MATH 3795
Lecture 15. Polynomial Interpolation. Splines.

Dmitriy Leykekhman

Fall 2008

Goals

- Approximation Properties of Interpolating Polynomials.
- Interpolation at Chebyshev Points.
- Spline Interpolation.
- Some MATLAB's interpolation tools.

Approximation Properties of Interpolating Polynomials.

One motivation for the investigation of interpolation by polynomials is the attempt to use interpolating polynomials to approximate unknown function values from a discrete set of given function values.

How well does the interpolating polynomial $P(f|x_1, \ldots, x_n)$ approximate the function f?

Approximation Properties of Interpolating Polynomials.

One motivation for the investigation of interpolation by polynomials is the attempt to use interpolating polynomials to approximate unknown function values from a discrete set of given function values.

How well does the interpolating polynomial $P(f|x_1, \ldots, x_n)$ approximate the function f?

Theorem

Let $x_1, x_2, \ldots, x_n$ be unequal points. If f is n times differentiable, then for each $\bar{x}$ there exists $\xi(\bar{x})$ in the smallest interval containing the points $x_1, x_2, \ldots, x_n, \bar{x}$ such that

$$f(\bar{x}) - P(f|x_1, x_2, \ldots, x_n)(\bar{x}) = \frac{1}{n!}\omega(\bar{x}) f^{(n)}(\xi(\bar{x}))$$

where $\omega(x) = \prod_{j=1}^{n}(x - x_j)$.

Approximation Properties of Interpolating Polynomials.

Corollary (Convergence of Interpolating Polynomials)

If $P(f|x_1, \ldots, x_n)$ is the polynomial of degree less or equal to $n-1$ that interpolates f at the n distinct nodes $x_1, x_2, \ldots, x_n$ belonging to the interval $[a, b]$ and if the nth derivative $f^{(n)}$ of f is continuous on $[a, b]$, then

$$\max_{x \in [a,b]} |f(x) - P(f|x_1, \ldots, x_n)(x)| \leq \frac{1}{n!} \max_{x \in [a,b]} |f^{(n)}(x)| \max_{x \in [a,b]} \left| \prod_{i=1}^{n} (x - x_i) \right|.$$

Approximation Properties of Interpolating Polynomials.

Corollary (Convergence of Interpolating Polynomials)

If $P(f|x_1, \ldots, x_n)$ is the polynomial of degree less or equal to $n-1$ that interpolates f at the n distinct nodes $x_1, x_2, \ldots, x_n$ belonging to the interval $[a,b]$ and if the nth derivative $f^{(n)}$ of f is continuous on $[a,b]$, then

$$\max_{x\in[a,b]} |f(x)-P(f|x_1,\ldots,x_n)(x)| \le \frac{1}{n!} \max_{x\in[a,b]} |f^{(n)}(x)| \max_{x\in[a,b]} \left|\prod_{i=1}^{n}(x-x_i)\right|.$$

The size of the error between the interpolating polynomial $P(f|x_1, \ldots, x_n)$ and f depends on

- the smoothness of the function $(max_{x\in[a,b]}|f^{(n)}(x)|)$ and
- he interpolation nodes $(max_{x\in[a,b]}\,|\prod_{i=1}^{n}(x-x_i)|)$.

Approximation Properties of Interpolating Polynomials.

Example

Consider the function

$$f(x) = \sin(x).$$

For $n = 0, 1, \ldots,$ it holds that

$$f^{(n)}(x) = \begin{cases} (-1)^k \sin(x), & \text{if} \quad n = 2k \\ (-1)^k \cos(x), & \text{if} \quad n = 2k+1. \end{cases}$$

Since $|f^{(n)}(x)| \leq 1$ for all x we obtain that

$$\max_{x \in [a,b]} |f(x) - P(f|x_1, \ldots, x_n)(x)| \leq \frac{1}{n!}(b-a)^n.$$

Thus, on any interval $[a, b]$ the sine function can be uniformly approximated by interpolating polynomials.

