INTERPOLATING CURVES

Rodrigo Silveira

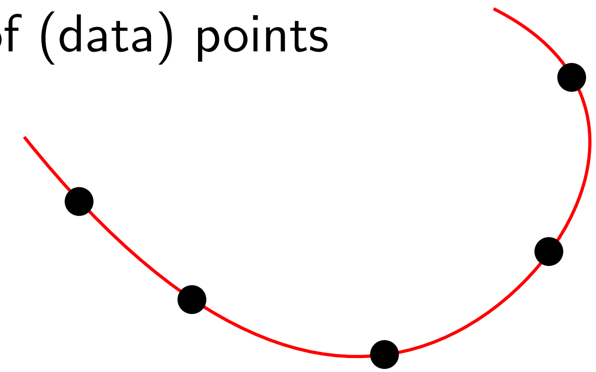
Curve and Surface Design
Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya

INTERPOLATING A SET OF POINTS

Interpolation problem

Given a set of points (e.g., data samples) originated by some unknown function, the goal is to estimate the values of the function on locations between the known values

a set of (data) points



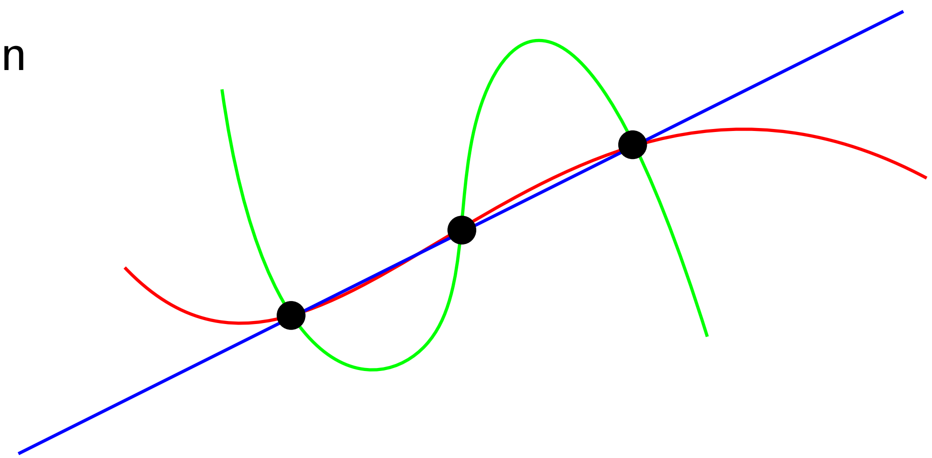
a curve interpolating the points

In other (more precise) words:

Given $P_0, \ldots, P_n$ in $\mathbb{R}^d$ our goal is to find a curve $\gamma(t)$ such that $\gamma(t_i) = P_i$ for some values of $t_0, \ldots, t_n$.

Observe that:

- The parameter values t_i are not known
- There are many curves that work

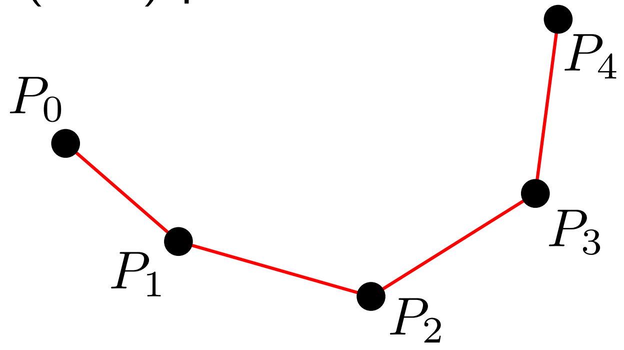


Piecewise linear interpolation

The simplest solution is piecewise linear interpolation: use a *polygonal line* that has the points $P_1, \ldots, P_n$ as vertices, and line segments $P_{i-1}P_i$ as edges.

Parametrization

a set of (data) points



a curve interpolating the points

Given a set P of $n+1$ points $P_0, P_1, \ldots, P_n$ in $\mathbb{R}^d$, and an increasing sequence of $n+1$ real values $t_0 < t_1 < \cdots < t_n$, the following curve interpolates the points in P:

$\gamma : [t_0, t_n] \to \mathbb{R}^d$

$$\gamma(t) = \frac{t_i - t}{t_i - t_{i-1}} P_{i-1} + \frac{t - t_{i-1}}{t_i - t_{i-1}} P_i \quad \text{if } t \in [t_{i-1}, t_i] \text{ for all } i = 1, \ldots, n$$

Observations

- γ_{t-1} is continuous: trivially in (t_{i-1}, t_i) for all i, and also at each t_i for all i because $\gamma(t_i)$ is well defined (i.e., consecutive line segments coincide at data points)
- γ_{t-1} is not differentiable at the points t_i (unless three consecutive points are aligned)

Piecewise linear interpolation

How to choose the values $t_0 < t_1 < \cdots < t_n$?

