CS 357: Numerical Methods

Hermite Cubic Interpolation
Fourier Basis
Radial Basis Functions

Eric Shaffer

Some slides adapted from:

Scientific Computing: An Introductory Survey, 2nd ed., McGraw-Hill, 2002. By Michael T. Heath

Hermite Cubic Interpolation

→ interpolate derivatives as well as values of functions

Piecewise polynomial

n nodes or knots

$\alpha_1 + \alpha_2 x + \alpha_3 x^2 + \alpha_4 x^3$

$4(n-1)$ parameters

$2(n-1)$ equations to interpolate function

$n-2$ equations from matching $f'(x)$

→ $3n-4$

$\boxed{n \text{ free parameters}}$

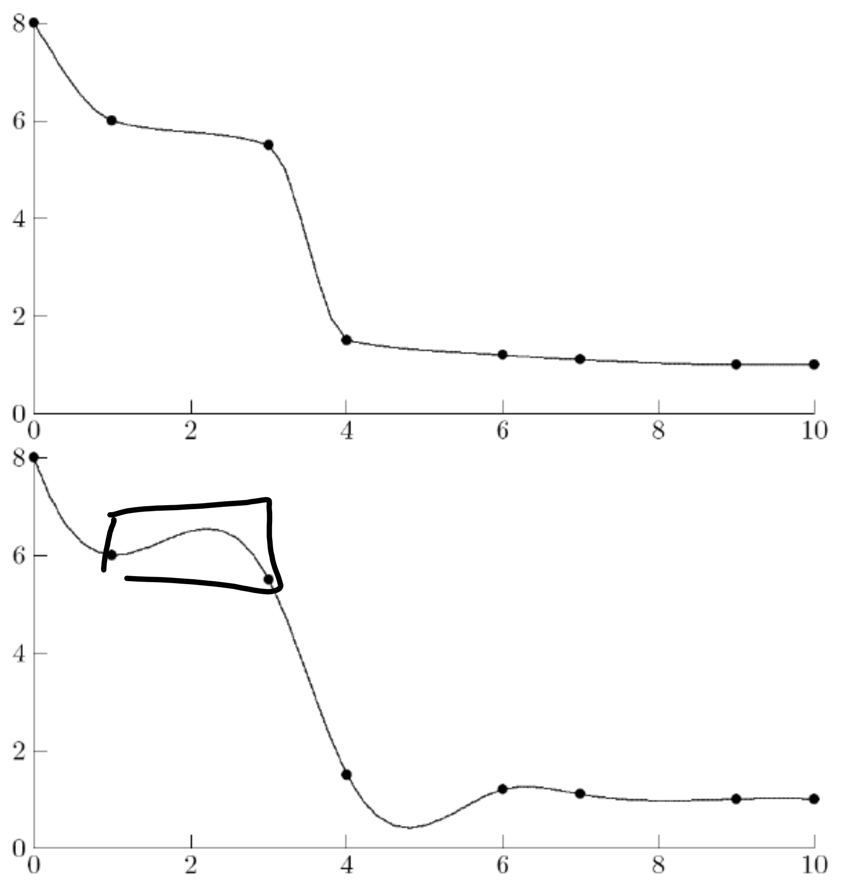
Hermite Cubic Interpolation

How is it different from splines?

Spline has $n-1$ matching derivatives @ knots for degree n polynomial

Cubic

Hermite versus Splines



Hermite

Spline

Review: Interpolation Error

n = degree of poly

If we use a cubic polynomial interpolant on an interval of length *h*

How much must reduce h to achieve an error bound 1/10,000 less than the original?

$$|f(x) - \tilde{f}(x)| < C h^{n+1}$$

function

interpolant

interval length

constant

any point in interval

$$C\left(\frac{h}{10}\right)^4 = \frac{1}{10000} C h^4$$

Review: Interpolation Error

Assume the function f being interpolated is smooth

We use a polynomial of degree n as an interpolant

The length ***h*** of the interpolation interval is "sufficiently small"
Then we have:

$$\left|f(x) - \tilde{f}(x)\right| \leq Ch^{n+1}$$

Error depends on the interval h (the "step-size")

