MM7 Spline Interpolation Methods

- kl.8.15-9.10 review of MM6 and some examples
- kl.9.10 – 10.40 exercise (see notes)
- kl.10.50-11.30 MM7 lecture (I)

Introduction to Interpolation (MM6)

- **Motivation:**
  Approximate evaluation of a function which is known only by its values at a set of data points
- **Interpolation:**
  the process of fitting a function of a particular nature (typically a polynomial, or a collection of polynomials) through the data.
- **Calssification:**
  - Polynomial interpolation, C
  - Cubic spline approximation,
  - Least square approximation

Polynomial Interpolation (MM6)

- Any continuous function can be approximated to any required accuracy by a polynomial (Weierstrass Theorem)
- Polynomials are the only functions, which we can, theoretically at least, evaluate exactly
- Efficient evaluation of a polynomial (Horner's Rule)

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$
$$p(x) = \{[(a_n x + a_{n-1})x + a_{n-2}]x + \cdots + a_1\}x + a_0$$

Horner's Rule - Example

Download horner.m and horner2.m

```
function p=horner(a,x)
 % horner's rule for evaluating the polynomial
% p(x)= a(1) + a(2) x + .... a(n)x^(n-1)
n=length(a);
p=a(n);
for k=n-1:-1:1
   p=p*x+ a(k);
end
```

```
function p=horner2(a,x)
% horner's rule for evaluating the polynomial
% p(x)= a(1)x^(n-1)+ a(2) x^[n-2] + ... a(n)
n=length(a);
p=a(1);
for k=2:n
   p=p*x+ a(k);
end
```

```
>> horner([1,2,3,4,5,6],0.1234)
ans =    1.3013

>> format long
>> horner([1,2,3,4,5,6],0.1234)
ans =   1.301330079437031

>> polyval([1,2,3,4,5,6],0.1234)
ans =   6.684039853696318

>> polyval([6,5,4,3,2,1],0.1234)
ans =   1.301330079437031

>> horner2([6,5,4,3,2,1],0.1234)
ans =   1.301330079437031
```

Lagrange Interpolation (MM6)

- **Problem formulation:**

Suppose we are given the values of a function *f* at *N+1* distinct points (nodes), we wish to find a polynomial *p* with minimum degree such as

p(x_k)=f(x_k), k=0, 1,2 ,…N

- **uniqueness**

$$p(x) = a_m x^m + a_{m-1} x^{m-1} + \cdots + a_1 x + a_0$$

$$\begin{cases} a_m x_0^{\ m} + a_{m-1} x_0^{\ m-1} + \cdots + a_1 x_0 + a_0 = f(x_0) \\ a_m x_1^{\ m} + a_{m-1} x_1^{\ m-1} + \cdots + a_1 x_1 + a_0 = f(x_1) \\ \vdots \\ a_m x_N^{\ m} + a_{m-1} x_N^{\ m-1} + \cdots + a_1 x_N + a_0 = f(x_N) \end{cases} \Leftrightarrow AX = F$$

Vandermonde determinant:

$$V = \begin{vmatrix} 1 & x_0 & \cdots & x_0^{\ N} \\ 1 & x_1 & \cdots & x_1^{\ N} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & \cdots & x_N^{\ N} \end{vmatrix}$$

Lagrange Polynomial (MM6)

If we can find a number of polynomials $l_j(x)$ $(j = 0,1,...N)$ of degree at most N such that

$$l_j(x_k) = \delta_{jk} = \begin{cases} 1 & if \quad j = k \\ 0 & if \quad j \neq k \end{cases}$$

then the Lagrange polynomial is defined as

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

The Lagrange basis polynomials $l_j(x)$ $(j = 0,1,...N)$ are defined as

$$l_j(x) = \frac{(x - x_0)\cdots(x - x_{j-1})(x - x_{j+1})\cdots(x - x_N)}{(x_j - x_0)\cdots(x_j - x_{j-1})(x_j - x_{j+1})\cdots(x_j - x_N)}$$

Lagrange Method - Example

find the Lagrange interpolation polynomial for the following data:

x	1	1.5	2
$f(x)$	0.0000	0.4055	0.6931

Lagrange basis polynomials:

$$l_1(x) = \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} = \frac{(x-1.5)(x-2)}{(1-1.5)(1-2)} = 2(x-1.5)(x-2)$$

$$l_2(x) = \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} = \frac{(x-1)(x-2)}{(1.5-1)(1.5-2)} = -4(x-1)(x-2)$$

$$l_3(x) = \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_1)} = \frac{(x-1)(x-1.5)}{(2-1)(2-1.5)} = 2(x-1)(x-1.5)$$

The Lagrange polynomial is:

$$p(x) = f(x_1)l_1(x) + f(x_2)l_2(x) + f(x_3)l_3(x)$$
$$= -1.6220(x-1)(x-2) + 1.3862(x-1)(x-1.5)$$