- Make each subinterval $[t_{i-1}, t_i]$ of the same length (say, 1)

  We get $t_i = i$ for each i, and $[t_{i-1}, t_i] = [i-1, i]$, therefore:

  $\gamma(t) = (i-t)P_{i-1} + (t-i+1)P_i$ if $t \in [i-1, i]$ for all $i = 1, \ldots, n$

  Parametrization $[0, n] \to \mathbb{R}^d$. *Speed* possibly different on each edge.

- Make each subinterval $[t_{i-1}, t_i]$ of length $||P_i - P_{i-1}|| = d_i$

  We get $t_i = \sum_{k=1}^{i} d_k$, for each i, therefore:

  $$\gamma(t) = \frac{\sum_{k=1}^{i} d_k - t}{d_i} P_{i-1} + \frac{t - \sum_{k=1}^{i-1} d_k}{d_i} P_i$$ if $t \in [t_{i-1}, t_i]$ for all $i = 1, \ldots, n$

  Parametrization $[0, \sum_{k=1}^{n} d_k] \to \mathbb{R}^d$, **unit-speed parametrization**

Variation diminishing property

An important property for any interpolating curve

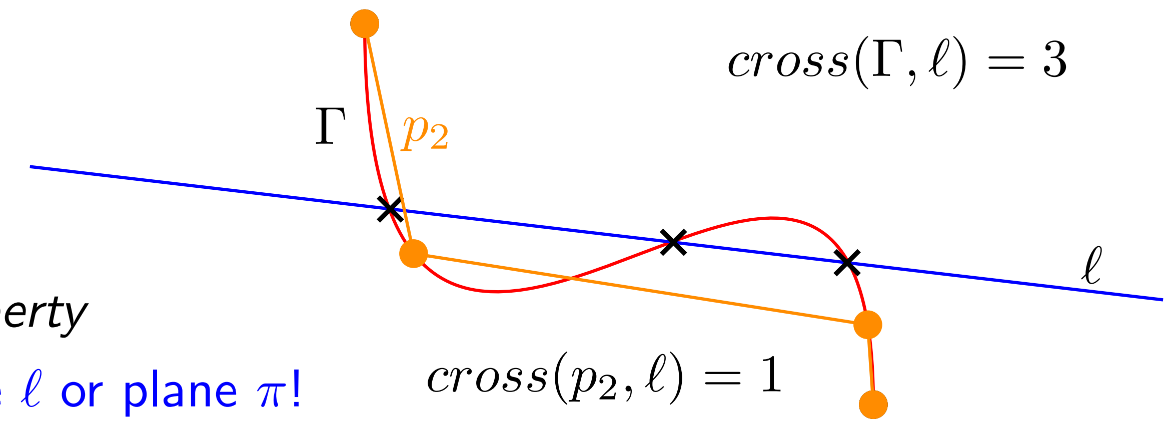
Suppose that the points you want to interpolate are samples from an unknown curve Γ

Given a curve Γ in $\mathbb{R}^2$ or $\mathbb{R}^3$, and **any** line ℓ (in $\mathbb{R}^2$) or plane π (in $\mathbb{R}^3$), let us denote $cross(\Gamma, \ell)$ or $cross(\Gamma, \pi)$ the number of crossings (i.e., intersection points) of Γ and ℓ or π.

Consider any piecewise linear interpolation $p(t)$ of Γ. Then:

$cross(\Gamma, \ell) \geq cross(p, \ell)$ (in $\mathbb{R}^2$)

$cross(\Gamma, \pi) \geq cross(p, \pi)$ (in $\mathbb{R}^3$)



This is called the *variation diminishing property*

$\rightarrow$ note that this must hold for **any** line ℓ or plane π!

Implies that the interpolating curve (p) does not wiggle much more than the original one (Γ)

Affine invariance of piecewise linear interpolation

Let $\gamma(t)$ be a curve in $\mathbb{R}^d$, and let $p(t)$ be a piecewise linear interpolation of $\gamma(t)$. Let f be any affinity $f : \mathbb{R}^d \to \mathbb{R}^d$. Then $f(p(t))$ is a piecewise linear interpolation of $f(\gamma(t))$.

Why? Proof sketch:

Let $P_0, P_1, \ldots, P_n$ be the vertices of the polygonal line of the piecewise linear interpolation $p(t)$.