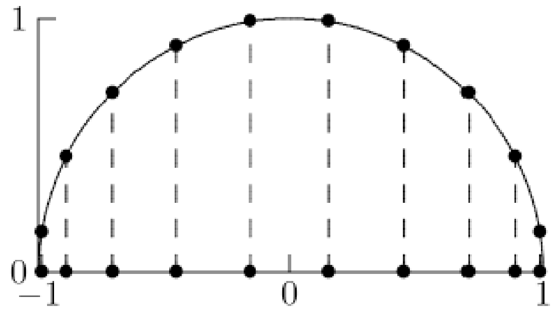
Choosing Nodes for Interpolation

- Best if nodes cluster towards the ends of the interval
- On [-1,1] the ***Chebyshev nodes*** perform best
  - $\underbrace{x_k = \cos(\frac{2k-1}{2n}\pi) \text{ for } k=1,\ldots,n}$
- What about over an arbitrary interval? (a, b)

$$x_k = \underbrace{\tfrac{1}{2}(a+b) + \tfrac{1}{2}(b-a)}\;\underbrace{\cos\left(\frac{2k-1}{2n}\pi\right)}_{\text{affine trans.}}$$

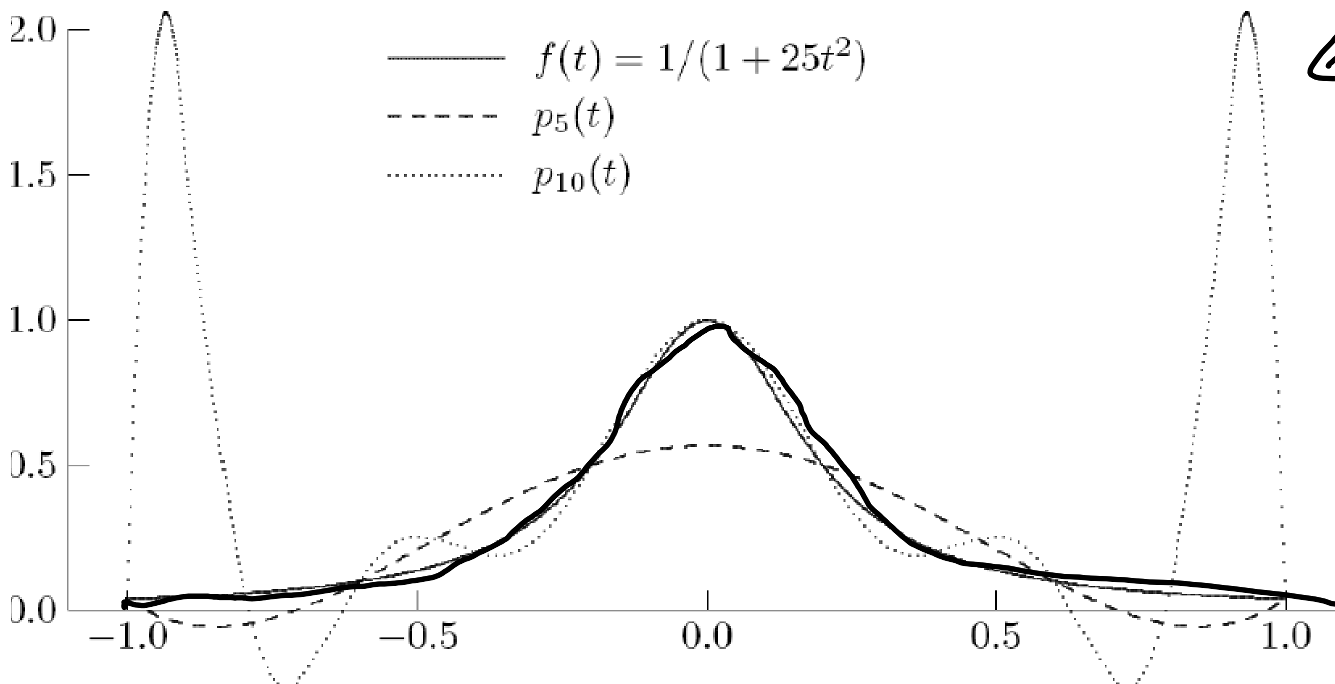
Choosing Nodes for Interpolation

Chebyshev points are abscissas of points equally spaced around unit circle in $\mathbb{R}^2$



