MTH 622
Interpolation by Polynomials

Ripu Singla
Y6387
April 22, 2010

Outline

- The problem
- Lagrange Polynomials
- Runge's Phenomenon
- Chebyshev Polynomials
- Divided Differences
- Forward Differences
- Spline Interpolation
- Hermite Interpolation
- Applications
- Polynomial vs Fractal Interpolation

The problem

- The problem is to interpolate *f(x)* by $p(x) \in P_n$ in C[a,b].
- P_n is the set of all polynomials of degree n or less.

  for example,

  $$P_3 = \{ax^3 + bx^2 + cx + d \mid a, b, c, d \text{ real}\}$$

- Let $x_0, x_1, \ldots, x_n$ be $n+1$ distinct points on interval [a,b]. Then *p(x)* is said to interpolate *f(x)* at each of these points if

  $$p(x_j) = f(x_j) \quad 0 \le j \le n$$

- In other words, given some data points, find the polynomial of least possible degree which goes exactly through these points.

Theorem

- Let $\{x_0, x_1, \ldots, x_n\}$ be n+1 distinct point in [a,b]. Let $\{y_0, y_1, \ldots, y_n\}$ be any set of real numbers then there exist a unique $p(x) \in P_n$ such that

$$p(x_j) = y_j \quad 0 \leq j \leq n$$

- For each j, $0 \leq j \leq n$, let $l_j(x)$ be the nth degree polynomial defined by

$$l_j(x) = \prod_{\substack{i=0 \\ i \neq j}}^{n} \frac{x - x_i}{x_j - x_i}$$

- These are called *Cardinal Functions*.

$$l_i(x_j) = \delta_{ij}$$

Lagrange Form

$$p(x) = \sum_{j=0}^{n} y_j l_j(x)$$

- if $f(x)$ is a function such that $f(x_i)=y_i$ then

$$p(x) = \sum_{j=0}^{n} f(x_j) l_j(x)$$

- Uniqueness

suppose $p(x)$ and $q(x)$ are two such polynomials.

Define $r(x) = p(x) - q(x)$ then $r(x) \in P_n$ and

$$r(x_j) = p(x_j) - q(x_j) = 0 \; for \; 0 \le j \le n$$

By fundamental theorem of algebra $r(x) \equiv 0$.

Example

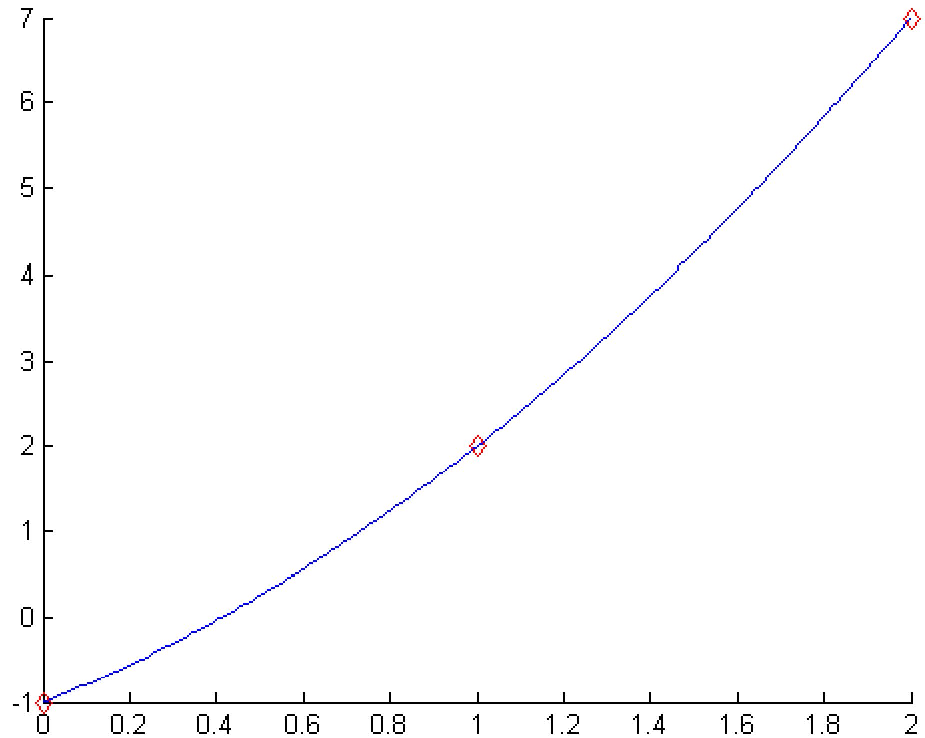
- $x_0 = 0, x_1 = 1, x_2 = 2$ *and* $y_0 = -1, y_1 = 2, y_2 = 7$