The estimation of $f(1.2)$ is

$$p(1.2) = \cdots = 0.176348$$

Lagrange Method – Matlab Codes

```
function y=lagrange(xdat,ydat,x)
% Computes the Lagrange polynomial
% using the datapoints in (xdat, ydat)

N=length(x); M=length(xdat);
D=zeros(M,M); y=zeros(1,N);
for k=1:N
   xtst=x(k);
   [xb,ind]=sort(abs(xtst-xdat));
   xsort=xdat(ind);     D(:,1)=ydat(ind)';
   for j=1:M
      for i=1:M-j
         D(i,j+1)=(D(i+1,j)-D(i,j))/(xsort(i+j)-xsort(i));
      end
   end
   xdiff=xtst-xsort;
   prod=1;
   for i=1:M
      y(k)=y(k)+prod*D(1,i);
      prod=prod*xdiff(i);
   end
end
```

```
>> format long
>> lagrange([1,1.5,2],[0.0,0.4055,0.6931],1.2)

ans =   0.176348000000000
```

```
% Eksempel: Befolkningstal i Danmark.

clear, clc

xdat=[1930,1940,1950,1960,1970,1980,1990,2000];
ydat=[3.55,3.84,4.28,4.59,4.94,5.12,5.14,5.33];

x=1930:2000;
 y=lagrange(xdat,ydat,x);

plot(xdat,ydat,'.',x,y)

format short
 disp('Befolkningstal i 1955 (i millioner):')
disp(lagrange(xdat,ydat,1955))
```

Lagrange Interpolation Error

Suppose the function $f(x)$ is $N+1$ times continuously differentiable on the interval $[a,b]$ which contains the distinct nodes $x_0, x_1, \ldots x_N$. Let $p(x)$ be the Lagrange interpolation polynomial , then for any $x \in [a,b]$, there is

$$f(x) - p(x) = \frac{(x-x_0)(x-x_1)\cdots(x-x_N)}{(N+1)!} f^{(N+1)}(\zeta)$$

for some $\zeta \in [a,b]$.

Continue the example: find the Lagrange interpolation polynomial for data:

x	1	1.5	2
$f(x)$	0.0000	0.4055	0.6931

The Lagrange polynomial is:

$$p(x) = f(x_1)l_1(x) + f(x_2)l_2(x) + f(x_3)l_3(x) = -1.6220(x-1)(x-2) + 1.3862(x-1)(x-1.5)$$

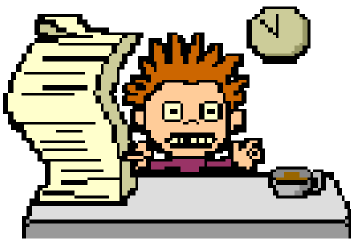
The estimation of $f(1.2)$ is $p(1.2) = \cdots = 0.176348$

The original data from the function $\ln x$. The estimation error at $x = 1.2$ can be analyzed as $\ln(1.2)$ - $p(1.2) =$ $0.1823 - 0.1763 = 0.0060$.

According to the error analysis, there is

$$\ln(1.2) - p(1.2) = \frac{(1.2-1)(1.2-1.5)(1.2-2)}{4!} \frac{d^4(\ln x)}{dx^4}\Big|_{x=\zeta} = \frac{0.048}{24} * \frac{6}{x^4}\Big|_{x=\zeta} \le 0.012$$

Exercises (MM6)



Question One (Exercise 4.2.1, 4.2.2, pp.83):

Find the Lagrange interpolation polynomial for the data

x	1	2	4
$f(x)$	3	2	1

- Use this to estimate $f(1.5)$;
- Repeat the above exercise with the additional data $f(0) = 4,\ f(3) = 1$.

Question Two (Exercise 4.2.4, 4.2.5, pp.83):

Consider the following table of values of the cosine function:

x	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
$cos(x)$	1.000	0.9950	0.9801	0.9553	0.9211	0.8776	0.8253	0.7648	0.6967

- write down the Lagrange interpolation polynomial using the nodes 0.0, 0.1, 0.2 and 0.3;
- Estimate the value of $cos(1.4)$ using this polynomial;
- Obtain the error bound for the above approximation.

MM7 Spline Interpolation Methods

Reading materials: Section 4.4 and 4.5

Motivation

- Accuracy of using single polynomial
- Computation efficiency

- Piecewise polynomials

- Continuity/smoothness problem at the transition points

Spline Function

Let $a=x_0 < x_1 < \dots x_N=b$. A function $s: [a,b] \to R$ is a **spline** or **spline function** of degree m with knots (or nodes) $x_0, x_1, \dots x_N$ if

1) s is a piecewise polynomial such that, on each subinterval $[x_k, x_{k+1}]$, s has degree at most m, and
2) s is $m-1$ times differentiable everywhere

Spline Function - Examples

Example 11: show that the following function is a spline of degree 2 (quadratic spline):

$$s(x)=\begin{cases} x^2+x & x\in[-1,0] \\ x & x\in[0,2] \\ x^2-3x+4 & x\in[2,5] \end{cases}$$

Example 12: is the following function a spline (cubic spline)?

$$s(x)=\begin{cases} x^3+2x & x\in[-1,0] \\ 2x+2x^2 & x\in[0,1] \\ x3-x2+5x-1 & x\in[1,3] \end{cases}$$

Linear Spline Interpolation

- Spline of degree 1

At the nodes $a=x_0<x_1<x_2\cdots<x_n=b$, there is

$s(x_k)=f(x_k) \quad for \quad k=0,1,2\cdots,n$

In the node interval $[x_k,x_{k+1}]$, s is a polynomial of degree 1 passing through the points $(x_k, f(x_k))$ and $(x_{k+1}, f(x_{k+1}))$, thereby

$$s(x)=f(x_k)+\frac{f(x_{k+1})-f(x_k)}{x_{k+1}-x_k}(x-x_k), \quad x\in[x_k,x_{k+1}]$$

simply

$s_k(x)=a_k+b_k(x-x_k)$

Example: Find the linear spline which takes the following values:

$x_0=0.0,\ f(x_0)=0.00;\quad x_1=0.2,\ f(x_1)=0.18;$

$x_3=0.3,\ f(x_3)=0.26;\quad x_4=0.5,\ f(x_4)=0.41;$