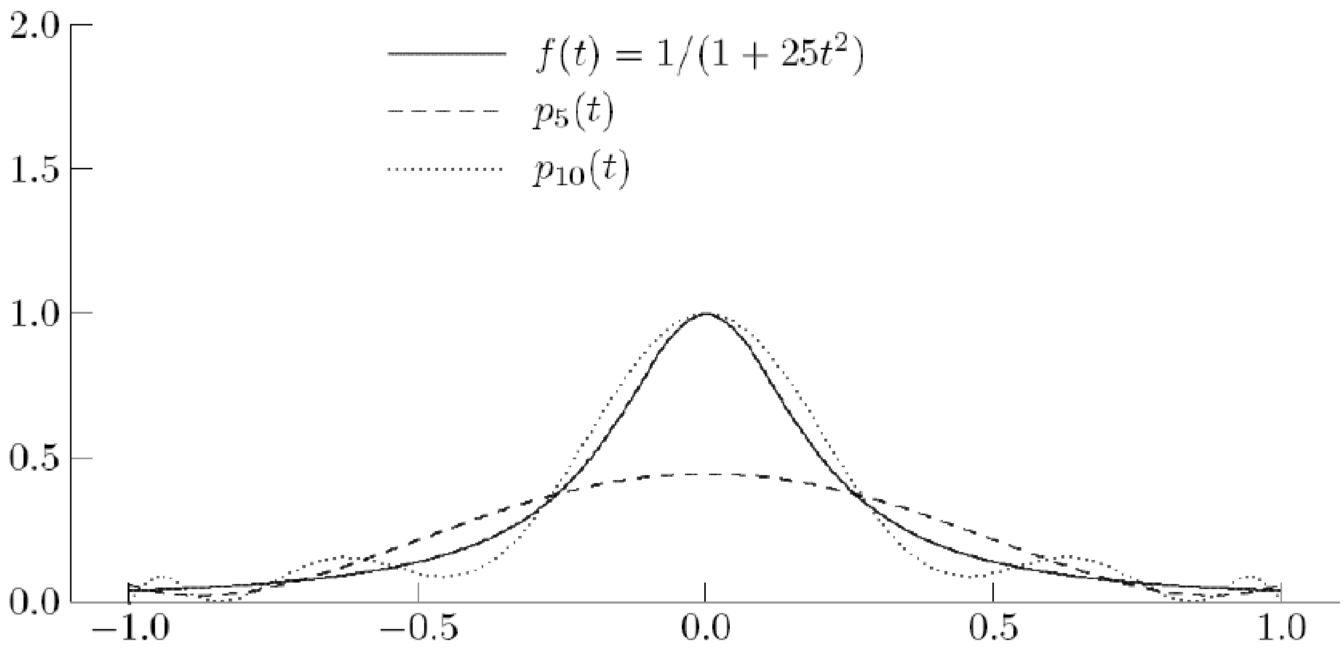
Choosing Nodes for Interpolation

Polynomial interpolants of Runge's function at *equally spaced* points *do not* converge



Choosing Nodes for Interpolation

- Polynomial interpolants of Runge's function at *Chebyshev* points *do* converge



$a + bx + cx^2$ $\qquad\qquad$ $d + ex + fx^2$

x_0 $\qquad\qquad$ x_1 $\qquad\qquad$ x_2

Piecewise Polynomial Interpolation

Review in-class exercise

$$\begin{bmatrix} 1 & x_0 & x_0^2 & 0 & 0 & 0 \\ 1 & x_1 & x_1^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_1 & x_1^2 \\ 0 & 0 & 0 & 1 & x_2 & x_2^2 \\ 0 & 1 & 2x_1 & 0 & -1 & -2x_1 \\ 0 & 1 & 2x_0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ f(x_1) \\ f(x_2) \\ 0 \\ 0 \end{bmatrix}$$

Example

Trigonometric Interpolation

- Trigonometric interpolation uses a sum of sines and cosines as an interpolant
- In modeling periodic or cyclic phenomena, sines and cosines are more appropriate functions than polynomials or piecewise polynomials
- Representation as a linear combination of sines and cosines decomposes continuous function or discrete data into components of various frequencies
- Representation in frequency space may enable more efficient manipulations than in original time or space