$$l_0(x) = \frac{(x-1)(x-2)}{2}$$

$$l_1(x) = -x(x-2)$$

$$l_2(x) = \frac{x(x-1)}{2}$$

$$p(x) = -l_0(x) + 2l_1(x) + 7l_2(x) = x^2 + 2x - 1$$



7
6
5
4
3
2
1
0
-1
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2

Example

- $x_0 = 2$, $x_1 = 2.5$, $x_2 = 4$ and $f(x) = \dfrac{1}{x}$

$$l_0(x) = (x - 6.5)x + 10$$

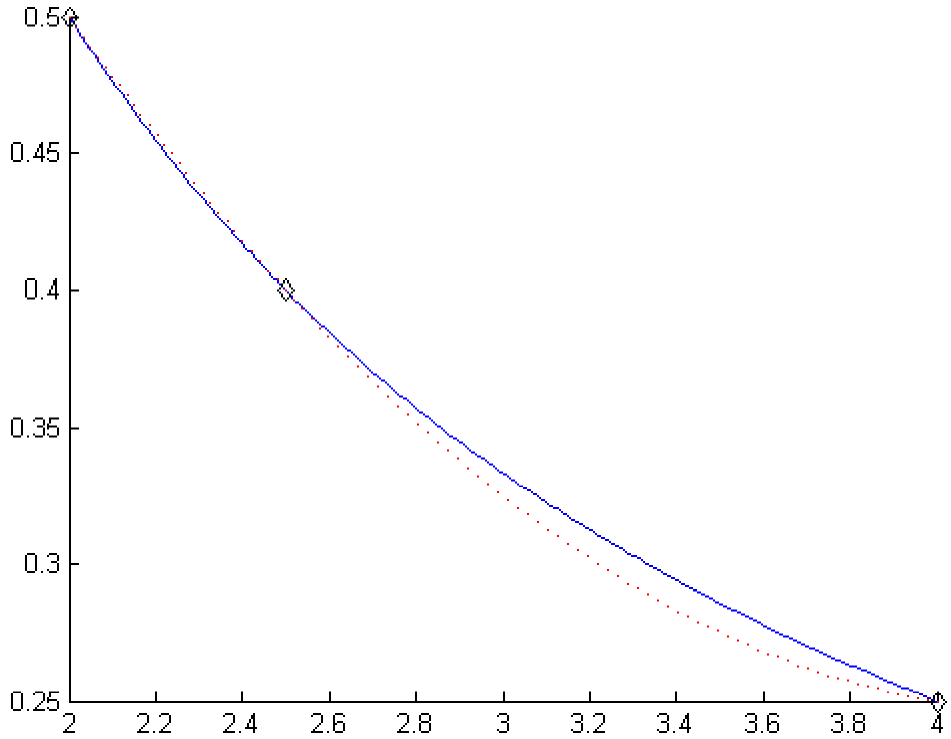
$$l_1(x) = \frac{(-4x + 24)x - 32}{3}$$

$$l_2(x) = \frac{(x - 4.5)x + 5}{3}$$

$f(2) = 0.5$, $f(2.5) = 0.4$, $f(4) = 0.25$

$$p(x) = \sum_{j=0}^{2} f(x_j) l_j(x) = 0.05x^2 - 0.425x + 1.15$$

$f(x) = 1/x$



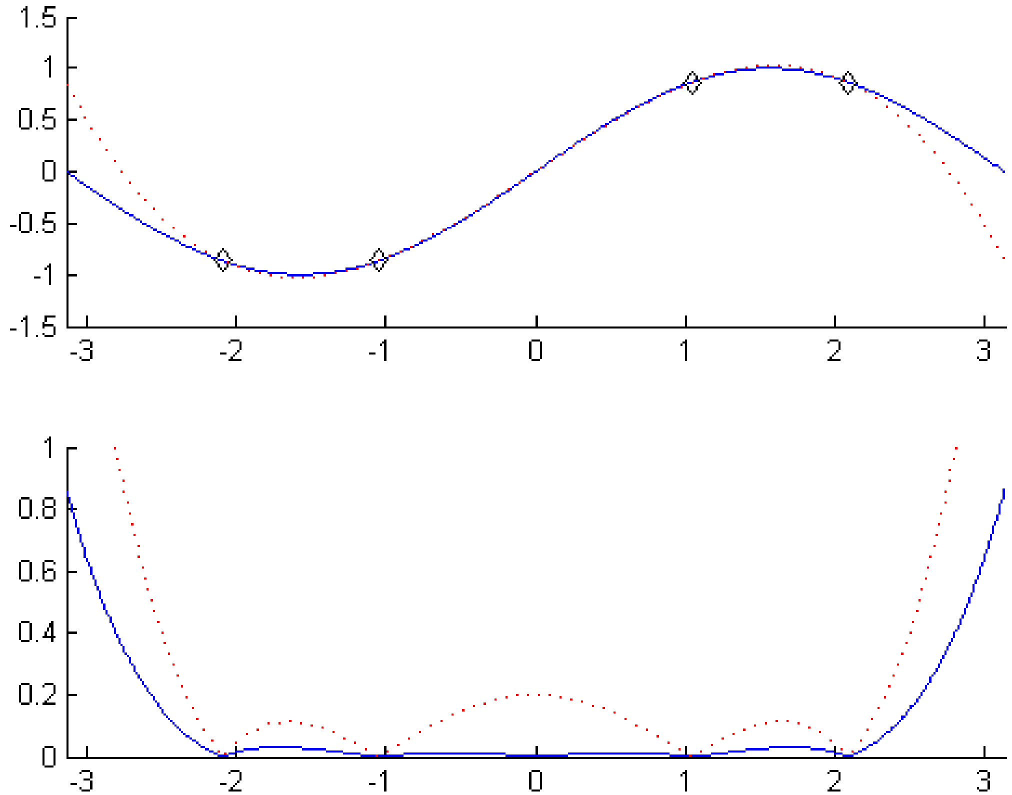
Theorem

- let $f \in C^{n+1}$[a,b] and p be the polynomial of degree at most n that interpolates f at $n+1$ points $x_0, x_1, \ldots, x_n$ in [a,b]. Then to each $x \in$ [a,b], there corresponds a point $\varphi \in (a,b)$ such that

$$f(x) - p(x) = \frac{f^{(n+1)}(\varphi(x))}{(n+1)!} \prod_{i=0}^{n} (x - x_i)$$

for some $\varphi \in (a,b)$.

$f(x) = sin(x)$



Runge's Phenomenon

- Runge's phenomenon is a problem that occurs when using polynomial interpolation with polynomials of high degree .

$$f(x)-p(x)=\frac{f^{(n+1)}(\varphi(x))}{(n+1)!}\prod_{i=0}^{n}(x-x_i)$$