We have:

- $f(p(t))$ is a polygonal line
- the vertices of $f(p(t))$ are $f(P_0), \ldots, f(P_n)$
- for all i, $f(P_i)$ is a point in $f(\gamma(t))$, since P_i is a point in $\gamma(t)$

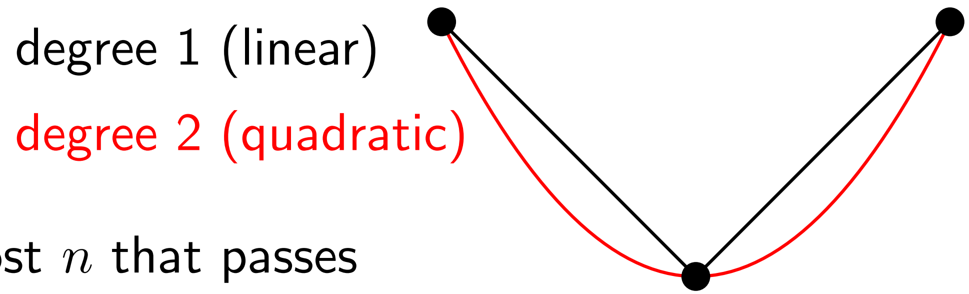
Therefore, $f(p(t))$ is a linear interpolation of $f(\gamma(t))$

Thus it is the same to (i) first linearly interpolate, then apply affine transformation, than (ii) first apply affine transformation, then linearly interpolate

Using higher degree polynomials

Idea: find a higher degree polynomial that interpolates the points in a smoother way

Let $P_0, P_1, \ldots, P_n$ with $P_i = (x_i, y_i)$, be $n+1$ points in $\mathbb{R}^2$



Interpolation theorem

There exists a unique polynomial of degree at most n that passes through $P_0, P_1, \ldots, P_n$

Proof

1) Uniqueness (if it exists, it is unique)

- Suppose that there are two polynomials $p(x)$ and $q(x)$ that interpolate the points. Consider then $r(x) = p(x) - q(x)$
- $r(x)$ is also a polynomial of degree at most n, but it has $n+1$ different roots: one at each x_i (since $r(x_i) = p(x_i) - q(x_i) = y_i - y_i = 0$)
- But a degree-n polynomial different from zero can have at most n roots! Then $r(x)$ must be the zero polynomial, i.e., $r(x) = 0$, implying that $p(x) = q(x)$!

Using higher degree polynomials

Interpolation theorem

There exists a unique polynomial of degree at most n that passes through $P_0, P_1, \ldots, P_n$

Proof (cont'd)

2) Existence (it exists!)

- We define the following auxiliary polynomials (known as *Lagrange weights*)

$$L_i^n(x) = \frac{\prod_{j \neq i}(x - x_j)}{\prod_{j \neq i}(x_i - x_j)}$$

Observe:

- $L_i^n(x)$ is a polynomial of degree n for all $i = 0, \ldots, n$
- $L_i^n(x_j) = 1$ if $j = i$, and $L_i^n(x_j) = 0$ if $j \neq i$

Here is our interpolating polynomial:

$$p(x) = \sum_{i=0}^{n} L_i^n(x) y_i$$

Observe:

- $L_i^n(x)$ is a polynomial of degree n
- $p(x_i) = y_i$ for all $i = 0, \ldots, n$

Langrange polynomial

$$p(x) = \sum_{i=0}^{n} L_i^n(x) y_i$$

- $L_i^n(x)$ is a polynomial of degree n
- $p(x_i) = y_i$ for all $i = 0, \ldots, n$

This way to express the unique polynomial that interpolates $n+1$ points is known as *Lagrange polynomial*

(other ways exist, but they are reformulations of the same—unique!—polynomial)

Would it be possible to achieve the same with a lower degree polynomial?

No!

That is: the only degree that *always* allows to interpolate $n+1$ points is degree n

The exception is when two or more data points lie on a low-degree polynomial.

Example: n points on a line can be interpolated with a polynomial of degree just 1

Langrange polynomial

Lemma: n is the minimum degree that guarantees the existence of an interpolating polynomial for *any* set of $n+1$ distinct points.

Why? **Proof sketch:** (by induction on n)

- Base case: $n = 1$. Then we have only two points P_0, P_1, and we know that two points are required to determine a line (i.e., a polynomial of degree 1)

- Induction step: assume that there exists a set of n points $P_1, \ldots, P_n$ whose interpolating polynomial has degree exactly $n-1$ (i.e., with lower degree it is not possible)

  Let P_0 be a point that does not lie on the polynomial curve that interpolates $P_1, \ldots, P_n$. Since that polynomial of degree $n-1$ is unique, and it does not go through P_0, then the polynomial through $P_0, P_1, \ldots, P_n$ must be different, and thus must have higher degree.