Interpolation at Equidistant Points.

- The interpolation points are $x_i = a + ih$, $i = 1, \ldots, n$, where $h = \frac{b-a}{n-1}$.
- With this choice of nodes, one can show that for arbitrary $x \in [a, b]$,

$$\left| \prod_{i=1}^{n} (x - x_i) \right| \leq \frac{1}{4} h^n (n-1)!$$

- The error between the interpolating polynomial $P(f|x_1, \ldots, x_n)$ and f is bounded by

$$\begin{aligned} &\max_{x \in [a,b]} |f(x) - P(f|x_1, \ldots, x_n)(x)| \\ &\leq \frac{1}{n!} \max_{x \in [a,b]} |f^{(n)}(x)| \max_{x \in [a,b]} \left| \prod_{i=1}^{n} (x - x_i) \right| \\ &\leq \frac{h^n}{4n} \max_{x \in [a,b]} |f^{(n)}(x)| \end{aligned}$$

provided that the nth derivative $f^{(n)}$ of f is continuous on $[a, b]$.

Interpolation at Chebyshev Points.

- Is there a choice $x_1^*, x_2^*, \dots, x_n^*$ of nodes such that

$$\max_{x \in [a,b]} \left| \prod_{i=1}^{n} (x - x_i^*) \right|$$

is minimal?

- This leads to the minmax, or Chebyshev approximation problem

$$\min_{x_1,\dots,x_n} \max_{x \in [a,b]} \left| \prod_{i=1}^{n} (x - x_i^*) \right|.$$

Interpolation at Chebyshev Points.

- The solution $x_1^*, \dots, x_n^*$ of this problem are the socalled Chebyshev points

$$x_i^* = \frac{1}{2}(a+b) + \frac{1}{2}(b-a)\cos\left(\frac{(2i-1)\pi}{2n}\right), \quad i = 1, \dots, n,$$

$$\max_{x \in [a,b]} \left| \prod_{i=1}^{n} (x - x_i^*) \right| \le 2^{1-2n}(b-a)^n.$$

Interpolation at Chebyshev Points.

- The solution $x_1^*, \dots, x_n^*$ of this problem are the socalled Chebyshev points

$$x_i^* = \frac{1}{2}(a+b) + \frac{1}{2}(b-a)\cos\left(\frac{(2i-1)\pi}{2n}\right), \quad i = 1, \dots, n,$$

$$\max_{x \in [a,b]} \left| \prod_{i=1}^{n} (x - x_i^*) \right| \leq 2^{1-2n}(b-a)^n.$$

- Error between the interpolating polynomial $P(f|x_1^*, \dots, x_n^*)$ and f:

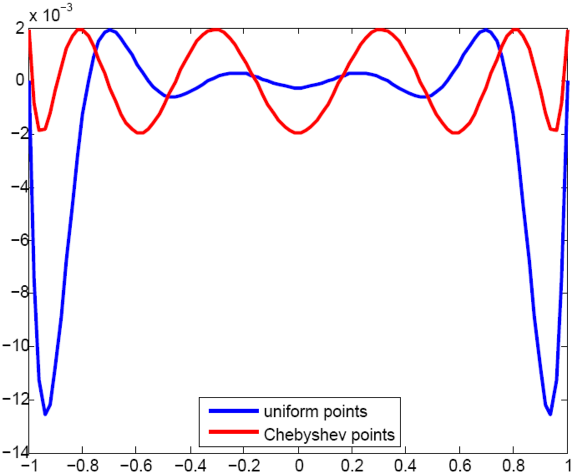
$$\max_{x \in [a,b]} |f(x) - P(f|x_1^*, \dots, x_n^*)(x)| \leq \frac{2^{1-2n}(b-a)^n}{n!} \max_{x \in [a,b]} |f^{(n)}(x)|,$$

provided that the nth derivative $f^{(n)}$ of f is continuous on $[a,b]$.