- Runge's function: $f(x)=\dfrac{1}{1+25x^2}$

$$|f'(1)|\approx 0.07 \qquad |f''(1)|\approx 0.2105$$

- Equidistant points should not be taken for the class of functions which have a pole in the nhd of interpolation interval.

Chebyshev Polynomials

- Chebyshev polynomials obey a recursion relation and are orthogonal polynomials with respect to a weight function over the interval -1 to 1.

$$T_0(x) = 1 \quad T_1(x) = x$$

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \qquad n \geq 1$$

- $\dfrac{T_n(x)}{2^{n-1}}$ is a monic polynomial.

- <u>Theorem:</u>

  For $x \in$ [-1,1], the Chebyshev polynomials have the expression

$$T_n(x) = \cos(n \cos^{-1}(x)) \qquad n \geq 0$$

How to minimize error?

- Only term in error formula, we have control over is

$$\prod_{i=0}^{n}(x-x_i)$$

- It suggests that we should carefully choose the nodes x_i's.
- Choose x_i so as

$$\max_{|x|\le 1}\left|\prod_{i=0}^{n}(x-x_i)\right| \text{ is minimized}$$

- Suppose we let x_i be the zeros of the polynomial

$$\frac{1}{2^n}T_{n+1}(x)$$

Cont….

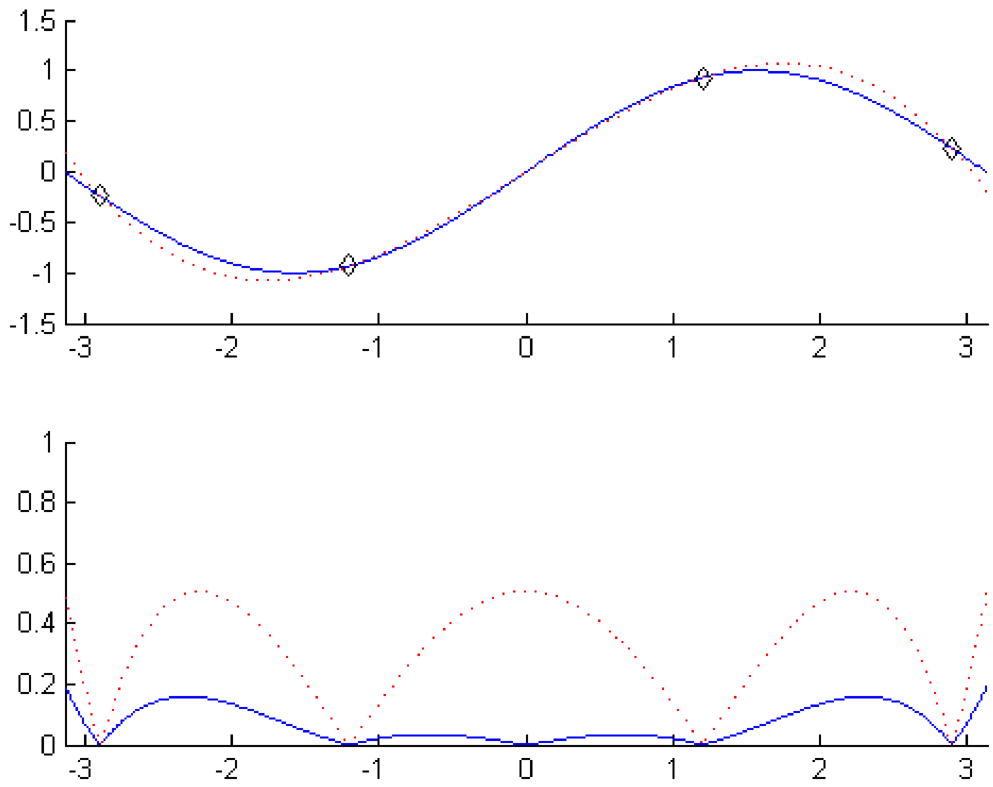
$$\frac{1}{2^n} T_{n+1}(x) = \frac{1}{2^n}\cos((n+1)\cos^{-1} x) = 0$$

