**ECE 4960**

**Spring 2017**

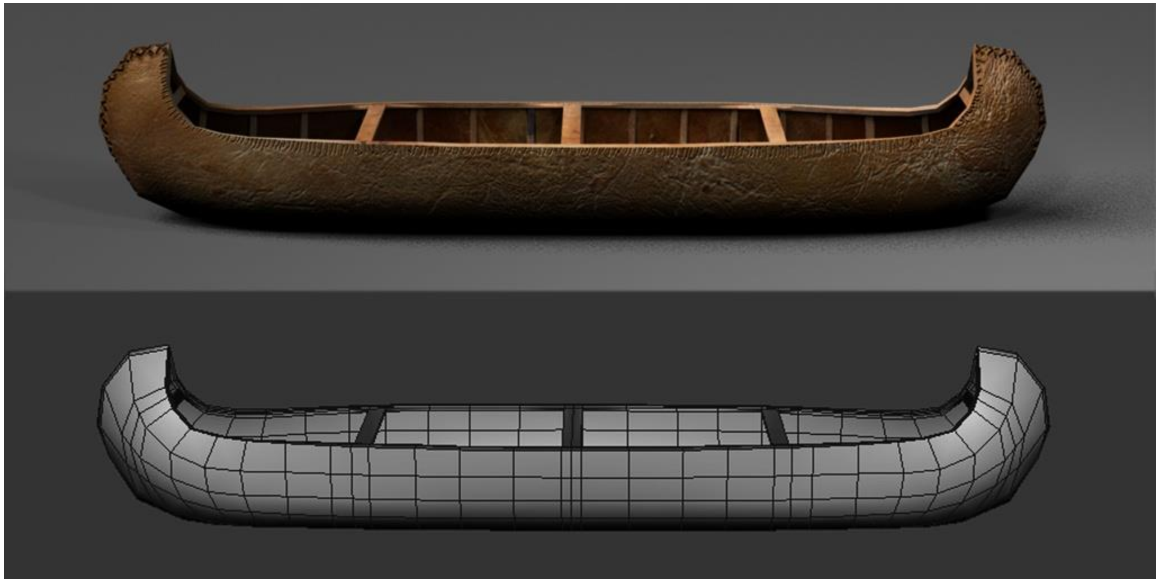
# Lecture 17

## Nonlinear Equations and Optimization:
## Geometry Optimization: Spline Fitting

**Edwin C. Kan**
School of Electrical and Computer Engineering
Cornell University

# Problems from the Ancient Time

- Vessels and canoes need fixed sizes (so that they can fit with other parts and easy for carpenters), but the shape should be smooth, as a box can easily experience much larger water resistance or drag.
- Polynomial fitting
- Fixed-point spline fitting
- NURBS (non-uniform rational basis spline)
- Cubic least-square spline fitting



# Polynomial Fitting in 2D

- If we know $d+1$ points in the $x$-$y$ plane, i.e., ,

$$(x_k, y_k) \text{ for } k = 0, 1, 2, \ldots, d.$$

$$\text{where } x_i \neq x_j \text{ if } i \neq j.$$

- There is a polynomial function of order $d$ that can pass through these $d+1$ points:

$$p(x) = p_0 + p_1 x + p_2 x^2 + \ldots + p_d x^d$$

$$p(x_k) = y_k \quad \Rightarrow \quad [V]\vec{p} = \vec{y}$$

$$[V] = \begin{bmatrix} 1 & x_0 & \ldots & x_0^d \\ 1 & x_1 & \ldots & x_1^d \\ \ldots & \ldots & \ldots & \ldots \\ 1 & x_d & \ldots & x_d^d \end{bmatrix} \qquad \vec{p} = \begin{bmatrix} p_0 \\ p_1 \\ \ldots \\ p_d \end{bmatrix} \qquad \vec{y} = \begin{bmatrix} y_0 \\ y_1 \\ \ldots \\ y_d \end{bmatrix}$$

Vandermonte matrix
Rank($V$) = $d + 1$

# Lagrange Polynomials

- There is always a solution to the polynomial fitting, which can be expressed through the Lagrange polynomials:

$$l_k(x) = \frac{\prod_{j \neq k} (x - x_j)}{\prod_{\substack{j \neq k \\ j = 0,1,\ldots d}} (x_k - x_j)}$$

