Constructing $A\mathbf{x} = \mathbf{b}$: Curve Interpolation

Linear Algebra

Josh Engwer

TTU

31 August 2015

Curve Interpolation (Definition)

A useful procedure in many applications is **curve interpolation**:

Definition

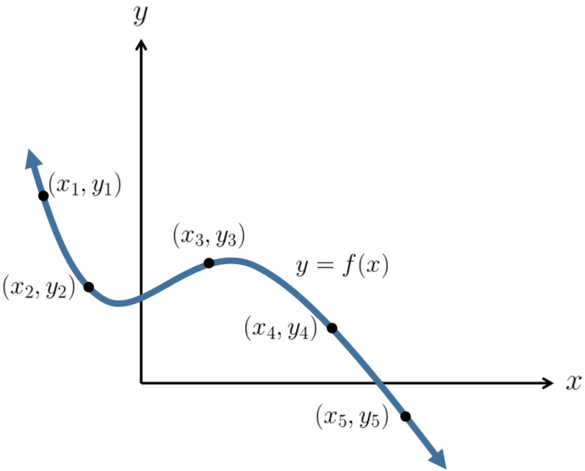
(Curve Interpolation)

Let function $f(x)$ be **continuous**. Then:
The curve $y = f(x)$ **interpolates** a set of n points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ if:

$$\begin{aligned} f(x_1) &= y_1 \\ f(x_2) &= y_2 \\ \vdots & \quad \vdots \\ f(x_n) &= y_n \end{aligned}$$

i.e. A curve **interpolates** a set of points if the curve **contains** all the points.

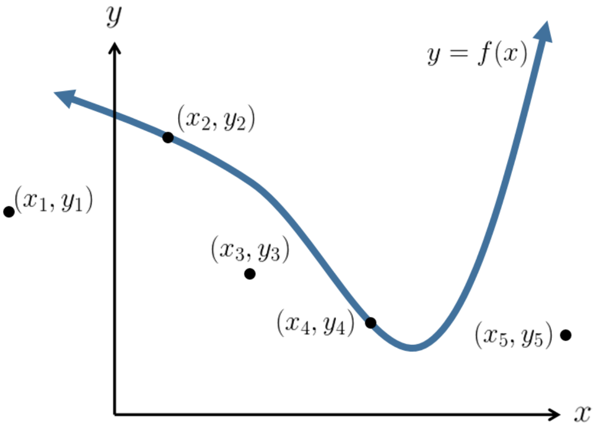
Curve Interpolation (Visual Example)



This is Curve Interpolation
since curve $y = f(x)$ contains points $(x_1, y_1), \ldots, (x_5, y_5)$

i.e. $f(x_1) = y_1,\ f(x_2) = y_2,\ f(x_3) = y_3,\ f(x_4) = y_4,\ f(x_5) = y_5$

Curve Interpolation (Visual Non-Example)



This is **NOT** Curve Interpolation
since curve $y = f(x)$ does **not** contain points $(x_1, y_1), (x_3, y_3), (x_5, y_5)$
i.e. $f(x_1) \neq y_1, \;\; f(x_3) \neq y_3, \;\; f(x_5) \neq y_5$

Polynomial Interpolation

To simplify matters, curve interpolation will always involve **polynomials**:

Proposition

(Polynomial Interpolation)

Given n points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ s.t. all x-coordinates are ***distinct****. Then, there exists a* ***unique*** *$(n-1)$-degree interpolating polynomial*

$$p(x) = c_0 + c_1x + c_2x^2 + c_3x^3 + \cdots + c_{n-1}x^{n-1}$$

where scalars $c_0, c_1, c_2, \ldots, c_{n-1} \in \mathbb{R}$ are to be determined such that

$$p(x_1) = y_1, \;\; p(x_2) = y_2, \;\; \ldots, \;\; p(x_n) = y_n$$

For instance, there's a unique quadratic $p(x) = c_0 + c_1x + c_2x^2$ that contains the **three** points $(-1, 4), (0, -2), (3, 5)$.

There's a unique cubic $p(x) = c_0 + c_1x + c_2x^2 + c_3x^3$ that contains **four** points, etc...