Interpolation at Chebyshev Points.

Example

The polynomial $\prod_{i=1}^{n}(x - x_i)$ with 10 equidistant points and 10 Chebychev points on $[-1, 1]$.



Polynomial Interpolation.

- Given data

$$\begin{array}{c|c|c|c} x_1 & x_2 & \cdots & x_n \\ \hline f_1 & f_2 & \cdots & f_n \end{array}$$

(think of $f_i = f(x_i)$) we want to compute a polynomial p_{n-1} of degree at most $n-1$ such that

$$p_{n-1}(x_i) = f_i, \quad i = 1, \ldots, n.$$

- If $x_i \neq x_j$ for $i \neq j$, there exists a unique interpolation polynomial.
- The larger n, the interpolation polynomial tends to become more oscillatory.
- Let $x_1, x_2, \ldots, x_n$ be unequal points. If f is n times differentiable, then for each $\bar{x}$ there exists $\xi(\bar{x})$ in the smallest interval containing the points $x_1, x_2, \ldots, x_n, \bar{x}$ such that

$$f(\bar{x}) - P(f|x_1, x_2, \ldots, x_n)(\bar{x}) = \frac{1}{n!} \left(\prod_{j=1}^{n} (\bar{x} - x_j) \right) f^{(n)}(\xi(\bar{x})).$$

Spline Interpolation.

- We do not use polynomials globally, but locally.
- Subdivide the interval $[a,b]$ such that

$$a = x_0 < x_1 < \cdots < x_n = b.$$

  Approximate the function f by a piecewise polynomial S such that
  - on each subinterval $[x_i, x_{i+1}]$ the function S is a polynomial S_i of degree k,
  - $S_i(x_i) = f(x_i)$ and $S_i(x_{i+1}) = f(x_{i+1})$, $i = 0, \dots, n-1$ (S interpolates f at $x_0, \dots, x_n$),
  - $$S_{i-1}^{(l)}(x_i) = S_i^{(l)}(x_i), \quad i = 1, \dots, n-1, \quad l = 1, \dots, k-1$$

    (the derivatives up to order $k-1$ of S are continuous at $x_1, \dots, x_{n-1}$).

  The function S is called a spline of degree k.
- We consider linear splines ($k = 1$) and cubic splines ($k = 3$).

Linear Splines.

- Let $a = x_0 < x_1 < \cdots < x_n = b$ be a partition of $[a, b]$.
- We want to approximate f by piecewise linear polynomials.
- On each subinterval $[x_i, x_{i+1}]$, $i = 0, 1, \ldots, n-1$, we consider the linear polynomials

$$S_i(x) = a_i + b_i(x - x_i).$$

- The linear spline S satisfies the following properties:
  1. $S(x) = S_i(x) = a_i + b_i(x - x_i)$, $x \in [x_i, x_{i+1}]$ for $i = 0, \ldots, n-1$,
  2. $S(x_i) = f(x_i)$ for $i = 0, \ldots, n$,
  3. $S_i(x_{i+1}) = S_{i+1}(x_{i+1})$ for $i = 0, \ldots, n-2$,
- The conditions (1-3) uniquely determine the linear functions $S_i(x) = a_i + b_i(x - x_i)$. If we consider the ith subinterval $[x_i, x_{i+1}]$, then a_i, b_i must satisfy

$$f(x_i) = S(x_i) = S_i(x_i) = a_i + b_i(x_i - x_i), \quad \text{and}$$
$$f(x_{i+1}) = S_{i+1}(x_{i+1}) = S_i(x_{i+1}) = a_i + b_i(x_{i+1} - x_i).$$

  This is a 2×2 system for the unknowns a_i, b_i. Its solution is given by

$$a_i = f(x_i), \quad b_i = (f(x_{i+1}) - f(x_i))/(x_{i+1} - x_i).$$

MATLAB's `interp1`.