Fourier Basis

$= 0$ implies we use odd # nodes

- sin(0x), cos(0x),sin(1x),cos(1x),sin(2x),(cos2x),…,sin(Kx),cos(Kx)

- A trigonometric polynomial of degree K has the form

$$p(x) = a_0 + \sum_{k=1}^{K} a_k \cos(kx) + \sum_{k=1}^{K} b_k \sin(kx)$$

Fourier Basis

- A trigonometric polynomial of degree K has the form

$$p(x) = \underline{\underline{a_0}} + \sum_{k=1}^{K} \underline{\underline{a_k}}\cos(kx) + \sum_{k=1}^{K} \underline{\underline{b_k}}\sin(kx)$$

- We wish to find what values?

so p(x) passes through n points

Fourier Basis

- The trigonometric polynomial is periodic with period 2π
- The n points should be distributed as:
  $\underbrace{0 \le x_0 < x_1 < \ldots < x_{n-1} < 2\pi}$

$V x = b$ ← function eval'd at nodes

Vandermonde Matrix with Fourier Basis

Using 5 nodes $0, 2\pi(\frac{1}{5}), 2\pi(\frac{2}{5}), 2\pi(\frac{3}{5}), 2\pi(\frac{4}{5})$

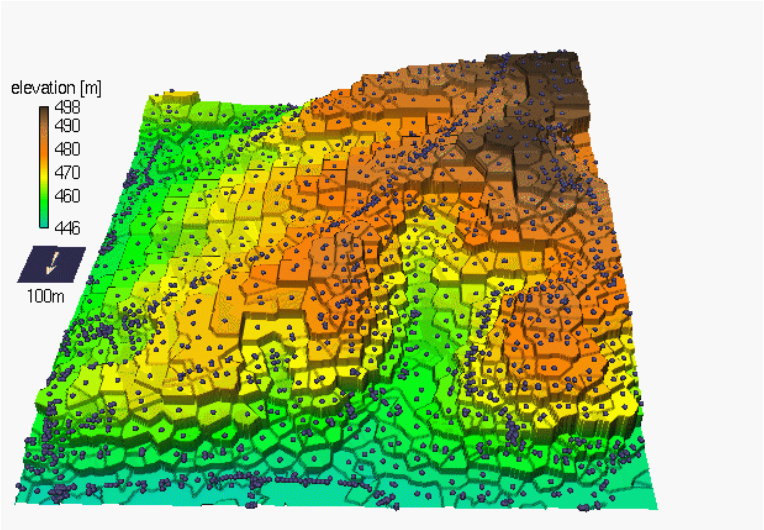
$$\begin{bmatrix} \cos(0\cdot 0) & \sin(1\cdot 0) & \cos(1\cdot 0) & \cdots \\ \cos(0\cdot \frac{2\pi}{5}) & \sin(1\cdot \frac{2\pi}{5}) & \vdots & \\ \cos(0\cdot 2\pi(\frac{2}{5})) & \sin(1\cdot 2\pi\frac{2}{5}) & \vdots & \\ \cos(0\cdot 2\pi(\frac{3}{5})) & \vdots & \vdots & \\ \cos(0\cdot 2\pi(\frac{4}{5})) & \vdots & \vdots & \end{bmatrix} \begin{bmatrix} a_0 \\ b_1 \\ a_1 \\ b_2 \\ a_2 \end{bmatrix} = \begin{bmatrix} y_0 \\ \\ \vdots \\ \\ y_4 \end{bmatrix}$$