Polynomial Interpolation (Procedure)

So how to find the unknown coefficients $c_0, c_1, c_2, \ldots, c_{n-1}$
of an interpolating polynomial $p(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{n-1}x^{n-1}$?

Proposition

(Polynomial Interpolation Procedure)

*GIVEN: Points $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ s.t. all x-coordinates are **distinct**.*
TASK: Find unique interpolating polynomial

$$p(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3 + \cdots + c_{n-1}x^{n-1}$$

(1) Setup linear system $A\mathbf{x} = \mathbf{b}$ using $p(x_1) = y_1,\ p(x_2) = y_2,\ \ldots,\ p(x_n) = y_n$:

$$\begin{array}{ccccccccccccc} c_0 & + & c_1x_1 & + & c_2x_1^2 & + & c_3x_1^3 & + & \cdots & + & c_{n-1}x_1^{n-1} & = & y_1 \\ c_0 & + & c_1x_2 & + & c_2x_2^2 & + & c_3x_2^3 & + & \cdots & + & c_{n-1}x_2^{n-1} & = & y_2 \\ & \vdots & & \vdots & & \vdots & & \vdots & \ddots & & & \vdots & \\ c_0 & + & c_1x_n & + & c_2x_n^2 & + & c_3x_n^3 & + & \cdots & + & c_{n-1}x_n^{n-1} & = & y_n \end{array}$$

This is a $n \times n$ square linear system with unknowns $c_0, c_1, c_2, \ldots, c_{n-1}$.

(2) Solve linear system using Gauss-Jordan Elimination as usual.

Differential Curve Interpolation (Definition)

Sometimes, besides a curve having to contain certain points, its **derivatives** (such as **slope** & **concavity**) at some points must be particular values:

Definition

(Differential Curve Interpolation)

Let function $f \in C^{n-1}$. Then:
The curve $y = f(x)$ **interpolates** a point (x_0, y_0) **in the differential sense** if:

$$\begin{array}{rcl} f(x_0) & = & y_0 \\ f'(x_0) & = & \alpha_1 \\ f''(x_0) & = & \alpha_2 \\ \vdots & & \vdots \\ f^{(n-1)}(x_0) & = & \alpha_{n-1} \end{array}$$

where scalars $\alpha_1, \alpha_2, \ldots, \alpha_{n-1} \in \mathbb{R}$.

i.e. The curve contains a single point but also must satisfy prescribed derivative values at that point.

NOTATION: $f \in C^n$ means function f is n-times continuously differentiable.

Differential Curve Interpolation (Procedure)

Proposition

(Differential Curve Interpolation Procedure)

GIVEN: Point (x_0, y_0) & function $f \in C^{n-1}$ s.t. $f(x) = c_1 f_1(x) + \cdots + c_n f_n(x)$
TASK: Find coefficients $c_1, \ldots, c_n$ s.t. f satisfies the following conditions:

$$f(x_0) = y_0,\ f'(x_0) = \alpha_1,\ f''(x_0) = \alpha_2,\ f'''(x_0) = \alpha_3,\ \ldots,\ f^{(n-1)}(x_0) = \alpha_{n-1}$$

(1) Setup $n \times n$ linear system where each equation satisfies a condition.
(2) Solve linear system using Gauss-Jordan Elimination as usual.

($\star$) For simplicity, functions $f_1(x), f_2(x), \ldots, f_n(x)$ can only be:

- *Polynomials:* $1, x, x^2, x^3, \ldots$
- *Exponentials:* $e^x, e^{-x}, e^{2x}, e^{-2x}, \ldots$
- *Sines/Cosines:* $\sin x, \cos x, \sin 2x, \cos 2x, \ldots$
- *Products of these:* $xe^x, x \sin 2x, x^3 \cos x, e^{2x} \sin 3x, \ldots$

NOTATION: $f \in C^n$ means function f is n-times continuously differentiable.

Advanced Interpolation

The interpolation methods discussed so far are the most elementary methods. However, there are more advanced (robust) methods that won't be considered in this course:

- Lagrange Interpolation
- Hermite Interpolation
- Spline Interpolation
- Interpolation using barycentric coordinates
- Interpolation using surfaces (instead of curves)

Many of these more advanced interpolation methods are encountered in **Numerical Analysis**. (MATH 4310)

Fin.