MATLAB has a build-in function called **interp1** that do $1-D$ data interpolation.

Syntax:

```
yi = interp1(x,Y,xi)
yi = interp1(Y,xi)
yi = interp1(x,Y,xi,method)
yi = interp1(x,Y,xi,method,'extrap')
yi = interp1(x,Y,xi,method,extrapval)
pp = interp1(x,Y,method,'pp')
```

MATLAB's `interp1`.

```
yi = interp1(x,Y,xi)
```

interpolates to find `yi`, the values of the underlying function `Y` at the points in the vector or array `xi`. `x` must be a vector. `Y` can be a scalar, a vector, or an array of any dimension, subject to the some conditions.
To find out more, type

```
help interp1
```

MATLAB's `interp1`.

Example

Consider,

```
>> x = linspace(0,1,10);
>> y = sin(x);
```

Thus we entered 10 uniform points of the sine function on the interval $[0, 1]$. Let's say we want to approximate the value at $\pi/6$ by linear interpolation. This can be done by

```
>> interp1(x,y,pi/6)
```

and give the answer

```
ans =    0.4994
```

which is rather crude since the exact answer is $\sin(\pi/6) = 0.5$.

Cubic Splines.

- Let $a = x_0 < x_1 < \cdots < x_n = b$ be a partition of $[a, b]$.
- On each subinterval $[x_i, x_{i+1}]$, $i = 0, 1, \ldots, n-1$, we consider the cubic polynomial $S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3$.
- The cubic spline S satisfies the following properties:
  1. $S(x) = S_i(x), \quad x \in [x_i, x_{i+1}]$ for $i = 0, \ldots, n-1$,
  2. $S(x_i) = f(x_i)$ for $i = 0, \ldots, n$,
  3. $S_i(x_{i+1}) = S_{i+1}(x_{i+1})$ for $i = 0, \ldots, n-2$,
  4. $S_i'(x_{i+1}) = S_{i+1}'(x_{i+1})$ for $i = 0, \ldots, n-2$,
  5. $S_i''(x_{i+1}) = S_{i+1}''(x_{i+1})$ for $i = 0, \ldots, n-2$,
- To determine S we have to determine $4n$ parameters

$$a_i, b_i, c_i, d_i, \quad i = 0, \ldots, n-1.$$

- Equations (2-5) impose $(n+1) + (n-1) + (n-1) + (n-1) = 4n - 2$ conditions on S. Therefore we need two additional conditions on S to specify the parameters uniquely.

Cubic Splines.

- The two conditions are either

$$S''(x_0) = S''(x_n) = 0 \quad \text{(natural or free boundary)}, \ (1)$$

or

$$S'(x_0) = f'(x_0), \ S'(x_n) = f'(x_n) \quad \text{(clamped boundary)}, \ (2)$$

or

$$S^{(i)}(x_0) = S^{(i)}(x_n), \quad i = 0, 1, 2 \quad \text{(periodic spline)}. \ (3)$$

- A function S satisfying (1) is called a **natural cubic splin**e, a function S satisfying (2) is called a **clamped cubic spline**, and a function S satisfying (3) is called a **periodic cubic spline**.

Convergence of Clamped Cubic Splines.

Theorem (Convergence of Clamped Cubic Splines)

Let $f \in C^4([a,b])$ and suppose that there exists $K > 0$ such that

$$h_{max} = \max_{i=0,\dots,n-1} h_i \le K \min_{i=0,\dots,n-1} h_i,$$

where $h_i = x_{i+1} - x_i$.
If S is the clamped cubic spline, i.e. spline satisfying (1), then there exist constants C_k such that

$$\max_{x\in[a,b]} |f^{(k)}(x) - S^{(k)}(x)| \le C_k h_{max}^{4-k} \max_{x\in[a,b]} |f^{(4)}(x)|, \quad k = 0, 1, 2,$$

and

$$|f^{(3)}(x) - S^{(3)}(x)| \le C_3 h_{max} \max_{x\in[a,b]} |f^{(4)}(x)|, \quad x \in \cup_{i=0}^{n-1}(x_i, x_{i+1}).$$

MATLAB's `interp1` (cont).