$$x_i = \cos\left(\frac{(i+\frac{1}{2})\pi}{n+1}\right) \qquad i = 0,1,2,\ldots.,n$$

Then $\displaystyle \max_{|x|\le 1} \left|\prod_{i=0}^{n}(x-x_i)\right| = \frac{1}{2^n}$

Hence $\displaystyle |f(x)-p(x)| \le \frac{1}{(n+1)!}\frac{1}{2^n}\max|f^{(n+1)}(\varphi(x))|$

$f(x) = sin(x)$



Divided Differences

- Used to generate the polynomials successively.
- Newton Form

$$p(x) = a_0 + \sum_{k=1}^{n} a_k (x - x_0)\ldots(x - x_{k-1})$$

- n^{th} divided difference is given by

$$f[x_0, x_1, \ldots, x_n] = \frac{f[x_0, \ldots, x_{n-1}] - f[x_1, \ldots, x_n]}{x_0 - x_n}$$

- First few divided differences are given by

$$f[x_0] = f(x_0) \qquad f[x_0, x_1] = \frac{f_0 - f_1}{x_0 - x_1}$$

Cont…

$$f[x_0, x_1, x_2] = \frac{f[x_0, x_1] - f[x_1, x_2]}{x_0 - x_2}$$

- With the help of divided differences, we can write *p(x)*

$$p(x) = f[x_0] + \sum_{k=1}^{n} f[x_0, x_1, ..., x_k](x - x_0)...(x - x_{k-1})$$

Example

- $x_0 = 0, x_1 = 1, x_2 = 2$ *and* $y_0 = -1, y_1 = 2, y_2 = 7$

0	-1		
		3	
1	2		1
		5	
2	7		

$$p(x) = -1 + 3(x-0) + 1(x-0)(x-1)$$
$$= x^2 + 2x - 1$$

Forward Difference

- Very often data points given are equally spaced. In this case

$$a = x_0, x_1, \ldots, x_n = b$$

$$x_i = x_0 + ih, \quad h = \frac{b-a}{n} \quad 0 \le i \le n$$

- The first order forward difference is defined as

$$\triangle f_i = f_{i+1} - f_i$$

- The kth order forward difference is defined as

$$\triangle^k f_i = \triangle^{k-1}\left(\triangle f_i\right) = \triangle^{k-1} f_{i+1} - \triangle^{k-1} f_i$$

Cont…

- We can express forward differences in terms of divided differences.

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{\triangle f_0}{h}$$

$$\begin{aligned}
\triangle^2 f_0 &= \triangle\left(\triangle f_0\right) = \triangle f_1 - \triangle f_0 \\
&= hf[x_1, x_2] - hf[x_0, x_1] \\
&= h.2h\left\{\frac{f[x_1, x_2] - f[x_0, x_1]}{2h}\right\} \\
&= 2h^2\left\{\frac{f[x_1, x_2] - f[x_0, x_1]}{(x_2 - x_0)}\right\} \\
&= 2h^2 f[x_0, x_1, x_2]
\end{aligned}$$

Cont…

- Similarly, $\Delta^k f_i = k!h^k f[x_i, x_{i+1}, \dots, x_{i+k}]$

$$f[x_i, x_{i+1}, \dots, x_{i+k}] = \frac{\Delta^k f_i}{k!h^k}$$

$$f[x_0, x_1, \dots, x_k] = \frac{\Delta^k f_0}{k!h^k}$$

- Then $p(x) = f[x_0] + \sum_{k=1}^{n} \frac{\Delta^k f_0}{k!h^k}(x - x_0)\dots(x - x_{k-1})$
- If we put $x=x_0+sh$, then

$$p(x) = \sum_{k=0}^{n} \binom{s}{k} \Delta^k f_0$$

Spline Interpolation

- In the previous interpolation techniques, if we need more accuracy, we should take more data points but then the degree of polynomial increases.
- An alternative is to use Spline interpolation: use lots of points but low degree polynomial for each segment and then patch them up
- We will consider two cases:
  - Piecewise linear case
  - Cubic Splines