- For example, for $d = 2$, $x_0 = 0$, $x_1 = 2$, $x_2 = 3$, we have the first two Lagrange polynomials as:

$$l_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} = \frac{1}{6}(x^2 - 5x + 6); \qquad l_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} = x^2 - 3x$$

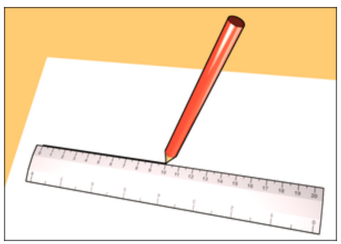
- We have $l_k(x_j) = \delta_{jk} = \begin{cases} 1 & j = k \\ 0 & j \neq k \end{cases}$. Finally, $p(x) = \sum_{k=0}^{d} y_k l_k(x)$

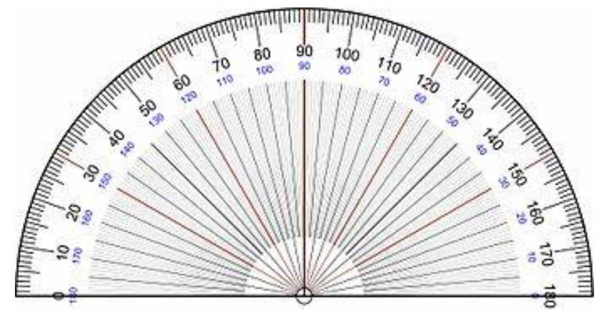
# Vandermonte Matrix

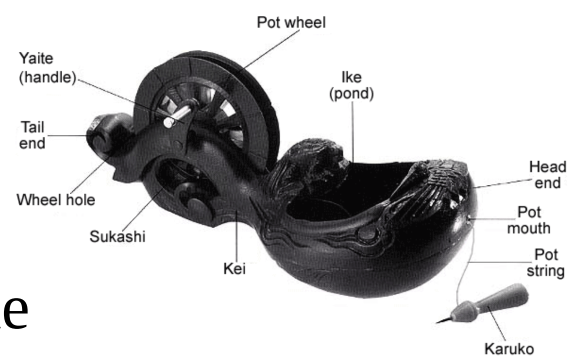
- As *[V]p = y* has a solution, the vandermonte matrix *V* is nondegenerate.
- Polynomial fitting is known to have many oscillations, especially when *d* is large. It is seldom used directly
- But polynomial fitting provides a useful intuitive understanding for the much more important spline fitting.

# Geometrical Fitting

- A straight line between two points: ruler; ink line.
- Circle: compass
- Angles: protractor
- Smooth curve with control or anchor points?






Ink line

French
curve

# Spline Curves

- Classic carpentry tools that use the minimization of elastic energy of the "spline" to get the smooth curve that passes through the "anchor" or duck or nail or knot points with **continuous 1st and 2nd** derivatives.
- Important for vehicular curves and computer graphics

# Piecewise Polynomials

- Let a spline curve $S$ which has $d+1$ fixed points it has to pass through between $[a, b]$:

$$a = x_0 < x_1 < x_2 < \ldots < x_{d-1} < x_d = b$$

- There are $d$ intervals where we need $d$ piecewise polynomials to describe $S$:

$$\begin{aligned} S(x) &= p_1(x); \quad x_0 \le x < x_1 \\ S(x) &= p_2(x); \quad x_1 \le x < x_2 \\ &\ldots \\ S(x) &= p_d(x); \quad x_{d-1} \le x < x_d \end{aligned}$$

- If we hope that $S(x)$ is smooth to the $n$-$1$ derivative, we will consturct a $n$-degree spline curves:

$$p_i^{(j)}(x_i) = p_{i+1}^{(j)}(x_i) \quad \text{for } i = 1,.., d-1; \quad j = 0, \ldots, n-1$$