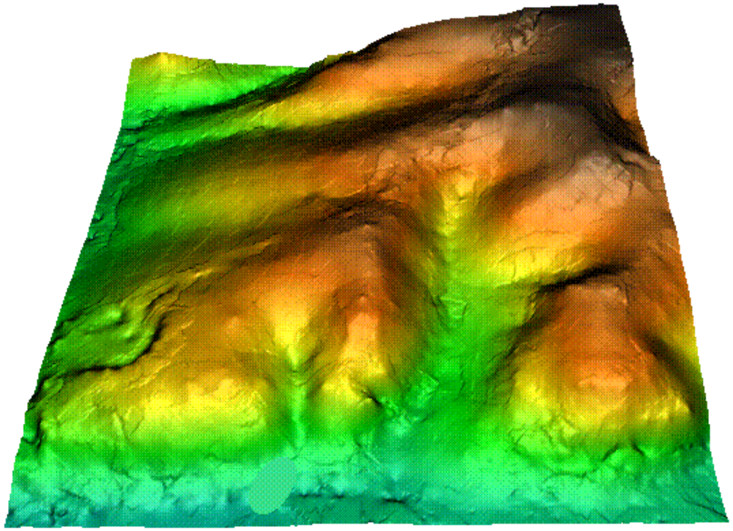
$y_i = f(2\pi \frac{i}{5})$

Vandermonde Matrix with Fourier Basis

Radial Basis Functions

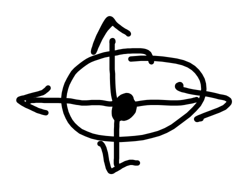
- What about higher-dimensional data?
- Example: imagine you have height samples on a surface





Radial Basis Functions

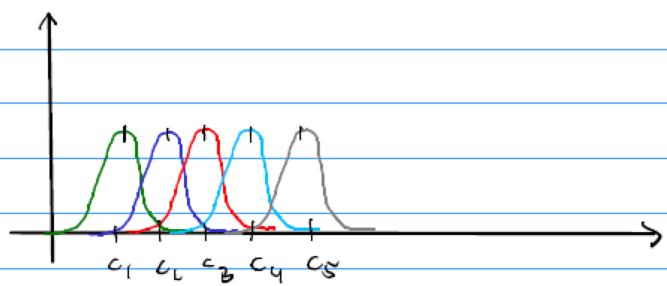
- There are interpolation methods for unstructured data
  - i.e. for data sampled in any pattern
- Radial Basis Functions allow us to interpolate without meshing
- Functions are defined in terms of distance from a point
  - Hence the the term *radial*



Radial Basis Functions

Idea: Use multiple copies of the same function, centered at

a number of locations on the real line.



- Function of distance from a center
- As distance increases, functions goes to 0

Radial Basis Functions (RBFs)

- Any function dependent on distance from a center is *radial*
- We can compute an approximate function as a weighted sum..

$$\underbrace{\phi(x,p)}_{\text{RBF}} = \phi(\|x-p\|)$$

x = center
p = some point

$$\boxed{f(x) \approx \sum_{i=1}^{N} w_i \phi(x, p_i)}$$

- Some popular RBFs include

$$\boxed{\phi(r) = e^{-\lambda r^2}}$$ Gaussian

$$\phi(r) = \frac{1}{1+r^2}$$ Inverse distance

r is a distance

RBFs

$$f(p_j) = \sum_{i=1}^{N} w_i \, \phi(p_j, p_i)$$

$$\begin{bmatrix} \phi(p_1, p_1) & \cdots & \phi(p_1, p_N) \\ \vdots & & \vdots \\ \phi(p_N, p_1) & \cdots & \phi(p_N, p_N) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} f(p_1) \\ \vdots \\ f(p_N) \end{bmatrix}$$

Finding coeffiencients

$$f(p_j) = \sum_{i=1}^{N} w_i \phi(p_j, p_i)$$

$$Aw = p$$

$$A = \begin{bmatrix} \phi(p_1, p_1) & \dots & \phi(p_1, p_N) \\ \dots & \dots & \dots \\ \phi(p_N, p_1) & \dots & \phi(p_N, p_N) \end{bmatrix}$$

$$w = \begin{bmatrix} w_1 \\ \dots \\ w_N \end{bmatrix}$$

$$p = \begin{bmatrix} f(p_1) \\ \dots \\ f(p_N) \end{bmatrix}$$

RBFs

- Easy to construct
- Generalizes easily to higher dimensions
- Tricky to stabilize
- Quality of interpolation depends on:
  - Choice of basis
    - How quickly does it decay?
  - Location of nodes (i.e. scattered data points)
    - Interpolant decays with distance from nodes

Example