Langrange polynomial

$$p(x) = \sum_{i=0}^{n} L_i^n(x) y_i$$

- $L_i^n(x)$ is a polynomial of degree n
- $p(x_i) = y_i$ for all $i = 0, \ldots, n$

Is the Lagrange polynomial affine invariant?

Yes!

How can we verify that?

We need to see if the weights that multiply the points (well, the y_is) add up to 1

Clever idea: let's choose a really simple set of points to interpolate (we can choose any, because the $L_i^n(x)$ weights are independent of the points)

Consider $n+1$ distinct points on the function $f(x) = 1$, that is: $P_i = (x_i, 1)$ for $i = 0, \ldots, n$. Then the unique polynomial of degree at most n that interpolates them is easy: $p(x) = 1$

$$1 = p(x) = \sum_{i=0}^{n} L_i^n(x) y_i = \sum_{i=0}^{n} L_i^n(x) \cdot 1 = \sum_{i=0}^{n} L_i^n(x) \ \rightarrow \sum_{i=0}^{n} L_i^n(x) = 1$$

Therefore, the Langrange polynomial is affine invariant

Parametric version of Langrange polynomial

$p(x) = \sum_{i=0}^{n} L_i^n(x) y_i$ $\rightarrow$ explicit equation

This only works to interpolate values from a function, but doesn't work for any set of points (e.g., as soon as two points have the same x coordiante, it doesn't work)

Parametric version (arbitrary dimension)

Let $P_0, P_1, \ldots, P_n$ be $n+1$ points in R^d, and consider $0 = t_0 < t_1 < \cdots < t_n = 1$.

$$L_i^n(t) = \frac{\prod_{j \neq i}(t - t_j)}{\prod_{j \neq i}(t_i - t_j)}, \text{ for } i = 0, \ldots, n \qquad \gamma(t) = \sum_{i=0}^{n} L_i^n(t) P_i$$

- $L_i^n(t)$ is a polynomial of degree n in t $\rightarrow$ so is $\gamma(t)$
- $L_i^n(t_j) = 1$ if $j = i$, and $L_i^n(t_j) = 0$ if $j \neq i$
- $\gamma(t_i) = P_i$
- For each value of t, $\gamma(t)$ is an affine combination of $P_0, \ldots, P_n$, with weights $L_i^n(t)$ that (you can prove) add up to 1 $\rightarrow$ this interpolation is also affine invariant

Parametric version of Langrange polynomial

Remarks

- As before, the parametric Lagrange interpolation is not unique: we are free to choose the parameter values $t_1, \ldots, t_{n-1}$.
    - Uniform version: $t_i - t_{i-1} = 1$ or $= \frac{1}{n}$
    - Non-uniform version: all other cases

You will play with this in Lab assignment 3

- There is a matricial expression of Lagrange interpolation

For example, for only two points P_0, P_1

We have $t_0 = 0$, $t_1 = 1$,

$$L_0^1(t) = \frac{t-t_1}{t_0-t_1} = \frac{t-1}{-1} = -t+1 = (t,1)(-1,1)^t$$

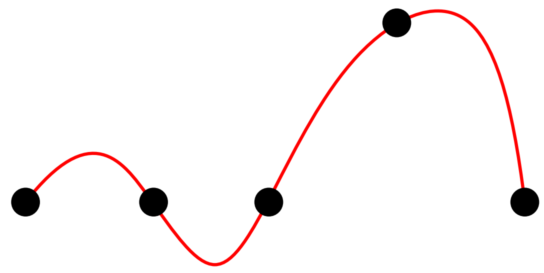
$$L_1^1(t) = \frac{t-t_0}{t_1-t_0} = \frac{t-0}{1} = t = (t,1)(1,0)^t$$

$$p(t) = L_0^1(t)P_0 + L_1^1(t)P_1 =$$

$$= (t,1)\begin{pmatrix} -1 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} P_0 \\ P_1 \end{pmatrix}$$

See bibliography for the matrix version for larger n

Issues with polynomial interpolation



- The high degree of the polynomial produces a curve with higher roughness (i.e., it can wiggle a lot) than probably desired

  $\rightarrow$ The variation diminishing property is not satisfied!

- Intuitively: adding data points should improve the resulting polynomial curve. But that is not always the case! This is known as *Runge's phenomenon*

  You will play with this in Lab assignment 3

- Lagrange's formula requires $\Theta(n^2)$ additions and products, which is quite a lot (although more efficient versions exist)

- If one has computed $\gamma(t)$ for n points and needs to add one extra point, everything needs to be recomputed

- Lagrange's formula is not numerically stable: small variations in the input points can produce large variations in the final curve

- The method is not easy to make interactive: if the curve is not what one wants, (and you cannot modify the data points) all you can do is to add more points