# Basis Splines

- Basis spline (b-spline): curves constrained by anchors and continuous derivatives
- The most common b-spline has $n$ = 3 and is **C-2 continuous** everywhere (including anchors).
- **Normalize** cubic $p_i(x)$ to $q_i(x)$ within the $i$-th interval

$$q_i(x) \equiv (1-t)y_{i-1} + ty_i + t(1-t)\left(a_i(1-t) + b_i(t)\right)$$

$$t = \frac{x - x_{i-1}}{x_i - x_{i-1}}; \qquad a_i = k_{i-1}(x_i - x_{i-1}) - (y_i - y_{i-1}); \quad b_i = -k_i(x_i - x_{i-1}) + (y_i - y_{i-1});$$

- Automatically satisfied (but we do not know the value of $k_i$):

$$q_i(x_i) = q_{i+1}(x_i) = y_i; \qquad q_{i-1}(x_{i-1}) = q_i(x_{i-1}) = y_{i-1};$$
$$q_i'(x_i) = q_{i+1}'(x_i) = k_i; \qquad q_{i-1}'(x_{i-1}) = q_i'(x_{i-1}) = k_{i-1};$$

# Determine Unknown 1st Derivative to Guarantee Continuous 2nd Derivative

$$q_i''(x_i) = q_{i+1}''(x_i) \text{ for } i = 1,\ldots d-1$$

- Only $d - 1$ conditions for $d + 1$ $k_i$'s,
- Two additional conditions needed
  - $q_i''(x_0) = q_{i+1}''(x_d) = 0$ √
  - Give $k_0$ and $k_d$ directly

$$\frac{k_{i-1}+2k_i}{x_i - x_{i-1}} + \frac{2k_i + k_{i+1}}{x_{i+1}-x_i} = 3\left(\frac{y_i - y_{i-1}}{(x_i - x_{i-1})^2} + \frac{y_{i+1}-y_i}{(x_{i+1}-x_i)^2}\right)$$

$$q''(x_0) = 2\cdot\frac{3(y_1 - y_0) - (k_1 + 2k_0)(x_1 - x_0)}{(x_1 - x_0)^2} = 0$$

$$q''(x_d) = -2\cdot\frac{3(y_d - y_{d-1}) - (k_d + 2k_{d-1})(x_d - x_{d-1})}{(x_d - x_{d-1})^2} = 0$$

⇨

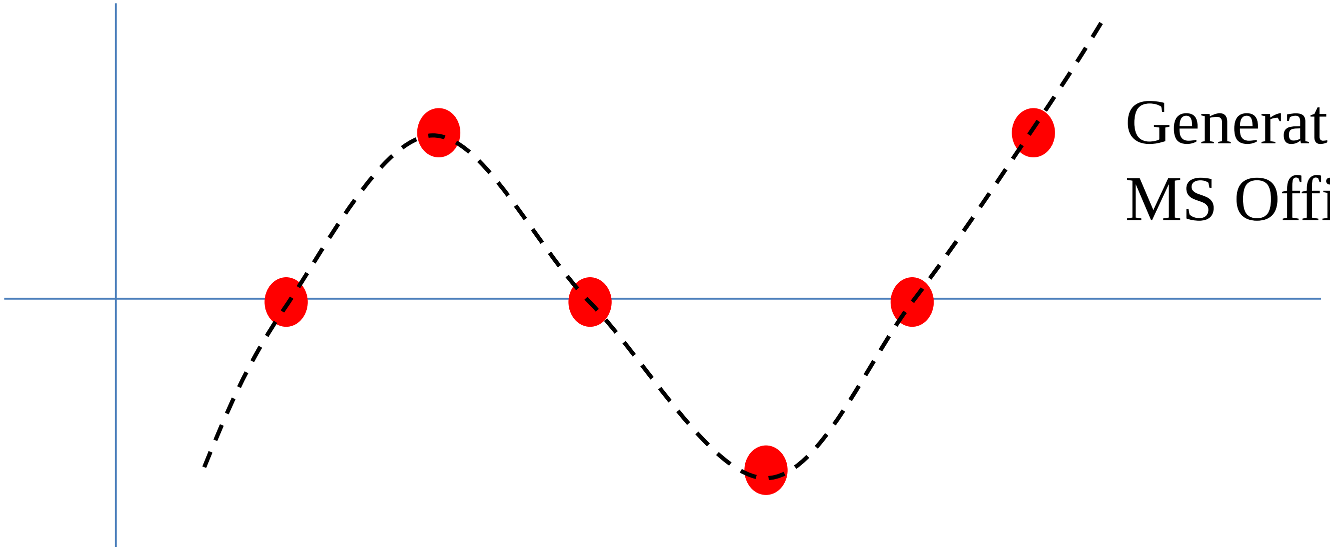
$$\begin{bmatrix} a_{11} & a_{12} & \ldots & 0 \\ a_{21} & a_{22} & \ldots & 0 \\ 0 & \ldots & \ldots & \ldots \\ 0 & \ldots & a_{d-1d} & a_{dd} \end{bmatrix}\begin{bmatrix} k_0 \\ k_1 \\ \ldots \\ k_d \end{bmatrix} = \begin{bmatrix} \ldots \\ \ldots \\ \ldots \\ \ldots \end{bmatrix}$$