Piecewise Linear

- Consider a subdivision of interval [a,b] as

$$a = x_0 < x_1 < \ldots < x_n = b$$

$$\triangle x_i = x_{i+1} - x_i$$

$$|\triangle| = \max_{1 \le i \le n-1} \triangle x_i$$

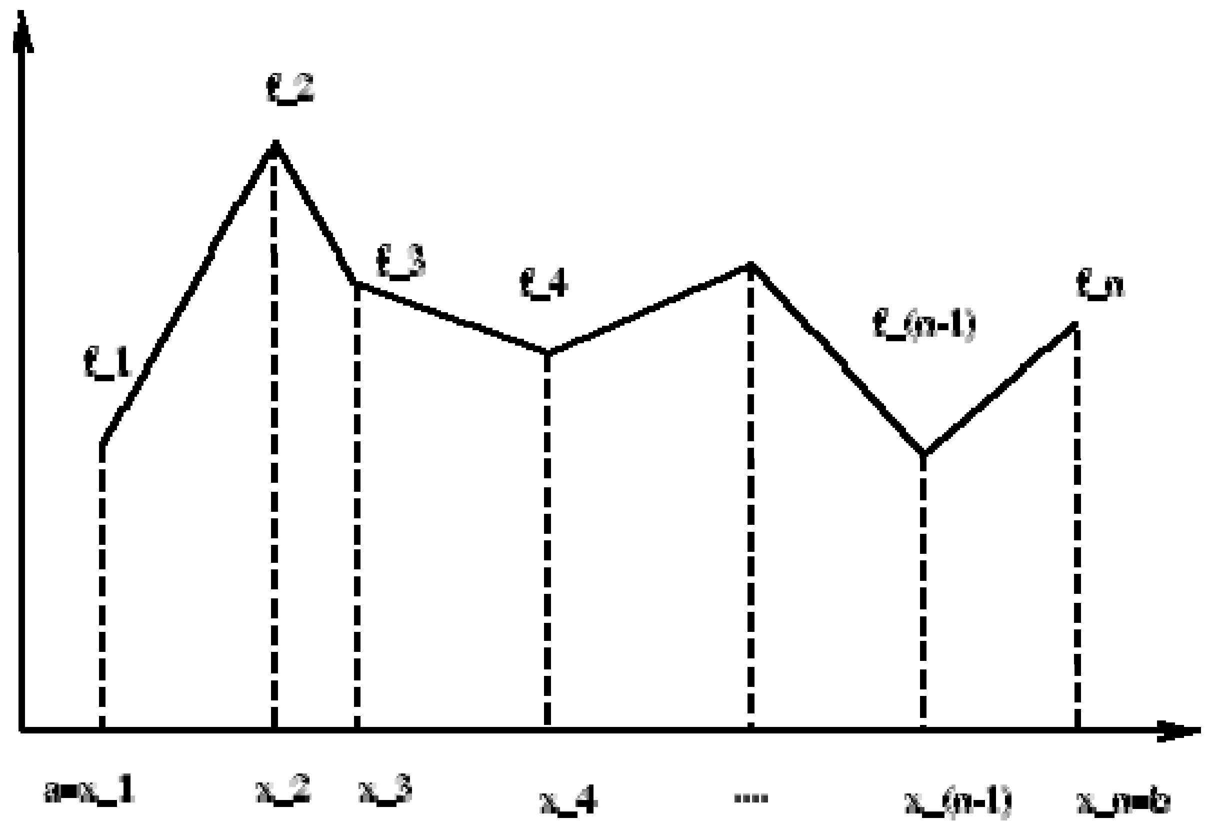
- Then use a linear polynomial on each interval $[x_i, x_{i+1}]$

$$s_i(f, x) = f_i + (x - x_i)\frac{f_{i+1} - f_i}{x_{i+1} - x_i}$$

- <u>Error:</u>

$$|f(x) - s_1(f, x)| \;\le\; \frac{1}{8}|\triangle|^2 \max|f''|$$

Cont…



Cubic Splines

- Cubic splines are lowest order polynomial endowed with inflection points.
- Avoids the Runge's phenomenon.
- Interpolate between any two points using cubics.

<u>Requirements:</u>

- The interpolating property: $s(x_i) = f(x_i)$
- The join up property: $s_{i-1}(x_i) = s_i(x_i)$
- Twice continuous differentiable property:

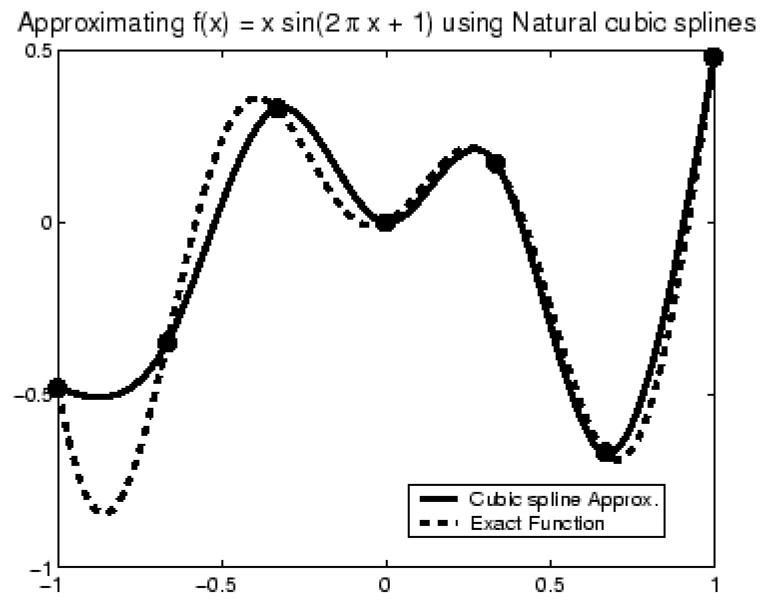
$$s'_{i-1}(x_i) = s'_i(x_i) \text{ and } s''_{i-1}(x_i) = s''_i(x_i)$$

Cont…

- For any two points $[x_i, x_{i+1}]$, $s_i(x)$ is a cubic polynomial given by

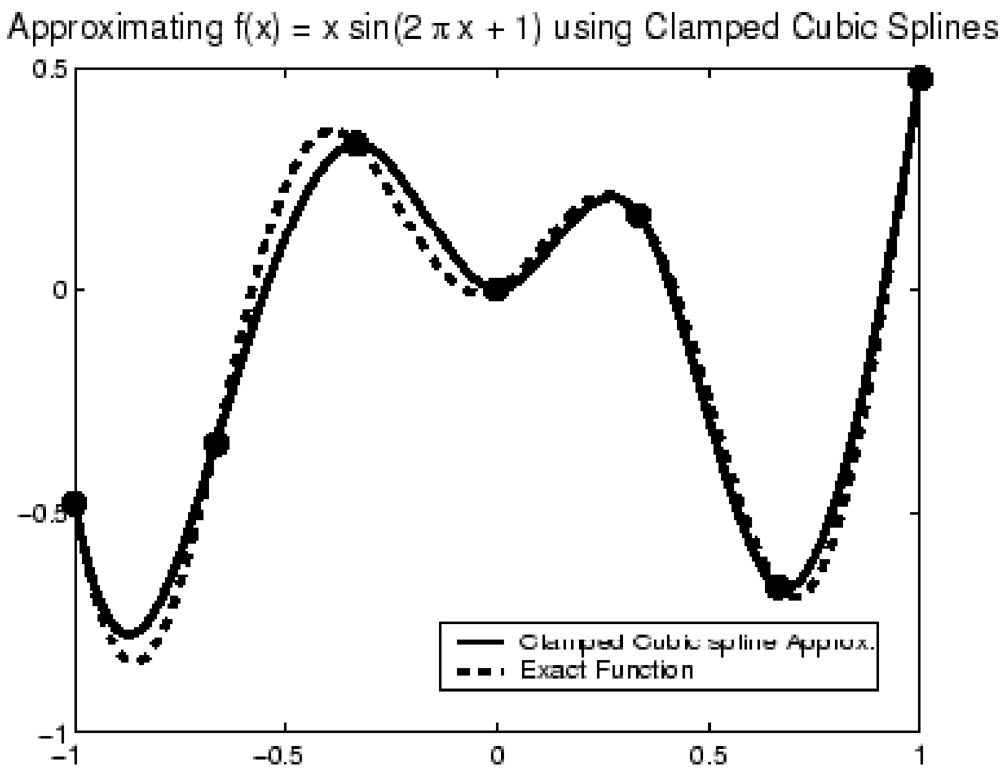
$$s_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$$

for x in $[x_i, x_{i+1}]$.



Reference: http://www.physics.arizona.edu/~restrepo/475A/Notes/sourcea/node35.html

Cont…



Reference: http://www.physics.arizona.edu/~restrepo/475A/Notes/sourcea/node35.html

Hermite Interpolation

- Hermite polynomial matches data points in both value and first derivative.
- $(x_0, y'_0), (x_1, y'_1), \dots, (x_n, y'_n)$ must be given in addition to n data points $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$.
- Resulting polynomial is of degree at most $2n+1$.

$$h(x) = \sum_{j=0}^{n} f(x_j) A_j(x) + \sum_{j=0}^{n} f'(x_j) B_j(x)$$

$$A_j(x) = \left(1 - 2(x - x_j) l'_j(x_j)\right) l_j^2(x)$$

$$B_j(x) = (x - x_j) l_j^2(x)$$

$$l_j(x) = \prod_{\substack{i=0 \\ i \neq j}}^{n} \frac{x - x_i}{x_j - x_i}$$

- <u>Hermite-Birkhoff Interpolation</u>: Uses higher order derivatives.