The MATLAB's function `interp1` gives a choice the specify the method of interpolation.

```
yi = interp1(x,Y,xi,method)
```

interpolates using alternative methods:

```
'nearest'      Nearest neighbor interpolation

'linear'       Linear interpolation (default)

'spline'       Cubic spline interpolation

'pchip'         Piecewise cubic Hermite interpolation
```

```
yi = interp1(x,Y,xi,method)
```

MATLAB's `interp1` (cont).

Example

In the previous example

```
>> x = linspace(0,1,10);
>> y = sin(x);
```

Typing

```
>> interp1(x,y,pi/6,'spline')
```

gives

```
ans =   0.499999897030974
```

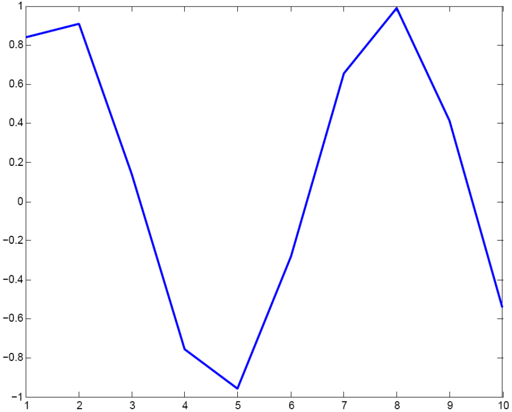
which is much closer to 0.5 then 0.4994 from the linear interpolation.

MATLAB's `interp1` (cont). Example.

Consider

```
>> x = 1:10;
>> y = sin(x);
plot(x,y)
```

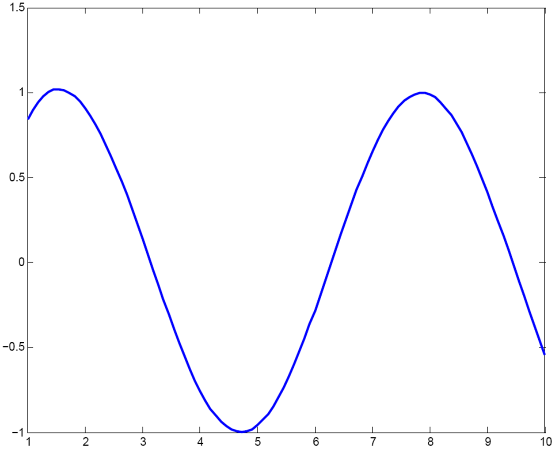
produces a graph, that looks rather rough.



MATLAB's `interp1` (cont). Example.

We can obtain a smoother graph by

```
>> xx = (1:10,100);
>> yy = interp1(x,y,'spline',xx);
plot(xx,yy)
```



Cubic Splines.

- Let g be twice continuously differentiable on $[a, b]$.
- The curvature of g at $x \in [a, b]$ is given by $g''(x)/(1 + (g'(x))^2)^{3/2}$.
- We approximate the curvature of g on $[a, b]$ by $\left(\int_a^b [g''(x)]^2 dx\right)^{1/2}$.
- Let S be a cubic spline. If g is a twice continuously differentiable function that satisfies

$$g(x_i) = f(x_i), \quad i = 0, \quad , n,$$

and

$$S''(x_0)[S'(x_0) - g'(x_0)] = 0, \quad S''(x_n)[S'(x_n) - g'(x_n)] = 0, (6)$$

then

$$\left(\int_a^b [S''(x)]^2 dx\right)^{1/2} \le \left(\int_a^b [g''(x)]^2 dx\right)^{1/2}.$$

A cubic spline is the function with smallest curvature among the twice continuously differentiable functions that interpolate f at $x_0, \ldots, x_n$ and satisfy (6).