# Hacker Practice

Construct a B-Spline curve with the following anchor points

| $x$ | 1.0 | 2.0 | 3.0 | 4.0 | 5.0 | 6.0 |
|---|---|---|---|---|---|---|
| $y$ | 0.0 | 1.0 | 0.0 | –1.0 | 0.0 | 1.0 |

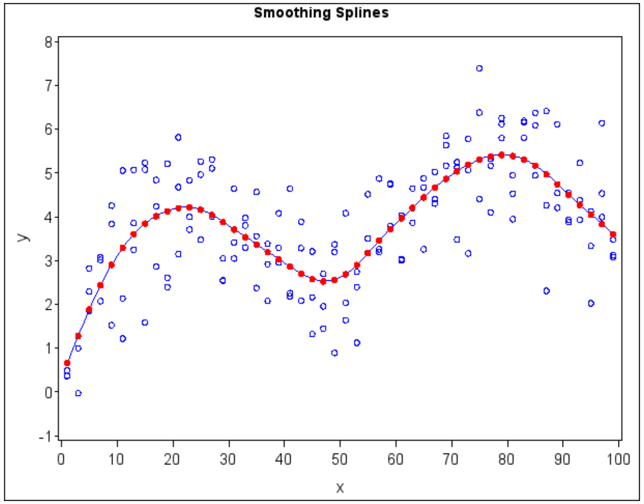
Define $S$ as piecewise b-spline functions, and if you can, plot it out!

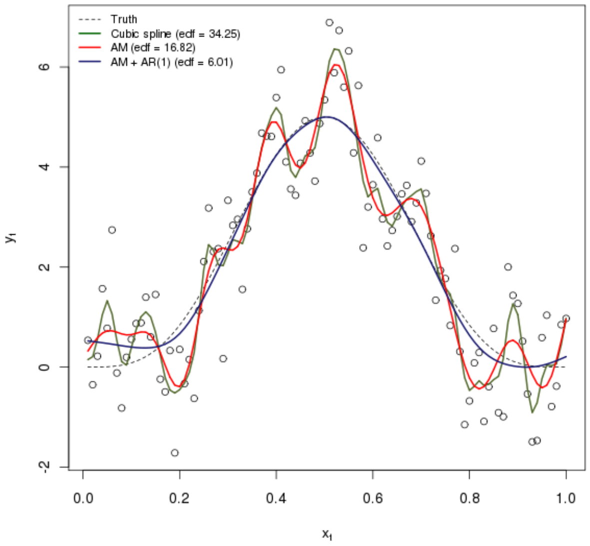


Generated in MS Office!

# Smoothing Splines and Control Points

- Least-square fitting of spline curves, or smoothing splines, with the control points instead of anchor points.
- The control points are where the piecewise cubic functions are separated, although *S(x)* is C-2 continuous everywhere.
- We may have a lot of experimental points from an image, and we ask the questions what the coefficients of $q_i(x)$ are to formulate least-square fitting with the experimental points.
- The control points are often assigned by some known features of *S(x)*, or adaptively by the trends in the experimental points.



# NURBS (Non-Uniform Rational B-Spline)

- 2D extenstion to 3D, i.e., instead of finding a spline curve, we are finding the spline surface that passes through anchor points, or the smoothing spline surface that gives the least-square fitting.
- Combining with all spline features (C-2 continuity, anchor or control points, etc.), we can now see how NURBS (non-uniform rational b-spline) are used universally in mechanical designs, animation and image processing