Divided Difference Method

- Suppose *n+1* data points are given consisting of $\left(x_i, f\left(x_i\right), f'\left(x_i\right)\right)$.
- We create a new dataset $z_0, z_1, \ldots, z_{2n+1}$ s.t $z_{2i} = z_{2i+1} = x_i$
- Now create divided differences table for points

  $z_0, z_1, \ldots, z_{2n+1}$.
- Define $f\left[x_i, x_i\right] = f'\left(x_i\right)$
- And for higher order derivatives, define

$$f\left[x_i, x_i, x_i\right] = \frac{f^{(2)}\left(x_i\right)}{2!}$$

$$f\left[x_i, x_i, x_i, x_i\right] = \frac{f^{(3)}\left(x_i\right)}{3!} \text{ and so on.}$$

Example

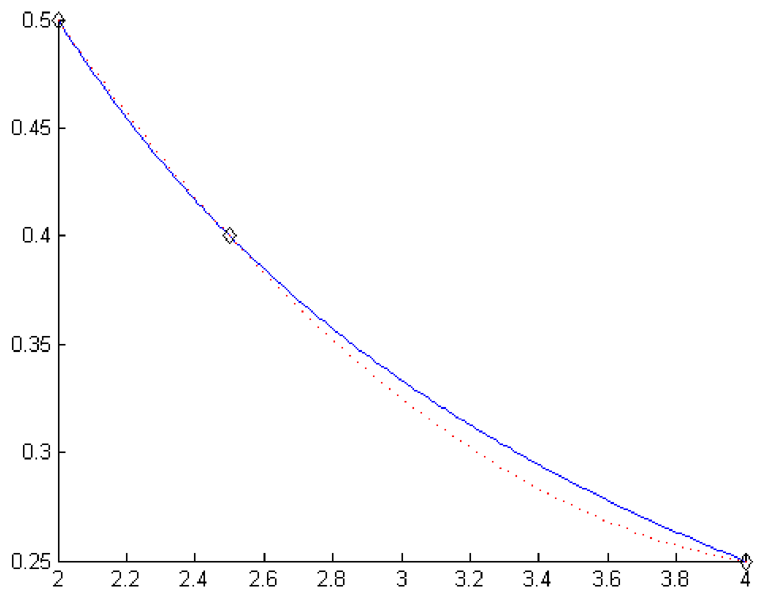
- Suppose we want to interpolate $f(x) = \dfrac{1}{x}$

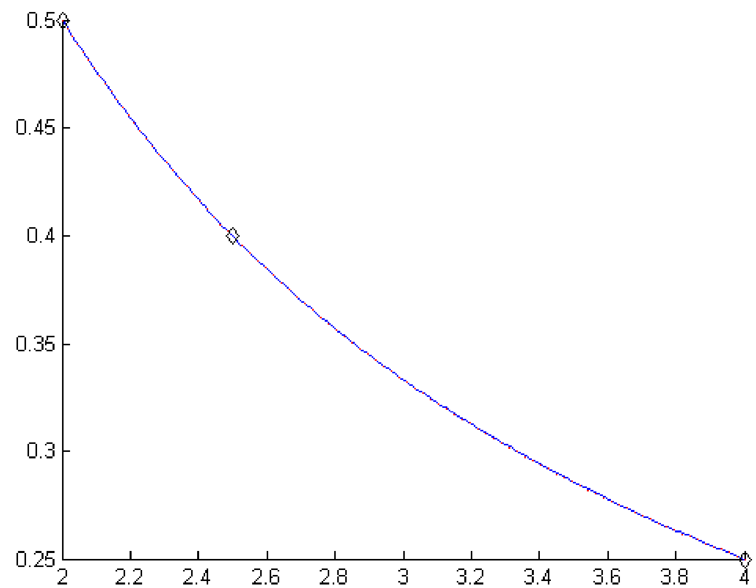
x	$f(x)$	$f'(x)$
2	0.5	-0.25
2.5	0.4	-0.16
4	0.25	-0.0625

2	0.5					
		-0.25				
2	0.5		0.1			
		-0.2		-0.04		
2.5	0.4		0.08		0.01	
		-0.16		-0.02		-0.0025
2.5	0.4		0.04		0.005	
		-0.1		-0.01		
4	0.25		0.025			
		-0.0625				
4	0.25					

