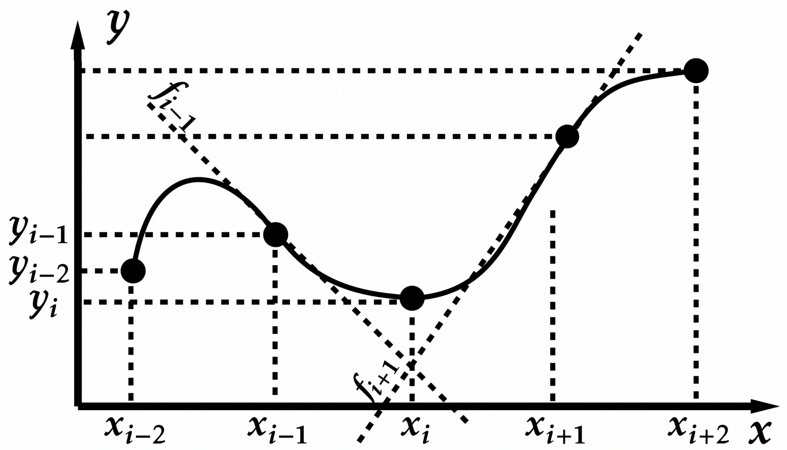
Finally, we have following formulation of the problem

$$f(x) = \begin{cases} f_1(x), & x \in I_1 \\ f_2(x), & x \in I_2 \\ \vdots & \\ f_n(x), & x \in I_n \end{cases},$$

where $f_i(x)$ satisfies

$$\begin{cases} f_i(x) = a_i x^2 + b_i x + c_i, & \forall i \\ f_i(x_i) = y_i, & \forall i \\ f_i(x_{i-1}) = y_{i-1}, & \forall i \\ f_i'(x_{i-1}) = f_{i-1}'(x_{i-1}), & i > 1 \\ f_1'(x_0) = g(x), & i = 1 \end{cases}.$$

Such $f(x)$ we call quadratic spline.



Example of quadratic spline

Cubic spline

If by "smooth" we mean with third continuous derivative then $f(x)$ could represented by cubic functions $f_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i$ on each segment $I_i = [x_{i-1}, x_i]$. Moreover, every $f_i(x)$ not only satisfies $f_i(x_{i-1}) = y_{i-1}$ and $f_i(x_i) = y_i$, but also $f_i'(x_{i-1}) = f_{i-1}'(x_{i-1})$ and $f_i''(x_{i-1}) = f_{i-1}''(x_{i-1})$. Since for $i = 1$ last conditions does not make sense, we use initial conditions $f_1'(x_0) = g_1(x)$ and $f_1''(x_0) = g_1(x)$ instead.

$$f(x) = \begin{cases} f_1(x), & x \in I_1 \\ f_2(x), & x \in I_2 \\ \vdots & \\ f_n(x), & x \in I_n \end{cases},$$

where $f_i(x)$ satisfies

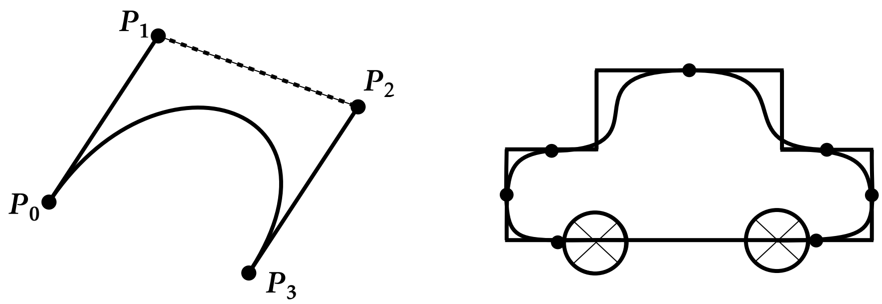
$$\begin{cases} f_i(x) = a_i x^2 + b_i x + c_i, & \forall i \\ f_i(x_i) = y_i, & \forall i \\ f_i(x_{i-1}) = y_{i-1}, & \forall i \\ f_i'(x_{i-1}) = f_{i-1}'(x_{i-1}), & i > 1 \\ f_i''(x_{i-1}) = f_{i-1}''(x_{i-1}), & i > 1 \\ f_1'(x_0) = g_1(x), & i = 1 \\ f_1''(x_0) = g_2(x), & i = 1 \end{cases}.$$

Such $f(x)$ we call cubic spline.

Bézier curves

A Bézier curve[4] is a parametric curve used in computer graphics and related fields. A set of discrete "control points"defines a smooth, continuous curve by means of a formula. Usually the curve is intended to approximate a real-world shape that otherwise has no mathematical representation or whose representation is unknown or too complicated.

Bézier curve interpolation problem is formulated as follows. Given a set of control points $P_0, P_1, \dots, P_n \in \mathbb{R}^m$ approximate the path $P_0 \to P_1 \to P_2 \to \dots \to P_n$ by a smooth parametric curve $B(t)$. Number of control points n is called degree of a Bézier curve. We denote Bézier curve that approximate k control points $P_0, P_1, \dots, P_k \in \mathbb{R}^m$ by $B_{P_0P_1\dots P_k}(t)$.



Examples of Bézier curves

Theorem: Recursive formula for Bézier curve

Bézier curve $B(t)$ could be find using following recursive formula

$$B_{P_0}(t) = P_0,\ t \in [0,1].$$
$$B_{P_0P_1\dots P_k}(t) = (1-t)B_{P_0P_k\dots P_{n-1}}(t) + tB_{P_1P_2\dots P_n}(t),\ t \in [0,1].$$

Example 5: Given a set of control points $P_0, P_1 \in \mathbb{R}^m$ approximate the path $P_0 \to P_1$ by a smooth parametric curve $B(t)$. Using recursive formula for Bézier curve, we get

$$B_{P_0}(t) = P_0,$$
$$B_{P_1}(t) = P_1,$$
$$B(t) = B_{P_0P_1}(t) = (1-t)B_{P_0} + tB_{P_1} = (1-t)P_0 + tP_1.$$

Theorem: Explicit formula for Bézier curve

Bézier curve $B(t)$ could be find using following explicit formula

$$B(t) = \sum_{i=0}^{n} P_i \cdot b_{n,i}(t),$$

where $b_{n,i}$ are Bernstein polynomials of a form

$$b_{n,i} = C_n^i (1-t)^{n-i} t^i.$$

[4]The Bézier curve is named after French engineer Pierre Bézier (1910–1999), who used it in the 1962 for designing curves for the bodywork of Renault cars. The method developed first developed in 1959 by Paul de Casteljau for Citroen cars. However, at the time Paul de Casteljau could not publish his works