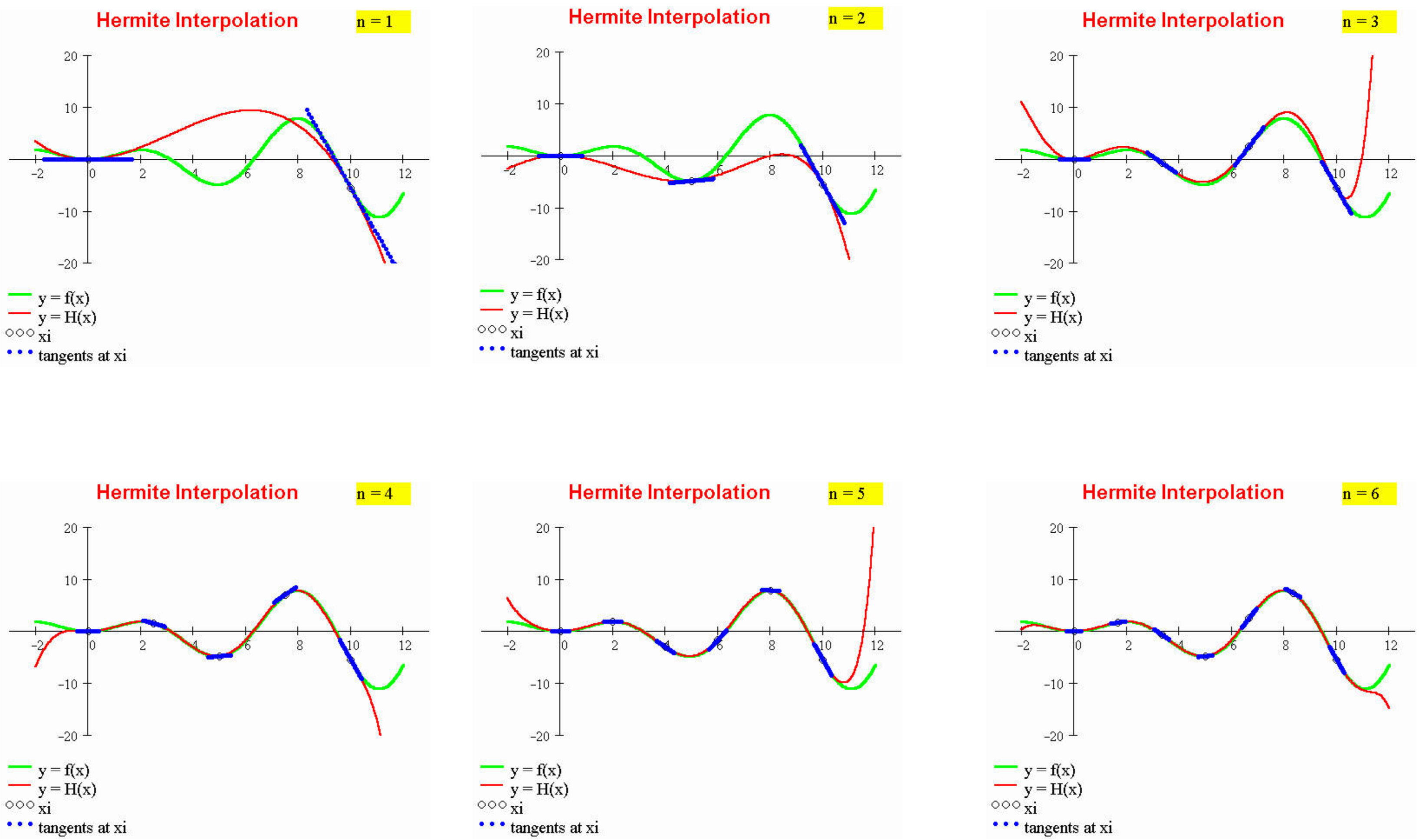
$$p(x) = 0.5 - 0.25(x-2) + 0.1(x-2)^2 - 0.04(x-2.5)(x-2)^2 + 0.01(x-2.5)^2(x-2)^2 - 0.0025(x-4)(x-2.5)^2(x-2)^2$$

$f(x)=1/x$


0.5
0.45
0.4
0.35
0.3
0.25
2
2.2
2.4
2.6
2.8
3
3.2
3.4
3.6
3.8
4


0.5
0.45
0.4
0.35
0.3
0.25
2
2.2
2.4
2.6
2.8
3
3.2
3.4
3.6
3.8
4

Example



Reference: http://www.math.odu.edu/~bogacki/videnum/hermit_1.htm

Applications

- Polynomials can be used to approximate more complicated curves, for example, the shapes of letters in typography, given a few points.
- Evaluation of the natural logarithm and trigonometric functions.
- Polynomial interpolation also forms the basis for algorithms in numerical quadrature and numerical ordinary differential equations.

<u>Reference</u>: http://en.wikipedia.org/wiki/Polynomial_interpolation

Polynomial vs Fractal Interpolation

- Polynomial functions can be expressed by simple formulas.
- FIF provide a new means for fitting experimental data.
- It does not suffice to make a polynomial fit to wild experimental data of strahle for the temperature in a jet exhaust as a function of time. However fractal interpolation can be used to fit such data.
- Graphics systems founded on traditional geometry are effective for making pictures of man-made objects such as bricks, wheels, roads, buildings etc. since these objects were designed using Euclidean geometry.

<u>Reference</u>: Fractals Everywhere, M. Barnsley ,chapter 6

Cont…

- However it is desirable for graphics systems to be able to deal with a wider range of problems. Fractal Interpolation functions can be used to approximate image components such as the profiles of mountain ranges, the tops of clouds and horizons over forests.
- Fractal interpolation, rather than treating the image component as arising from a random assemblage of objects such as individual mountains, cloudlets or tree tops, one model the image component as an interrelated single system.

<u>Reference</u>: Fractals Everywhere, M. Barnsley ,chapter 6

Thank You.